USER’S MANUAL OF THE BCTRY SYSTEM

OF

CLUSTER AND FACTOR ANALYSIS

ROBERT C. TRYON
UNIVERSITY OF CALIFORNIA, BERKELEY

AND

DANIEL E. BAILEY
UNIVERSITY OF COLORADO

ORIGINAL EDITORS

MAIN CONTRIBUTORS

DANIEL BAILEY
JOHN BAUER
JAMES CAMERON
CHEN-LIN CHU
DON FLORY
TOM KIBLER
ELEANOR KRASNOW
DAVID MATULA
ROBERT MENHENETT
WILLIAM MEREDITH
LOUISE RUHLAND
ROBERT RUSSELL
VALERIE SIEBERT
ROBERT TRYON
JOHN VINSONHALER
JOHN WOLFE

Note 1: Jerome Smith of the U.C., Berkeley Computer Center retrieved the original ALL CAPS BCTRY User’s Manual, and other BCTRY documentation, from a 9-track tape in the late 1970s. Christopher Winters and Robert Dean converted the text from all caps to upper/lower in the early 1980s.

Note 2: Recently found ALL CAPS missing sections of the Manual have been added (2022).

Table of contents

PREFACE.. ROBERT C. TRYON

THE BC TRY SYSTEM OF CLUSTER AND FACTOR ANALYSIS **1, PRESENTLY PROGRAMMED FOR IBM 709, 7090, 7094 IS AN INTEGRATED ASSEMBLAGE OF COMPONENT PROGRAMS DESIGNED TO PERFORM ALL THE MAIN TYPES OF CLUSTER AND FACTOR ANALYSIS ON VARIABLES (V-ANALYSIS) OR ON INDIVIDUAL OBJECTS (O-ANALYSIS). THE SYSTEM ESTABLISHES AS A FACT THE THEORETICAL PROPOSITION THAT ALL THESE METHODS (AND MANY OTHERS) ARE MERE VARIANTS OF ONE GENERAL FORMULATION OF MULTIDIMENSIONAL ANALYSIS BASED ON DOMAIN SAMPLING PRINCIPLES (TRYON, 1959).

THIS USER'S MANUAL GIVES THE DETAILS OF CONTROL CARD PREPARATION FOR THE MANY VARIETIES OF CLUSTER AND FACTOR ANALYSIS THAT CAN BE PERFORMED BY BC TRY. BEING ON TAPE IT IS AVAILABLE TO ANY COMPUTER CENTER THAT HAS THE BC TRY SYSTEM. A COPY OF THE MANUAL CAN BE PRINTED FOR ANY USER IN MINUTES.

A COMPANION ISSUE IS THE PROGRAMMER'S MANUAL. ALSO ON TAPE, IT PROVIDES THE PROGRAMMER AT THE COMPUTER CENTER WITH ESSENTIAL INFORMATION ON THE CONVERSION OF THE BC TRY SYSTEM (AVAILABLE ON A SOURCE TAPE AND AN OBJECT TAPE) TO THE PARTICULAR COMPUTER USED.

ANOTHER COMPANION ISSUE IS THE USER'S HANDBOOK OF EXAMPLES, ALSO ON TAPE (IN PROCESS). THE HANDBOOK IS A CONCRETE EXTENSION OF THE 'GENERAL EXPOSITION' SECTION OF A FORTHCOMING BOOK (SEE BELOW). IT PROVIDES THE SPECIFIC CONTROL CARDS FOR ALL THE MAIN FORMS OF V AND O ANALYSIS, AS APPLIED TO THE FAMOUS 'HOLZINGER-HARMAN (HH) PROBLEM' AND TO OTHER EXAMPLES, THE RAW DATA CARDS ON THE HH PROBLEM WILL ALSO BE AVAILABLE FROM THIS TAPE.

SINCE THE BC TRY SYSTEM IS LIKELY TO UNDERGO REVISION, THE TWO MANUALS AND THE HANDBOOK THAT CARRY THE SAME DATE AND SYSTEM VERSION INDICATION WILL ALWAYS REFER TO THE SAME BC TRY SYSTEM, AS CURRENTLY REVISED.

IN PREPARATION IS THE BOOK, CLUSTER AND FACTOR ANALYSIS, BY R.C. TRYON AND D. E. BAILEY. CURRENTLY AVAILABLE FOR LIBRARY ISSUE IS THE DITTO VERSION OF THE THEORY OF THE BC TRY SYSTEM, BEING THE DERIVATIVE LOGIC OF THE PROGRAMS OF THE SYSTEM. FROM THIS THEORY, THERE DEVELOPS THE PROGRAM DESIGN (IN PROCESS) OF THE VARIOUS COMPONENTS OF THE SYSTEM. A NON-TECHNICAL SECTION ON GENERAL EXPOSITION (IN PROCESS) IS INTENDED TO INTRODUCE THE READER BY THE USE OF A WEALTH OF EXAMPLES TO THE MAIN CAPABILITIES OF THE SYSTEM, CENTRAL TO THIS PRESENTATION ARE ILLUSTRATIVE ANALYSES ON THE HH PROBLEM, AVAILABLE IN DETAIL, AS INDICATED ABOVE, IN THE USER'S HANDBOOK.

THE FIRST SIMPLE VERSION OF THE SYSTEM WAS PROGRAMMED ON IBM 701 IN 1957, THEN RATHER FULLY DEVELOPED ON THE IBM 704 IN 1960-62, AND FINALLY CONVERTED TO THE IBM 7090 IN 1963-64. MANY MINDS AND HANDS HAVE BUILT THE BC TRY SYSTEM OVER THE LAST SEVEN YEARS. I AM ESPECIALLY

INDEBTED TO THE TWO PROGRAMMING COORDINATORS WHO INTEGRATED THE COMPLEX PROGRAMMING. DANIEL BAILEY (WHO WROTE THE 701 VERSION WITH JACK NEUHAUS) SUPERVISED THE INITIAL DEVELOPMENT OF THE 704 PROGRAMS, AND JOHN VINSONHALER SUPERVISED THE COMPLETION OF THAT WORK AND THE TASK OF 7090 PROGRAMMING. ROBERT RUSSELL WAS THE MAIN INNOVATOR OF MANY SPECIAL PROGRAMMING FEATURES OF THE SYSTEM AND WROTE MANY OF ITS PROGRAMS. BUT OTHERS HAVE MADE SIGNAL CONTRIBUTIONS TO THE PROGRAMMING PROPER, ESPECIALLY JOHN VINSONHALER, DAVID MATULA AND JOHN BAUER. IN FORMULATING THE METHODS FOR PROGRAMMING, THOSE MOST CENTRALLY INVOLVED WERE DANIEL BAILEY, JOHN VINSONHALER, ROBERT RUSSELL, WILLIAM MEREDITH, JOHN BAUER, AND THE PRINCIPAL INVESTIGATOR. CHEN-LIN CHU, WITH THE PRINCIPAL INVESTIGATOR, PERFORMED MANY METHODOLOGICAL STUDIES THAT LED TO DECISIONS INCORPORATED IN THE PROGRAMS. OTHERS WHO MADE IMPORTANT CONTRIBUTIONS IN DEVELOPMENT OR PRESENTATION ARE.. JAMES CAMERON, MIKE DAVIDSON, DON FLORY, KENT MITCHELL, VALERIE SIEBERT, JOHN WOLFE AND SARA BAILEY. PATRICIA BAUER, OF THE UNIVERSITY OF COLORADO COMPUTING CENTER, WAS INSTRUMENTAL IN THE PREPARATION OF THIS MANUAL IN KEY PUNCH FORM.

THE COMPUTER CENTER PROVIDED A CONSIDERABLE AMOUNT OF FREE PROGRAMMERS' TIME AND MACHINE TIME. SCORES OF INVESTIGATORS WHO HAVE USED THE SYSTEM HAVE CONTRIBUTED TO IT BY PAYING FOR MACHINE TIME AND BY HUNTING FOR 'BUGS' IN THE SYSTEM.

R. C. T. SEPTEMBER 1964

ADDITIONS TO PREFACE (JULY 1965)

SINCE THE PRECEDING PREFACE WAS WRITTEN IN SEPTEMBER 1964, PROFESSOR BAILEY HAS SET UP THE USERS' MANUAL ON TAPE AT THE UNIVERSITY OF COLORADO, AND HAS PROGRAMMED AN EDITOR PROGRAM TO REVISE IT BY THE COMPUTER. THE TAPED MANUAL IS A CONDENSATION BY PROFESSOR BAILEY OF THE ORIGINAL GARGANTUAN DITTOED LIBRARY VERSION OF THE MANUAL ISSUED FEBRUARY 1964, INCLUDING THE THREE SUPPLEMENTS ISSUED FEBRUARY 1964, FEBRUARY 1965, AND APRIL 1965.

SOME OF THE COMPONENTS OF THE SYSTEM HAVE BEEN SO DRASTICALLY REVISED (E.G., SPAN) THAT WE HAVE CHANGED THE CODE NAMES OF THEM. FURTHERMORE, WE HAVE PROGRAMMED THE OLD COMPONENTS SO THAT IF YOU SHOULD CALL FOR THEM UNDER THEIR OLD NAMES YOU WILL GET A PROGRAM STOP. YOU WILL HAVE TO CALL THEM BY THEIR NEW CODED NAMES AND MUST PREPARE CONTROL CARDS FOR THEM AS DIRECTED BY THE NEW USERS' DESCRIPTIONS GIVEN IN THIS TAPED MANUAL. HERE ARE THE MAIN CHANGES..

REVISED COMPONENTS		ENTIRELY NEW COMPONENTS
OLD NAME	NEW NAME	
DAP	DAP2	COMMENT
CSA	CSA2	DPRINT
NCSA	NCSA2	RSCAT
SPAN	SPAN2	BIGNV
		SAMPLER (TAG,PICK,PRY OR PRIEST, TRYST)
		MERGER (SETCOPY, COPY, PRANK)
		SMIS (BC TRY ADAPTATION)
		LEAKEY

OTHER CHANGES.. A NEW FILE OF SUBJECT NAMES, ONAMS1, HAS BEEN ADDED (SEE ONAMS1 IST I/O TABLE). THE FACS COMPONENT NOW OUTPUTS PUNCHED CARDS OF THE FSCOR1 FILE.
THIS MANUAL ALSO INITIATES A NEW SECTION CALLED INCITATION.

FOR THE NEW USER OF THE SYSTEM, I HAVE WRITTEN A NON-TECHNICAL ACCOUNT OF IT UNDER THE TITLE 'THE COMPONENT PROGRAMS OF THE BC TRY SYSTEM' (TRYON, 1964B, AVAILABLE IN DOTTOED LIBRARY VERSION). AFTER READING THE INTRODUCTION IN THE MANUAL BY PROFESSOR BAILEY AND FOLLOWING HIS SUGGESTIONS THERE FOR GETTING A GENERAL GRASP OF THE SYSTEM, THE NEW USER MIGHT ALSO INCREASE HIS UNDERSTANDING OF WHAT THE COMPONENTS MEAN AND DO BY READING THOSE SECTIONS IN 'THE COMPONENT PROGRAMS, ETC.' THAT COVER THE PROGRAMS HE WISHED TO APPLY TO HIS DATA. OF SPECIAL IMPORTANCE IN EACH OF THESE SECTIONS IS THE HIGHLY CONDENSED 'INPUT-OUTPUT SUMMARY'. THIS DOCUMENT ON THE COMPONENTS DOES NOT INCLUDE ACCOUNTS OF THE NEW AND REVISED PROGRAMS THAT ARE INCLUDED IN THIS MANUAL. I THINK, HOWEVER, THAT THE USERS' DESCRIPTIONS GIVEN HERE WILL BE SUFFICIENT.

A NEW SUMMARY DESCRIPTION OF THE SYSTEM WILL SHORTLY BE PUBLISHED (TRYON AND BAILEY, 1965), COPIES OF WHICH MAY BE OBTAINED FROM THE AUTHORS. THIS ACCOUNT DESCRIBES THE SYSTEM AS A CONTRIBUTION TO MULTIVARIATE BEHAVIORAL RESEARCH AND TO COMPUTER SCIENCE.

THE EARLIER PREFACE DID NOT ACKNOWLEDGE IN SUFFICIENT DETAIL THE CONTRIBUTIONS OF MANY PERSONS TO THE DEVELOPMENT OF THE SYSTEM. IN THE DOCUMENT, 'THE COMPONENT PROGRAMS, ETC.' MENTIONED ABOVE, I HAVE INCLUDED FOR EACH PROGRAM A SECTION ON 'SOURCES', GIVING THERE THE NAMES OF THE PERSONS MOST CENTRALLY INVOLVED IN THE CONCEPTUALIZATION, PROGRAM DESIGN AND ACTUAL PROGRAMMING OF EACH COMPONENT. IN THE LIBRARY VERSION OF THE MANUAL, IN ORDER TO GIVE COHESION TO, AND SHOW THE INTEGRATION OF THE MANY PROGRAMS OF THE SYSTEM, I MYSELF WROTE MOST OF THE 'INTRODUCTION' AND 'METHOD' SECTIONS OF THE STANALOG COMPONENTS. THE =USE' SECTIONS WERE USUALLY WRITTEN JOINTLY BY THE PROGRAMMER AND MYSELF. THE AUTHORS OF THE INTRODUCTORY SECTIONS AND THE TIME CHARTS IN THE LIBRARY VERSION ARE LISTED IN THE LIBRARY VERSION. THIS TAPED MANUAL IS NOT MERELY A CONDENSATION BY PROFESSOR BAILEY OF THE LIBRARY VERSION. IT IS AN INTEGRATED RECASTING OF IT, WITH A NEW INTRODUCTION. MY OWN ROLE IN THE TAPING HAS BEEN AS AN EDITOR, AND AUTHOR OF THE NEW AND REVISED COMPONENTS NOT COVERED IN THE LIBRARY VERSION (LEAKEY AND SMIS ARE BY BAILEY). I OWE MUCH TO MY PRESENT CREATIVE GROUP OF PROGRAMMERS WHO SHOW MUCH ACUMEN IN

CONVERTING TO USEFUL PROGRAMS MY TOO OFTEN RATHER GENERAL VERBAL AND MATHEMATICAL ALGORITHMS --TO TOM KIBLER FOR HIS CSA2, SPAN2, AND RSCAT, TO ELEANOR KRASNOW FOR HER DAP2, PRANK, AND UP-COMING SAMPLER, AND ESPECIALLY TO JOHN BAUER FOR HIS GENERAL SYSTEM PROGRAMMING, HIS AID TO AND SUPERVISION OF THE OTHER PROGRAMMERS, AND FOR HIS OWN SPECIAL WRITING OF COMMENT, SETCOPY, COPY, AND THE BC TRY ADAPTATION OF SMIS.

SPECIAL NOTE ON BUGS.

THOUGH ALL THE NEW AND REVISED COMPONENTS HAVE BEEN SUBJECTED TO STANDARD DEBUGGING BY PROGRAMMERS, AND MANY PROBLEMS BEEN ANALYSED BY THE COMPONENTS, SPECIAL UNDETECTED BUGS CAN NEVERTHELESS STILL EXIST. THE 'SLEEPERS' USUALLY APPEAR IN PROBLEMS THAT HAVE AN ATYPICAL PATTERN OF INPUT DATA AND SELECTED OPTIONS - A PATTERN NOT NOW BUG-TESTED AMONG THE MANY POSSIBLE, OFTEN UNKNOWN, PATTERNS THAT COULD BE TESTED. IF YOU SHOULD UNCOVER SUCH A BUG, PLEASE NOTIFY THE LOCAL BC TRY CONSULTANT.

EXAMPLES

A COMPENDIUM OF ILLUSTRATIONS OF THE CONTROL CARDS AND PRINTOUTS OF ALMOST ALL OF THE COMPONENTS OF THE BC TRY SYSTEM IS NOW MADE AVAILABLE ON TAPE TO ALL COMPUTER CENTERS THAT HAVE THE JULY 1965 VERSION OF THE SYSTEM. COPIES OF THIS COMPENDIUM ARE READILY PRINTED FROM THIS TAPE AND SHOULD BE AVAILABLE AT DIFFERENT PLACES CONVENIENT TO USERS.

R.C.T., JULY 31, 1965

**1 SUPPORTED IN LARGE PART BY USPHS, NIH GRANT MH 0811 - 01 TO 04 AND GRANT MH 08314 - 01 TO 05 (PROJECT CAP - CLUSTER ANALYSIS PROGRAMS) PRINCIPAL INVESTIGATOR -- R. C. TRYON.

INTRODUCTION.. DANIEL E. BAILEY

THIS MANUAL, PREPARED BY PROJECT CAP FOR COMPUTER PRINTOUT LISTING, IS A SLIGHTLY ABBREVIATED FORM OF THE DITTOED BOOK (PUBCAP) VERSION OF THE USER'S MANUAL OF THE BC TRY SYSTEM. IT IS A TAPE VERSION DESIGNED TO PERMIT OTHER INSTALLATIONS REQUESTING THE SYSTEM TO HAVE ACCESS TO THE MANUAL BY REPRODUCING IT AT THEIR SITES.

THE PRESENT FORM OF THE USER'S MANUAL INCLUDES THREE MAJOR SECTIONS.. A DESCRIPTION OF THE OVERALL CHARACTERISTICS OF THE SYSTEM, A DISCUSSION OF GENERAL RULES GOVERNING THE USE OF THE BC TRY PROGRAMS, AND A DETAILED PRESENTATION OF THE USE OF EACH SPECIFIC COMPONENT IN TERMS OF THE CONTROL CARDS NECESSARY TO USE A COMPONENT AND OF THE INPUT AND OUTPUT OF THE COMPONENT.

THIS IS NOT A COMPLETE DESCRIPTION OF THE SYSTEM OR OF THE BASIC THEORY OF THE SYSTEM. RATHER, IT IS A GUIDE INTENDED SOLELY TO DIRECT A USER OF THE PROGRAMS IN THE SYSTEM TO SET UP CORRECTLY A CONTROL DECK FOR A COMPUTER RUN. THE MANUAL INCLUDES, AT THE PRESENT TIME, BRIEF INTRODUCTIONS TO THE COMPONENTS AND DISCUSSIONS OF METHODS, WHERE NECESSARY, TO UNDERSTAND THE OPERATION OF THE PROGRAMS.

THE UNITIATED USER MAY AT FIRST SEEM OVERWHELMED BY THE APPARENT COMPLEXITY AND DIFFICULTY OF THE BC TRY SYSTEM AND THE USERS' MANUAL. LET ME HASTEN WITH ASSURANCES THAT THE COMPLEXITY AND DIFFICULTY ARE APPARENT ONLY. THE USE OF THE SYSTEM FOR STANDARD CLUSTER AND FACTOR ANALYSES IS SIMPLICITY ITSELF. ONLY WHEN THE USER HAS GONE THROUGH THE FIRST, AND STANDARD, EXPERIENCES WITH THE SYSTEM WILL HE BECOME 'CREATIVE' IN HIS USE OF THE SYSTEM (SEE INCITATION). HE WILL HAVE COME TO 'AN UNDERSTANDING' WITH THE SYSTEM AND WILL NEED NO ASSURANCES REGARDING ITS BASIC SIMPLICITY.

THE BEGINNER SHOULD NOT ATTEMPT TO READ THIS MANUAL AT A SINGLE SITTING. DO NOT ATTEMPT TO BEGIN AT THE BEGINNING AND WORK THROUGH TO THE END. THERE IS NO PROFIT IN SYSTEMATICALY READING THE MANUAL. RATHER, I RECOMMEND THAT THE READER BEGIN WITH THE PREFACE IF HE HAS NOT ALREADY READ IT. NEXT, AFTER COMPLETING THE PREFACE AND INTRODUCTION, TURN TO THE GENERAL SECTIONS ON FACTORING (FACTOR) AND TYPOLOGY (TYPO). THESE SECTIONS OUTLINE THE GENERAL PURPOSES AND METHODS OF CLUSTER AND FACTOR ANALYSIS EMBODIED IN THE SYSTEM.

THE READER SHOULD TURN NEXT TO THE SECTIONS DESCRIBING THE COMPONENT PROGRAMS HE WILL USE TO EXECUTE THE DESIRED ANALYSIS. THE COMPONENT PROGRAMS USED IN STANDARD CLUSTER ANALYSIS AND ORTHODOX FACTOR ANALYSIS ARE GIVEN BELOW IN A STEP BY STEP OUTLINE. HERE AGAIN THE READER NEED NOT BE CONCERNED WITH THE SPECIFIC DETAILS OF THE RULES FOR CARD PUNCHING, ETC. RATHER, READ THE INTRODUCTORY COMMENTS AND METHODS SECTIONS. HAVING THUS BECOME ACQUAINTED WITH THE BROAD PRINCIPLES OF THE USE OF THE SYSTEM THE NOVICE IS READY TO GET DOWN TO THE TECHNICAL DETAILS.

EACH OF THE COMPONENTS OF THE SYSTEM IS DESCRIBED IN THIS MANUAL. PART OF THESE DESCRIPTIONS IS A USE SECTION DETAILING THE SPECIFIC

COMPONENTS, THE COMPOUNDS, AND SOME OF THE MAJOR OPTIONS OF THE COMPONENTS ARE GIVEN IN THE FOLLOWING LISTING.

COMPONENT PROGRAMS OF THE BC TRY SYSTEM

EXECUTIVE OPERATIONS.

GEP -- RECEIVES CONTROL FROM MONITOR, SELECTS AND PASSES CONTROL TO COMPONENT PROGRAMS AS REQUIRED BY THE ANALYSIS, TERMINATES THE ANALYSIS, AND PASSES CONTROL BACK TO MONITOR.

DATA SHARING OPERATIONS.

(1) INTERMEDIATE STORAGE TAPE (IST) - RECEIVES MAJOR RESULTS CALCULATED AND OUTPUT BY THE COMPONENT PROGRAMS, STORES THEM ON FILES, AND PASSES THEM AS INPUT TO OTHER COMPONENT PROGRAMS.

GIST -- RECEIVES INTO IST FILES DATA INPUT BY ANALYST, AND WHEN DIRECTED BY GEP OUTPUTS RESULTS IN THE IST FILES ON DECIMAL CARDS.

GIVE -- TERMINATES CALCULATIONS AND DIRECTS IST TO DUMP ALL ITS FILES ON A DECK OF BINARY CARDS FOR LATER RESTARTING OF THE PROBLEM AT THE POINT OF TERMINATION.

TAKE -- RESTORES ON IST FILES ALL RESULTS FROM A GIVE DECK.

(2) DATA STORAGE TAPE (DST) -- STORES THE INPUT RAW DATA (E.G., SCORES) ON WHICH ANALYSES ARE TO BE PERFORMED.

DAP -- PREPARES, READS, AND ERROR CHECKS RAW SCORES FOR INPUT TO COR2, COR3, AND FACS.

STANALOG (STATISTICAL AND LOGICAL) OPERATIONS.
V-ANALYSIS (ANALYSIS OF VARIABLES)

(1) CORRELATING VARIABLES

COR2 -- FOR COMPLETE DATA, COMPUTES CORRELATIONS BETWEEN VARIABLES AND OTHER STATISTICAL CONSTANTS.

COR3 -- FOR MISSING DATA AND COMPLETE DATA, SAME AS COR2, BUT SLOWER. ALSO COMPUTES COVARIANCES.

(2) SUPPRESSING VARIABLES DURING FACTORING

SLEP1 SUPPRESSES SLEEPER VARIABLES DURING FACTORING., SLEP2 REACTIVATES THEM AFTER FACTORING AND COMPUTES THEIR FACTOR STATISTICS, RETURNING THEM TO THEIR ORIGINAL STATUS AS VARIABLES.

(3) COMPUTES VARIANCE TO BE DESCRIBED BY FACTORING

DVP -- ESTIMATES COMMUNALITIES BY ANY ONE OF SIX WAYS FOR INPUT TO DIAGONAL CELLS IN THE R-MATRIX. IT CAN ALSO INSERT UNITIES OR ZEROS AS DIAGONAL VALUES.

(4) FACTORS BY SIMPLE SUMS OR LEAST SQUARES ON KEY CLUSTER OR TOTAL-SET DIMENSIONS WITH DIAGONAL VALUES AND ITERATIVE FACTORING OPTIONAL.

CC5 -- FACTORS ON KEY CLUSTER (SUBSET) DIMENSIONS WITH USE OF DIAGONAL VALUES, AND REITERATES. OPTIONS PERMIT DIAGONAL (PV) FACTORING AND CENTROID (CENT) TOTAL-SET OR SALIENT-SET FACTORING.

NC2 -- SAME AS CC5 EXCEPTING THAT DIAGONAL VALUES ARE IGNORED AND NO REITERATION OF FACTORING IS ALLOWED.

LEAKEY -- SAME AS CC5 EXCEPT FOR THE SOLUTION OF COEFFICIENTS BY LEAST SQUARES

FALS -- FACTORS ON TOTAL-SET DIMENSIONS THAT ARE DEFINED BY ALL VARIABLES USING ANY ONE OF THREE LEAST SQUARES SOLUTIONS, AND REITERATES. OPTIONS PERMIT PRINCIPAL AXES (PRIN OR PFA), CANONICAL (CANON OR CFA), OR AUGMENTED (AUG OR AFA) FACTORING.

FAST -- AUXILIARY PROGRAMS, PRESENTLY INCLUDING ONE THAT COMPUTES A RESIDUAL MATRIX AFTER FACTORING ON ANY DIMENSION, AND ONE THAT REPRODUCES A CORRELATED MATRIX FROM A FACTOR MATRIX.

(5) STRUCTURE ANALYSIS (INCLUDING ORTHOGONAL AND OBLIQUE ROTATION).

CSA (STATISTICAL) -- USING COMMUNALITIES, COMPUTES OBLIQUE FACTOR COEFFICIENTS, CORRELATIONS BETWEEN OBLIQUE RAW CLUSTER SCORES AND BETWEEN DIMENSIONS ('FACTORS'), ALSO CALCULATES DOMAIN VALIDITIES OF RAW CLUSTER SCORES ('ACCURACY OF FACTOR ESTIMATES') AND THE AUGMENTED CORRELATION MATRIX. IT ALSO STRUCTURES A CLUSTERED RAW CORRELATION MATRIX (CLUR).

NCSA (STATISTICAL) -- SAME AS CSA, WITHOUT COMMUNALITIES, BUT PROVIDES NO AUGMENTED CORRELATION COEFFICIENTS.

SPAN (GEOMETRIC) -- ALLOCATES VARIABLES TO MINIMAL SUBSPACES (A 'SIMPLE STRUCTURE' SOLUTION) AND PLOTS THE VARIABLES AS POINTS ON THE SURFACE OF A MINIMAL SET OF SPHERES, PHYSICALLY ROTATED FOR VISUALLY CENTERING THE CONFIGURATION.

GYRO -- COMPUTES QUARTIMAX OR VARIMAX ROTATION FOLLOWING ANY FACTORING PROCEDURE.

(6) ANALYZING A PROBLEM WITH A LARGE NUMBER OF VARIABLES
BIGNV -- PROGRESSIVELY PERFORMS V-ANALYSES ON SAMPLES OF 90

VARIABLES TO THE POINT AT WHICH THE CLUSTER AND FACOTR STRUCTURE OF THE ENTIRE SET OF VARIABLES IS REPRESENTED IN A CONDENSED STRUCTURE ANALYSIS.

(7) COMPARING THE DIMENSIONS OF ONE GROUP WITH THOSE OF ANOTHER ('MATCHING FACTORS')

COMP -- PERFORMS A COMPARATIVE ANALYSIS OF THE RESULTS OF ANY TYPE OF V-ANALYSIS ON SEPARATE GROUPS OF SUBJECTS MEASURED BY A COMMON SET OF VARIABLES. COMP1 PREPARES THE INPUT DECKS OF THE DIFFERENT GROUPS. COMP2 ADJOINS THE FACTOR MATRICES OF THE GROUPS, COMPUTES SIMILARITY INDICES OF THEIR DIMENSIONS AND RUNS ON INTO A SINGLE V-ANALYSIS THAT ENCOMPASSES ALL THE DIMENSIONS OF ALL THE GROUPS. COMP CAN BE USED TO COMPARE DIFFERENT FACTORING PROCEDURES ON THE SAME GROUP.

SIMRO -- SAME PURPOSE AS COMP, BUT USES A LEAST SQUARES SOLUTION TO COMPARE THE DIMENSIONS OF DIFFERENT GROUPS WITH AN 'UNKNOWN' POPULATION.

STANALOG (STATISTICAL AND LOGICAL) OPERATIONS
O-ANALYSIS (CLUSTER ANALYSIS OF INDIVIDUAL OBJECTS)

(8) SCORING INDIVIDUALS ON GENERAL DIMENSIONS

FACS -- COMPUTES SIMPLE SUM SCORES OF INDIVIDUALS IN Z-SCORE FORM ON OBLIQUE CLUSTER DIMENSIONS, OR REGRESSION ESTIMATES ON ORTHOGONAL AND OBLIQUE FACTORED DIMENSIONS. FOR COMPLETE DATA ONLY.

FACS3 -- SAME AS FACS BUT FOR MISSING DATA.

RSCAT -- PLOTS THE CORRELATION SCATTER BETWEEN ANY OF THE CLUSTER OR FACTOR SCORES (OR INDIVIDUAL VARIABLES) AND COMPUTES PEARSON CORRELATIONS, BOTH ETAS, AND OTHER STATISTICAL CONSTANTS ON EACH SCATTER.

STANALOG (STATISTICAL AND LOGICAL) OPERATIONS
GENERAL O-ANALYSIS, USING CLUSTER OR FACTOR SCORES

EUCO-ANALYSIS -- COMPUTES EUCLIDEAN DISTANCE SCORE MATRIX BETWEEN INDIVIDUALS ON OBLIQUE OR ORTHOGONAL DIMENSION SCORES FROM FACS. THE DIMENSIONS CAN BE DIFFERENTIALLY WEIGHTED. EUCO RUNS ON INTO A FULL CYCLE KEY CLUSTER SOLUTION WHICH LOCATES KO PIVOTAL CORE O-TYPES.

OMARK -- PROJECTS MODEL AND OTHER MARKER INDIVIDUALS INTO EUCO-ANALYSIS, SETS SCORE GRADIENTS INTO THE SPAN CONFIGURATION, AND SELECTS THE FINAL SET OF M CORE O-TYPES.

EUFIT -- IDENTIFIES EACH OF THE NS INDIVIDUALS, WITH EACH OF THE M CORE O-TYPES, THUS REDUCING THE NS INDIVIDUALS TO M O-CLUSTERS, HOWEVER LARGE M MAY BE.

OSTAT -- METRICALLY DESCRIBES EACH OF THE M O-CLUSTERS IN TERMS OF SCORE PROFILE AND HOMOGENEITY.

PREDICT -- PREDICTS AN 'OUTSIDE' ATTRIBUTE OF O-CLUSTERS BY A NON-LINEAR INDEX OF CORRELATION AND REGRESSION. ASSESSES SAMPLING ERROR BY MONTE CARLO RUNS.

OCOMP -- COMPARES O-CLUSTER STRUCTURE IN DIFFERENT GROUPS AND ASSESSES THE EFFECTS OF UNIVARIATE AND MULTIVARIATE SELECTION IN THE POPULATION SAMPLES COMPARED.

STANALOG (STATISTICAL AND LOGICAL) OPERATIONS
SPECIFIC O-ANALYSIS, USING SCORES ON THE SPECIFIC VARIABLES

SPECO -- CLUSTERS INDIVIDUAL OBJECTS ON BASIS OF THEIR SCORES ON THE NV SPECIFIC VARIABLES, EITHER BY INVERSE FACTORING OR BY EUCO-ANALYSIS PROCEDURES.

TYPES OF ANALYSES PERFORMED BY BC TRY USING COMPONENTS IN TANDEM

THE CAPABILITIES OF THE SYSTEM MAY BE ILLUSTRATED BY SOME TYPES OF ANALYSES PERFORMED ON THE HH PROBLEM (SEE HANDBOOK).

V-ANALYSIS

EMPIRICAL KEY CLUSTER
PRESET KEY CLUSTER
PIVOT VARIABLE (DIAGONAL FACTORING)
CENTROID (THURSTONIAN AND SALIENT)
PRINCIPAL AXES (INCLUDING CANONICAL AND AUGMENTED)
ORTHOGONAL AND OBLIQUE ROTATION
BIFACTOR
DESIGNED AND ORDERED
RATIONAL NON-DIMENSIONAL
HIERARCHICAL
COMPARATIVE (GROUPS AND METHODS)

O-ANALYSIS

CLUSTER AND FACTOR SCORING
OBJECT CLUSTERING (SEE O-ANALYSIS ABOVE, I.E., EUCO, OMARK, EUFIT, OSTAT, PREDICT, OCOMP, SPECO).

STANDARD CLUSTER ANALYSIS OF VARIABLES AND OBJECTS

THE THIRTEEN STEPS OUTLINED BELOW GIVE THE DETAILED COMPONENT BY COMPONENT UTILIZATION OF THE COMPONENTS AND COMPOUNDS OF THE SYSTEM USED IN PERFORMING A STANDARD 'FULL CYCLE' KEY CLUSTER ANALYSIS OF VARIABLES AND SUBJECTS (OBJECTS).

A. INPUT OF RAW DATA

1. PREPARING THE NV VARIABLES ON NS INDIVIDUALS FOR PROCESSING BY LATER COMPONENTS.

   A. SETTING THE DATA ON THE DATA STORAGE TAPE (DST) — DAP

   B. WHEN NV OR NS ARE VERY LARGE, CHOOSING RANDOM OR FORCED SAMPLES — SAMPLER

B. V-ANALYSIS.. CLUSTER ANALYSIS OF VARIABLES

2. DETERMINING GENERALITY OF VARIABLES.. CORRELATION MATRIX (MISSING DATA REQUIRE COR3) — COR2

3. DECIDING ON VARIANCE TO BE FACTORED — DVP

4. PERFORMING DIMENSIONALITY ANALYSIS (FACTORING).

   A. SELECTING MAXIMALLY-COLLINEAR DIMENSION-DEFINING CLUSTERS AND COMPUTING ORTHOGONAL FACTOR COEFFICIENTS ON THEM — CC
   B. COMPUTING RESIDUALS TO TEST DIMENSIONAL SUFFICIENCY — CC

5. DESCRIBING THE OBLIQUE STRUCTURE OF THE DIMENSION-DEFINING CLUSTERS (AND OF DEPENDENT CLUSTERS, IF ANY)

   A. BY STATISTICAL QUANTITIES (A 'DIRECT SOLUTION') — CSA

   B. BY A GEOMETRIC CONFIGURATION — SPAN

6. STOP.. AFTER STUDY OF CSA AND SPAN, REVISING THE CLUSTER SELECTION BY RE-DEFINING THE SUBSETS (INCLUDING HIERARCHICAL CONDENSATION, I.E., HIGHER-ORDER ANALYSIS) IF NECESSARY AND REPEATING STEPS 4 OR 5.

C. O-ANALYSIS.. CLUSTER ANALYSIS OF INDIVIDUALS (OBJECTS)

7. SCORING INDIVIDUALS ON THE OBLIQUE DIMENSION-DEFINING CLUSTERS. — FACS

8. LINEAR AND NONLINEAR RELATING OF CLUSTER SCORES, AND 2-DIMENSIONAL O-ANALYSIS ON CORRELATION SCATTERGRAMS — RSCAT

9. SELECTING CORE O-TYPES FROM CLUSTER SCORE SPACE

   A. CALCULATING EUCLIDEAN DISTANCES BETWEEN INDIVIDUALS — EUCO

B. INTRODUCING MARKER-INDIVIDUALS IN THE CONFIGURATION OMARK

C. SELECTING CORE O-TYPES BY DIMENSIONALITY AND STRUCTURE ANALYSIS NC

10.STOP.. REVISION OF CORE O-ANALYSIS, AS IN STAGE 6.

11.DETERMINING O-CLUSTERS BY IDENTIFYING EACH INDIVIDUAL WITH A CORE O-TYPE, HOWEVER LARGE IS NS EUFIT

12.DESCRIBING THE CLUSTER SCORE PATTERN AND HOMOGENEITY OF RESULTING O-CLUSTERS.. OSTAT

13.NON-LINEAR PREDICTING OF 'OUTSIDE' ATTRIBUTES FROM THE CATEGORICAL SERIES OF O-CLUSTERS PREDICT

ORTHODOX FACTOR ANALYSIS BY BC TRY

THE LIST OF THE TYPES OF ANALYSES PERFORMED BY BC TRY USING COMPONENTS IN TANDEM INCLUDES THE TRADITIONAL SIMPLE SUM OR THURSTONIAN CENTROID FACTOR ANALYSIS AND PRINCIPAL AXES FACTOR ANALYSIS. THE FOLLOWING LIST OF STEPS ILLUSTRATES THE USE OF BC TRY IN EXECUTING THESE VARIETIES OF FACTOR ANALYSIS.

A. SIMPLE SUM FACTORING

1. THURSTONE CENTROID FOR NV TO 20.. SALIENT CENTROID FOR NV ABOVE 20 CC(CENT)

2. BIFACTOR ANALYSIS (1ST DIMENSION A CENTROID, REMAINING DIMENSIONS KEY CLUSTER) CC(CENT+)

3. SQUARE ROOT OR DIAGONAL FACTORING (OR PIVOT VARIABLE ANALYSIS) CC(PV)

B. LEAST SQUARES TOTAL-SET FACTORING AND AUXILIARIES

4. PRINCIPAL COMPONENT OR PRINCIPAL AXES FACTORING FALS(PFA)

5. CANONICAL FACTORING FALS(CFA)

6. AUGMENTED (OR 'ALPHA') FACTORING FALS(AFA)
(ALSO BY SIMPLE SUM FACTORING)

7. RESIDUALS FROM LEAST SQUARE (OR ANY OTHER) FACTORING FAST(RESID)

8. REPRODUCED CORRELATION MATRIX FROM LEAST SQUARES (OR ANY OTHER) FACTORING FAST (REPRODUCED)

9. ROTATION OF TOTAL-SET FACTORS (ANY TYPE).. VARIMAX OR QUARTIMAX GYRO

10.REGRESSION SCORES OF ORTHOGONAL OR OBLIQUE TOTAL-SET FACTORS FACS

11.COMPARISON OF TOTAL-SET FACTORS IN DIFFERENT GROUPS WITH THOSE OF A BEST-FITTING POPULATION SIMRO

SOME COMMENTS REGARDING THE TAPE VERSION OF THE BC TRY USERS' MANUAL

THIS FORM OF THE BC TRY USERS' MANUAL WAS ADOPTED FOR SEVERAL REASONS. FOREMOST AMONG THESE REASONS WAS THE GREAT DEMAND FOR MANUALS, NOT ONLY AT BERKELEY BUT FROM SCIENTISTS ELSEWHERE. AS THE REQUESTS FOR THE SYSTEM ITSELF (FOR LOCAL INSTALLATION) INCREASED IT BECAME CLEAR THAT WE NEEDED SOME WAY OF DISTRIBUTING MASTER COPIES OF THE MANUAL FROM WHICH UNLIMITED COPIES COULD BE PRODUCED. AT THE SAME TIME, CHANGES IN THE SYSTEM MADE IT CLEAR THAT PRODUCTION OF THE MANUAL HAD TO BE HIGHLY ADAPTABLE IN FORM -- SEVERAL EDITORIAL CHANGES A YEAR MIGHT BE EXPECTED. WE TRIED BOTH MIMEOGRAPH AND DITTO EDITIONS ONLY TO BE DISAPPOINTED BECAUSE OF THE DIFFICULTY OF EDITING AND UPDATING AND THE SHORTLIFE SPAN OF THE MASTERS. ALSO, THE 300 OR MORE TYPEWRITTEN PAGES OF THE MANUAL POSED SERIOUS PROBLEMS OF LOGISTICS (MATERIALS AND LABOR INVOLVED IN EACH COPY OF THE MANUAL FOR EXAMPLE).

IN RESPONSE TO THESE PROBLEMS I INITIATED AN EDITION TO BE PRODUCED BY COMPUTER PRINTER. I HAD HAD SOME EXPERIENCE WITH COMPUTER PRODUCED 'COMPUTING CENTER OPERATIONS' MANUALS AT THE UNIVERSITY OF COLORADO. THESE MANUALS WERE INITIALLY SET UP ON PUNCHED CARDS AND PRODUCED ON A CARD-LISTING MACHINE. EDITING A MANUAL CONSISTED OF REPLACING CARDS -- OUTDATED LINES AND PARAGRAPHS WERE UPDATED SIMPLY BY REPLACING OUTDATED CARDS WITH NEWLY PUNCHED CARDS.

THE CARD FORM OF A MANUAL IS CONVENIENT AT A GIVEN LOCALITY BUT TRANSPORTING A LARGE NUMBER OF CARDS (THE USERS' MANUAL INVOLVES SOME 12,000 CARDS) IS IMPRACTICAL. HENCE, I DEVISED A SYSTEM WHEREBY THE CARD FORM OF THE USERS' MANUAL WAS RECORDED ON MAGNETIC COMPUTER TAPE. THE TAPED MANUAL IS KEPT IN TWO FORMS.. THE 'OUTPUT' FORM AND THE 'EDIT' FORM. THE OUTPUT FORM IS A TAPE CONTAINING THE MANUAL IN THE FORM YOU ARE NOW READING. THE TAPE IS PLACED ON A TAPE DRIVE ATTACHED TO A PRINTER AND THE CONTENTS OF THE TAPE, THE MANUAL, ARE PRINTED. THE EDIT FORM IS SOMEWHAT CONDENSED AND EACH LINE IS SPECIALLY NUMBERED. IN ORDER TO CHANGE THE EDIT COPY AN EDIT PROGRAM IS INPUT TO THE COMPUTER ALONG WITH CARDS CONTAINING INSTRUCTIONS REGARDING LINES OF THE MANUAL TO BE DELETED AND LINES TO BE INSERTED. FROM AN UPDATED EDIT TAPE AN UPDATED OUTPUT TAPE IS MADE.

A COMPUTER CENTER INSTALLING THE BC TRY SYSTEM CAN OBTAIN THE MANUAL ON EITHER THE OUTPUT TAPE OR THE EDIT TAPE. THE PROGRAMS REQUIRED TO UPDATE THE EDIT TAPE AND TO PRODUCE AN OUTPUT TAPE FROM THE EDIT TAPE ALSO ARE PROVIDED.

THE RECASTING OF THE MANUAL, FROM THE DITTOED VERSION TO THE COMPUTER TAPE VERSION WAS A MORE MASSIVE TASK THAN I HAD IMAGINED IT WOULD BE. THE KEYPUNCH IS A LIMITED TYPOGRAPHICAL INSTRUMENT AND THE DISCRETE CHARACTER OF EACH LINE, A SEPARATE CARD, MADE LAYOUT A DIFFICULT JOB. HAVING MOVED FROM CARDS TO TAPE (ONCE ALL THE CARDS WERE PREPARED) ADDITIONAL PROBLEMS WERE ENCOUNTERED. MOST TROUBLESOME OF THESE PROBLEMS WAS THE TIGHT ORGANIC CHARACTER OF THE TAPE. DELETING OR ADDING A LINE COULD CAUSE A CORRESPONDING SHIFT OF THE LAYOUT OF MANUAL PAGES BEYOND THE PAGE MODIFIED. A MORE SOPHISTICATED EDITOR PROGRAM IS PLANNED THAT WILL ALLEVIATE THIS PROBLEM AND OTHER PROBLEMS ENCOUNTERED WITH THE TAPE FORM OF THE MANUAL.

PHYSICAL ASPECTS OF THE BC TRY SYSTEM

TO USE THE BC TRY SYSTEM WITH EXPERTISE, AN ANALYST REQUIRES ONLY THE MOST GENERAL UNDERSTANDING OF THE PHYSICAL NATURE OF THE COMPUTER AND OF THE WAY IT FUNCTIONS IN IMPLEMENTING THE PROGRAMS ON HIS DATA. THIS SECTION IS INTENDED TO PROVIDE THIS MINIMAL KNOWLEDGE.

THIS SECTION GIVES A GENERAL DESCRIPTION OF THE PHYSICAL ASPECTS.. (1) THE CONFIGURATION OF STORAGE UNITS (TAPES, DISCS) THAT ARE EMPLOYED AS DEPOSITORIES OF DATA, PROGRAMS AND RESULTS, AND (2) THE INTERRELATED FUNCTIONING OF THESE ELEMENTS AS THEY OPERATE TO SELECT AND PERFORM PARTICULAR TYPES OF ANALYSES. THE NEXT SECTION IS A DESCRIPTION OF THE EXECUTIVE OPERATIONS ON THE MOST CRUCIAL STORAGE UNIT, THE SYSTEM TAPE (OR DISC)--THE UNIT ON WHICH THE COMPONENT PROGRAMS OF THE BC TRY SYSTEM ARE DEPOSITED. EXECUTIVE CONTROL IS ACHIEVED BY THE GENERAL EXECUTIVE PROGRAM (GEP). THIS PROGRAM INITIATES AND TERMINATES THE WORK, BUT MORE IMPORTANT, IT SELECTS, TURNS CONTROL OVER TO, AND ACCEPTS CONTROL FROM THE COMPONENT PROGRAMS ON THE SYSTEM TAPE THAT PERFORM THE DESIRED WORK. BECAUSE EACH PROGRAM MUST PERFORM ITS PARTICULAR WORK ON SOME TYPE OF INPUT DATA AND STORE ITS FINDINGS SOMEPLACE, THESE ARE THE IST (INTERMEDIATE STORAGE TAPE) AND THE DST (DATA STORAGE TAPE). THEY ARE CALLED 'SHARED DATA TAPES' BECAUSE DATA AND RESULTS ARE STORED ON THEM AS COMMON DEPOSITORIES THAT CAN BE UTILIZED BY THE DIFFERENT COMPONENT PROGRAMS OF THE SYSTEM.

THE PHYSICAL CONFIGURATION OF STORAGE UNITS AND THE COORDINATED FUNCTIONING OF THEM ARE DESIGNED TO PERMIT THE ANALYST TO PERFORM ANY PARTICULAR KIND OF CLUSTER OR FACTOR ANALYSIS THAT SUITS HIM. THE DESIGN WAS CREATED IN ORDER TO PROVIDE THE ANALYST WITH A CHOICE OF METHODS AT EACH OF THE MAIN STAGES OF A MULTIDIMENSIONAL ANALYSIS, AND TO PERMIT HIM TO START, TO INTERRUPT, OR FINALLY TO TERMINATE HIS ANALYSIS AT ANY STAGE. THE STORAGE UNITS AND THE MATERIAL ON THEM ARE ORGANIZED IN A FASHION THAT ENABLES THE USER TO MAKE CHOICES AND TO START, INTERRUPT, OR TERMINATE AT ANY ONE OF THE SEVEN 'REGIONS OF DECISION' IN THE PROGRESSIVE STAGES OF A FULL SCALE MULTI-DIMENSIONAL ANALYSIS (TRYON, 1959).

BASIC STORAGE UNITS.

THE ESSENTIAL INPUT, OUTPUT, AND STORAGE UNITS USED BY THE SYSTEM ARE AS FOLLOWS.

THE MONITOR INPUT TAPE CONTAINS ALL USER INITIATED INPUT (I.E., ALL CONTROL AND DATA CARDS).

THE BC TRY SYSTEM TAPE CONTAINS THE PROGRAMS WHICH PERFORM THE FACTOR OR CLUSTER ANALYSES. THUS THIS TAPE CONTAINS (1) A SET OF ''COMPONENT'' PROGRAMS CORRESPONDING ROUGHLY TO THE SEVEN STAGES DESCRIBED ABOVE, AND (2) AN EXECUTIVE PROGRAM WHICH PUTS THESE COMPONENTS INTO OPERATION.

THE BC TRY SHARED DATA TAPES CONTAIN ALL INFORMATION THAT IS OUTPUT BY SOME COMPONENT PROGRAMS TO BE USED FOR FURTHER ANALYSIS BY OTHER COMPONENT PROGRAMS. THE DATA STORAGE TAPE (DST) STORES THE RAW DATA FOR ANY ANALYSIS, I.E., THE RAW OBSERVATIONS ON THE VARIABLES OR OBJECTS WHICH ARE TO BE CORRELATED AND EVENTUALLY ANALYZED BY THE COMPONENTS. THE INTERMEDIATE STORAGE TAPE (IST) STORES INTERMEDIATE RESULTS OF ANY ANALYSIS. THAT IS, THIS TAPE STORES THE RESULTS OF EACH OF THE STAGES DESCRIBED ABOVE (1) THE CORRELATION MATRIX, (2) COMMUNALITY ESTIMATES, (3) FACTOR COEFFICIENTS, ETC.

THE MONITOR OUTPUT TAPES CONTAIN ALL PRINTED AND PUNCHED OUTPUT AVAILABLE TO THE USER.

BASIC STAGES OF OPERATION.

THE NORMAL OPERATION OF THE SYSTEM MAY BE BROKEN DOWN INTO TWO LEVELS.. (A) MONITOR OPERATIONS, WHICH CONSIST PRIMARILY OF MONITOR CALLING THE BC TRY EXECUTIVE PROGRAM INTO OPERATION, AND (B) BC TRY OPERATIONS, WHICH CONSIST OF THE INITIATION STAGE, THE COMPONENT SELECTION STAGE, IN WHICH THE EXECUTIVE CALLS A SERIES OF COMPONENT PROGRAMS INTO OPERATION, AND THE TERMINATION STAGE, IN WHICH CONTROL IS RETURNED TO MONITOR.

MONITOR ACCESS. AFTER PERFORMING THE USUAL JOB INITIATION OPERATIONS, MONITOR MUST BE INSTRUCTED TO READ THE BC TRY EXECUTIVE PROGRAM FROM THE SYSTEM TAPE AND TURN CONTROL OVER TO IT. HOW THIS THIS ACCESSING IS ACCOMPLISHED WILL DEPEND ON THE INDIVIDUAL COMPUTER CENTER PROCEDURES.

BC TRY OPERATIONS. THE USER COMMUNICATES HIS COMMANDS TO THE BC TRY EXECUTIVE PROGRAM BY MEANS OF SPECIAL EXECUTIVE CONTROL (E.C.) CARDS. A PROPERLY SET UP DECK OF THESE CARDS YIELDS THE THREE STAGES DESCRIBED BELOW. ONCE THE EXECUTIVE IS PLACED IN CONTROL IT CONTINUES READING AND EXECUTING E.C. CARDS UNTIL IT ENCOUNTERS ONE REQUESTING TERMINATION.

(1) INITIATION. THIS STAGE CONSISTS OF SUCH THINGS AS REWINDING TAPES AND SETTING UP THE INTERMEDIATE STORAGE TAPE (IST).

(2) COMPONENT SELECTION. DURING THIS STAGE THE EXECUTIVE PROGRAM CALLS A SERIES OF COMPONENT PROGRAMS INTO OPERATION. THE USER SPECIFIES THE PARTICULAR PROGRAMS TO BE CALLED, AND THE ORDER IN WHICH THEY ARE TO BE CALLED BY MEANS OF E.C. CARDS. THUS, WHEN THE EXECUTIVE ENCOUNTERS AN E.C. CARD REQUESTING A PARTICULAR COMPONENT, IT READS THE COMPONENT FROM THE SYSTEM TAPE AND TURNS CONTROL OVER TO THE COMPONENT. THE COMPONENT IN TURN (A) READS ITS OWN COMPONENT CONTROL CARDS AND DATA CARDS OR A SHARED DATA TAPE, (B) PERFORMS ITS CALCULATIONS, (C) WRITES ITS PRINTED, PUNCHED, AND SHARED DATA TAPE OUTPUT, AND (D) RETURNS CONTROL TO THE EXECUTIVE PROGRAM. SINCE THE EXECUTIVE CONTINUES READING E.C. CARDS, THE PROCESS OF COMPONENT SELECTION MAY BE REPEATED

UNTIL THE COMPLETE ANALYSIS HAS BEEN PERFORMED. EACH TIME CONTROL IS RETURNED TO THE EXECUTIVE, ONE MORE CONTROL CARD IS READ. THE SHIFT OF CONTROL FROM EXECUTIVE TO COMPONENT PROGRAM AND BACK CONTINUES UNTIL A CARD IS READ WHICH CALLS FOR TERMINATION.

(3) TERMINATION. WHEN THE EXECUTIVE PROGRAM ENCOUNTERS AN E.C. CARD REQUESTING TERMINATION, IT ''NEATENS UP'' (I.E., REWINDS TAPES, ETC.), AND RETURNS CONTROL TO MONITOR.

EXECUTIVE OPERATIONS

THE PURPOSE OF THIS SECTION IS TO PROVIDE THE USER WITH AN OVERALL UNDERSTANDING OF THE EXECUTIVE OPERATIONS, THAT IS, OF THOSE OPERATIONS WHICH PRIMARILY INVOLVE THE STORAGE AND RETRIEVAL OF INFORMATION OF THE BC TRY SYSTEM TAPE. THE SYSTEM TAPE CONTAINS PROGRAMS, I.E., SETS OF MACHINE LANGUAGE INSTRUCTIONS, ALL READY TO BE PUT INTO OPERATION. IN THIS SECTION WE SHALL FIRST DESCRIBE BRIEFLY THESE PROGRAMS ON THE SYSTEM TAPE, EXPLAIN GENERALLY HOW THE PROGRAMS ARE STORED ON THE TAPE, AND, FINALLY, INDICATE HOW THEY ARE READ FROM THE TAPE AND PUT INTO OPERATION.

IT SHOULD BE CLEAR THAT THE USER NEED NOT UNDERSTAND TECHNICAL DETAILS. TO SUCCESSFULLY USE THE BC TRY SYSTEM ALL HE NEEDS TO DO IS FOLLOW THE INSTRUCTIONS PRESENTED IN THE SECTIONS OF THIS MANUAL DESCRIBING THE USE OF EACH COMPONENT.

THE SYSTEM TAPE.

THE FOLLOWING PROGRAMS ARE PRESENTLY AVAILABLE ON THE BC TRY SYSTEM TAPE. IN THIS PRESENTATION THEY ARE ORGANIZED ACCORDING TO THEIR PRINCIPAL FUNCTIONS, I.E., EXECUTIVE, DATA SHARING, AND STATISTICAL (STANALOG) PROGRAMS.

THE DESCRIPTIVE TITLE OF EACH PROGRAM IS FOLLOWED BY THE CODE NAME OF THE PROGRAM, E.G., '' THE GENERAL EXECUTIVE PROGRAM.. GEP.'' A ''BC'' PREFIX IS OFTEN ADDED TO THE CODE NAME TO INDICATE THAT THE PROGRAM WAS WRITTEN ON THE BERKELEY CAMPUS OF THE UNIVERSITY OF CALIFORNIA. SINCE ALL OF THE FOLLOWING WERE WRITTEN ON THE BERKELEY CAMPUS THE BC PREFIX IS USUALLY DROPPED IN THE TITLES ON COMPONENT PROGRAMS.

PROGRAMS SERVICING THE SYSTEM TAPE. THE GENERAL EXECUTIVE PROGRAM, GEP, PERFORMS CERTAIN INITIAL AND TERMINAL OPERATIONS AND READS COMPONENT PROGRAMS FROM THE SYSTEM TAPE AND PLACES THEM IN OPERATION.

PROGRAMS SERVICING THE SHARED DATA TAPES, IST AND DST. THE IST RESTART PROGRAM, GIVE (TAKE), SAVES THE INTERMEDIATE STORAGE TAPE (IST) ON CARDS FOR LATER ANALYSIS. THE IST GENERATING PROGRAM, GIST, TRANSLATES INDIVIDUAL IST FILES TO AND FROM CARDS. THE DATA PROCESSING PROGRAM, DAP, PREPARES THE DATA STORAGE TAPE (DST) FROM CARDS.

STANALOG PROGRAMS PERFORMING PRIMARILY STATISTICAL ANALYSES, AS LISTED IN THE INTRODUCTION. EACH OF THE PROGRAMS IS STORED AS A SEPARATE BLOCK OF SYMBOLS ON THE SYSTEM TAPE. THE PROGRAMS ARE STORED SEQUENTIALLY. THE PROGRAMS ARE LOCATED SYMBOLICALLY, I.E., BY MEANS OF CODE NAMES AS EXPLAINED IN THE FOLLOWING SECTION.

RETRIEVAL OF PROGRAMS STORED ON THE SYSTEM TAPE.

THE PROCESS OF CALLING PROGRAMS FROM THE SYSTEM TAPE AND PLACING THEM IN OPERATION IS THE PRIMARY FUNCTION OF THE GENERAL EXECUTIVE PROGRAM (GEP). THE EXACT PROCEDURES FOR USING THE GEP ARE FULLY DESCRIBED BELOW.

STORING PROGRAMS ON THE BC TRY SYSTEM TAPE.

PROPERLY SPEADING THE PROCESS OF STORING PROGRAMS ON THE SYSTEMS TAPE FOR LATER USE IS THE TASK OF THE PROGRAMMER, NOT THE USER. HENCE, WE SHALL MERELY POINT OUT SOME OF THE BASIC OPERATIONS WHICH MAY BE PERFORMED AND REFER INTERESTED READERS TO ''EXECUTIVE OPERATIONS AND DATA SHARING'' OF THE PROGRAMMER'S MANUAL. THESE ARE.. (A) AN ENTIRE SYSTEM TAPE MAY BE CONSTRUCTED FROM BINARY CARDS BY SIMPLY USING ''* CHAIN'' FORTRAN II MONITOR CONTROL CARDS, (B) INDIVIDUAL PROGRAMS MAY BE REPLACED BY USING THE EDITING PROGRAM, EDIT, (3) PROGRAMS NOT PRESENT IN THE BC TRY SYSTEM MAY ALSO BE ADDED BY USING THE EDITING PROGRAM, AND (D) PROGRAMS COMPATIBLE WITH THE SYSTEM BUT NOT ON THE SYSTEM TAPE MAY BE RUN USING A SPECIAL FEATURE OF THE EXECUTIVE PROGRAM, TEST.

STORAGE AND RETRIEVAL OF SHARED DATA

THE SECOND MAJOR SET OF STORAGE UNITS IN THE BC TRY SYSTEM ARE THE SHARED DATA TAPES. THE SYSTEM USES TWO SUCH TAPES, THE DATA STORAGE TAPE (DST) AND THE INTERMEDIATE STORAGE TAPE (IST). THE DST IS USED TO STORE RAW DATA ONLY, I.E., THE RAW OBSERVATIONS ON THE VARIABLES WHICH ARE TO BE CORRELATED BY MEANS OF COR2 OR COR3 OR USED BY OTHER COMPONENTS (E.G., FACS). IST IS THE PRINCIPAL RESPOSITORY OF PROCESSED DATA SHARED AMONG COMPONENT PROGRAMS. IN THE FOLLOWING SECTIONS THE STORAGE AND RETRIEVAL OF INFORMATION ON THESE TAPES IS DESCRIBED. IN ADDITION SHARED DATA OUTPUT AND INPUT ARE LISTED IN THE USER'S DESCRIPTIONS OF THE COMPONENT PROGRAMS. THE PURPOSE OF THIS SECTION IS TO PRESENT SOME GENERAL CONCEPTS NECESSARY FOR THE USE OF THE SHARED DATA CHARACTERISTICS OF BC TRY.

THE INTERMEDIATE STORAGE TAPE (IST).

THE IST FILES. DATA ARE WRITTEN ON A TAPE IN SETS CALLED ''FILES.'' THE COMPUTER USES A SPECIAL CHARACTER CALLED AN ''END OF FILE MARK'' TO DETERMINE WHEN ONE FILE ENDS AND ANOTHER ONE BEGINS. A SPECIFIC SET OF DATA IS LOCATED BY KEEPING TRACK OF WHERE ITS FILE IS LOCATED ON THE TAPE. THUS INFORMATION MAY BE LOCATED ON A TAPE BY EITHER COUNTING FILES OR BY WRITING A SPECIAL LABEL IN FRONT OF THE DATA AND SEARCHING FOR THIS LABEL. BC TRY USES AN IMPROVED VERSION OF THE LATTER PROCEDURE TO LOCATE INFORMATION ON IST. ALL OF WHICH LEADS US TO THE MAIN POINT, NAMELY THAT EACH TYPE OF DATA, I.E., MEANS, CORRELATION MATRICES, ETC., IS, IN ESSENCE, ASSIGNED ITS OWN

UNIQUELY NAMED IST FILE. THE TYPES OF DATA STORED ON IST AND THE NAME OF THE FILE WHERE EACH TYPE OF DATA IS STORED ARE..

CONTENTS OF IST FILE	NAME OF IST FILE
MEANS	MEANS1
STANDARD DEVIATIONS	STDEV1
CORRELATIONS	CORRM1
CORRELATIONS, UNDELETED MATRIX	CORRM2
LIST OF REORDERED AND DELETED VARIABLES	REORD1
DIAGONAL VALUES	DIAGV1
UNAUGMENTED FACTOR COEFFICIENTS	UFACT1
ROTATED FACTOR COEFFICIENTS	RFACT1
CLUSTER INDICATORS	CLUST1
REFLECTION INDICATORS	REFLX1
FACTOR SCORES	FSCOR1
BASIS OF ROTATED FACTOR SPACE	BASIS1
MEAN OF FACTOR SCORES	MEANS2
STANDARD DEVIATIONS OF FACTOR SCORES	STDEV2
WEIGHTS FOR FACTOR SCORES	WEIGH1
NAMES OF VARIABLES	VNAMS1
SUMS OF VARIABLES	VSUMS1
SAMPLE SIZES FOR THE VARIABLES	VAREN1
COVARIANCES	COVAR1
IDENTIFICATION OF DATA	IDFILE
MATCHING CONSTANTS, ALL PAIRS OF VARIABLES	
SAMPLE SIZES	2SENS1
MEANS	2MEANS1
STANDARD DEVIATIONS	2STDEV1
NAMES OF OBJECTS (I.E., SUBJECTS)	ONAMS1

SEVERAL POINTS SHOULD BE NOTED. FIRST, THE USER NEED NOT WORRY OVER UNDERSTANDING HOW IST WORKS. TO USE IST ALL HE NEEDS TO DO IS FOLLOW THE DIRECTIONS GIVEN IN THE USER'S DESCRIPTIONS. SECOND, WHEN THE CONTENTS OF A PARTICULAR FILE ARE READ FROM IST BY A COMPONENT PROGRAM, THE CONTENTS ARE NOT DESTROYED. THAT IS, THE DATA IN AN IST FILE AFTER IT HAS BEEN READ BY A COMPONENT ARE IDENTICAL TO THE DATA IN THE FILE BEFORE THE READING.

WHEN, HOWEVER, A COMPONENT PROGRAM COMPUTES AND WRITES A SET OF DATA IN A PARTICULAR IST FILE, THE PRIOR CONTENTS OF THAT FILE ARE REPLACED BY THE NEW DATA. THAT IS, WRITING AN IST FILE DESTROYS THE PRIOR CONTENTS OF THAT FILE. NEITHER READING NOR WRITING A PARTICULAR IST FILE HAS ANY EFFECT ON THE OTHER IST FILES. FINALLY, THE OCCURRENCE OF EITHER A ''/START'' OR AN ''/END'' EXECUTIVE CONTROL CARD EFFECTIVELY DESTROYS THE ENTIRE IST TAPE. THE CONTENTS OF ANY IST MAY, HOWEVER, BE PRESERVED ON IST RESTART CARDS, AS DESCRIBED BELOW.

STORAGE AND RETRIEVAL OF IST FILES BY THE USER. THE FLEXIBILITY OF ANY SYSTEM USING SHARED DATA STORAGE IS GREATLY INCREASED WHEN THE USER IS PERMITTED TO COMMUNICATE DIRECTLY WITH THE SHARED DATA, THAT IS, WHEN HE CAN PLACE DATA DIRECTLY IN SHARED STORAGE, ALTER INFORMA-

TION ALREADY THERE, SAVE THE SHARED DATA FOR LATER ANALYSIS, ETC. TWO BC TRY COMPONENT PROGRAMS ARE DESIGNED SPECIFICALLY TO PERMIT THE USER TO COMMUNICATE WITH IST VIA PUNCHED CARDS.. ONE OF THESE PROGRAMS, THE IST RESTART PROGRAM, GIVE, PERMITS THE USER TO SAVE IST FILES ON CARDS FOR FURTHER ANALYSIS AT A LATER TIME. THE OTHER PROGRAM, THE IST GENERATING PROGRAM, GIST, PERMITS THE USER COMPLETE CONTROL OVER THE CONTENTS OF ANY OR ALL IST FILES. THE USER'S DESCRIPTIONS OF THESE PROGRAMS ARE PRESENTED BELOW.

STORAGE AND RETRIEVAL OF IST FILES BY COMPONENT PROGRAMS. ALL COMPONENTS READ AND/OR WRITE SOME IST FILES. EXPERIENCE HAS SHOWN THAT IT IS VALUABLE TO HAVE IN ONE PLACE A COMPLETE LIST OF ALL IST FILES INPUT AND/OR OUTPUT BY ALL COMPONENT PROGRAMS. SUCH A COMPLETE SUMMARY OF ALL IST INPUT AND OUTPUT IS GIVEN IN TABULAR FORM IN THE SECTION TITLED IST I/O AT THE END OF THIS MANUAL.

DATA STORAGE TAPE (DST).

THE DST FILES. THE INFORMATION WRITTEN ON THE DATA STORAGE TAPE IS NOT LOCATED BY FILE NAMES BUT BY THE ORDINAL POSITION OF THE DATA. THUS THE FIRST FILE ON THE TAPE CONTAINS AN IDENTIFYING TITLE FOR THE DATA, THE NUMBER OF VARIABLES AND AN INDICATOR FOR MISSING DATA. THE SECOND FILE CONTAINS THE RAW SCORES ON THE VARIABLES WHICH ARE TO BE CORRELATED. THE EXACT FORMAT OF THIS TAPE IS GIVEN IN THE PROGRAMMER'S MANUAL. TO OPERATE DST, THE USER NEED ONLY FOLLOW THE DIRECTIONS PRESENTED IN THE USER'S DESCRIPTIONS, HE NEED NOT CONCERN HIMSELF WITH UNDERSTANDING HOW THE TAPE OPERATES.

STORAGE AND RETRIEVAL OF DST FILES BY THE USER. THE PROCEDURE FOR INPUTTING RAW SCORES TO DST IS DESCRIBED IN THE USER'S DESCRIPTION OF THE DATA PROCESSOR PROGRAM, DAP. THIS IS LIMITED TO MAKING DST FROM CARDS, AND PERFORMING SOME ELEMENTARY SORTING OPERATIONS ON THE VARIABLES. THERE ARE NO PROVISIONS AT PRESENT FOR ANY OTHER TYPE OF USER-INITIATED COMMUNICATION WITH DST.

General rules for the use of the BCTRY programs

This chapter contains a miscellany of information most of which is crucial for the use of the BCTRY system. The sections on 'notational conventions' and 'types of cards' are prerequisite to the understanding of later chapters and should be read before going further. Although the introduction to the section, 'Key phrases : input-output conventions,' should also be read before continuing, the detailed subsections can be put aside for later reference. The sections, 'General restrictions' and 'Error termination,' may be skipped until the analyst begins the mundane task of actually running data. The 'Error termination' section may, in theory, be skipped altogether but is in practice a very popular section.

Notational conventions

Several kinds of expressions are needed so continually throughout this Manual that special symbols have been invented to stand for more complete statements. These symbols are given precise definition in this chapter and are used consistently in the remaining pages.

One of the most frequently encountered problems in writing the Manual was that of separating characters to be punched on cards from symbols which stand for numerical values to be punched. The convention has been adapted always to place characters to be punched within quotation marks. Thus the instructions to punch '/DAP' in columns 1-4 mean to punch each character between the quotes. Instructions to punch N (without quotes) in column 1-4, however, mean to punch the numerical value of N, not the symbol N.

The only exception to this rule is the plus sign, +. A blank column is always indicated by a plus sign +. Instructions to punch '/DAP+++' in columns 1-7 thefore mean to leave three blank columns after '/DAP'. The plus sign + is the only character not actually to be punched which occurs within quotation marks.

Two other notation conventions are those of the standard options and the key phrases. The standard options are described in the section immediately following.

Types of cards read

The cards read by the system may be divided into three categories: (1) executive control cards, which control the operation of the executive program, (2) component control cards, which control the operation of component programs, and (3) data cards, which provide the bulk of the information to be processed by the component programs. Certain punching conventions that apply to each type of card are discussed in this section.

Executive control (E.C.) Cards. Although the contents of E.C. cards will depend upon the operation desired (see user's descriptions, below), they are always punched with a slash in column 1 and the code name of the desired component punched with trailing blanks in columns 2-7, as in '/COR2++', '/DVP+++'. Columns 8-80 are ignored by the program and may be used for identification. The function of the E.C. card is in almost all cases to call a component program.

component control (C.C.) Cards. The component program called by the E.C. card usually requires information about methods and parameters to be used. The function of the C.C. card is to provide such information.

Although the rules for C.C. cards are specific to the particular component program which reads them, one convention is observed in all component programs: the 'standard' convention. This is a convention which relieves the analyst from the necessity of making a choice from the options allowed by the program and from the necessity of punching all the control cards required. For most options allowed by the program, one particular set of options is optimal for most purposes. This set has been labeled 'standard' in the program descriptions. A standard choice can be obtained by leaving the prescribed field blank. Through the use of this convention, control cards will often be blank. The cards, must, however, be present in the control deck.

The reader will find each control card for a particular program completely described in the user's description for that component program. The format for punching the control cards, however, will usually be more general and will be described by the use of key phrases (see below).

Data cards. Certain component programs called by an E.C. card require input of alphanumeric letters and numbers or numeric information. Parameters of these data cards are usually punched on C.C. cards.

The rules for inputting data cards are in nearly all cases specific to the type of data rather than to the component program which reads the data. Because of this fact the program descriptions give only the briefest description of data cards through the use of key phrases. In other words, description of data cards is provided primarily in the next section on key phrases.

Key phrases: input-output conventions

The user's descriptions of component programs below entail discussion of different types of information to be input or output on cards. If each type of information were punched in a unique format, or if card punching were completely described in each user's description, the use of this Manual and of the system would be tedious and error-prone. To meet this problem, a set of standard conventions is adopted.

This section describes the card formats and the array formats which are used by the system. A card format is a set of rules for punching one item of information or a series of items; an array format is a set of rules for punching a matrix or a row of a matrix (a vector).

All formats are defined precisely in this section and are given labels called 'key phrases.' In the following chapters of the Manual, cards will be described via key phrases and usually not otherwise. For example, the specifications for the input of factor coefficients consists only of a sentence identifying 'F' as a K by N matrix of factor coefficients and the phrase 'punch F as a matrix on decimal cards.' Through the use of the key phrase convention the reader has been spared pages of tedious and redundant detail.

KEY PHRASE	CARD FORMAT SECTION BELOW
title card	(1)
fixed field name card	(2)
free field name card	(3)
decimal card	(4)
integer card	(5)
variable format card	(6)
punch as a vector	(7)
punch as a matrix	(8)

Card formats. The symbols in parentheses printed beside each key phrase below are FORTRAN format statements defining the cards under discussion. This notation is intended as an aid to readers familiar with FORTRAN and is in no way necessary for understanding the key phrase definitions.

(1.) Title card format (12A6). Instructions to punch a title card means to punch an identifying label consisting of any combination of alphabetic or numeric characters, beginning in column 1 and ending in column 72 of the card. This information is usually used as an identifying label for other printed output.

(2.) Fixed field name cards (9(A6,2X)). Instructions to punch a list of names on fixed field name cards mean that the names should be punched in successive 8 column fields, beginning in column 1 and ending in column 72, with a maximum of 9 names per card. A fixed field consists of exactly six alphabetic or numeric characters. Within a field the name should be 'packed left,' That is, the first six columns from the left should contain the name and the last two columns in the 8 column field should be blank. For example, one might punch the following as a list of three variable-names: 'VAR001++VAR002++VAR003'. The list of names may be continued on successive name cards, leaving the unused fields of the last card blank.

(3.) Free field name cards. Because lists of names are commonly input, most components of BCTRY use the more convenient free field format. Instructions to punch a list of names on free field name cards mean the following. Here, a 'name' consists of from one to six alphabetic or numeric characters. The names are formed into a list by separating the names by commas and terminating them with a period. Punch this list in columns 1-72 on as many cards as required. Do not split a name between two cards. Within a field, blanks are ignored by the computer as are any name-characters after the first six. For example, one might punch the following as a list of three variable names: '1STVARIABLE,2NDVARIABLE,3RDVARIABLE.' In this case the names read would be the first six characters in each field, viz. '1STVAR, 2NDVAR, 3RDVAR'.

(4.) Decimal card format. (9F8.4). Numbers input to the BC TRY system are placed in two categories: rational numbers and integers. The essential difference is that rational numbers sometimes include a fractional or decimal part, while integers do not. Thus two different standard card formats are provided, a decimal format for rational numbers and an integer format for integers. Instructions to punch a set of numbers on 'decimal cards' mean that the numbers should be punched in successive eight column fields, beginning in column 1 and ending in column 72, with a maximum of nine numbers per card. Within a field, the indicated position of the decimal point is between the fourth and fifth columns of the eight column field. This indicated position may be overridden by punching a decimal point in any one of the eight columns, according to FORTRAN rules. For example, a communality might be punched '.7654321' instead of '++++7654' to obtain seven digit accuracy. As seen in the example, a number is always interpreted as positive unless it is preceded by a minus sign punched within its eight column field. The set of numbers may be continued on successive decimal cards. use as many cards as necessary. The unused fields of the last card are left blank.

(5.) Integer card format. (18I4). Instructions to punch a set of numbers on integer cards mean that the numbers should be punched in successive four column fields, beginning in column 1 and ending in column 72, with a maximum of 18 numbers per card. Within a field the right hand column is the unit's place, the second from the right is the ten's place, etc. In other words, the numbers are 'packed right.' The integers 1, 10, 100, 1000 may be punched '+++1', '++10', '+100', '1000'. As seen in the example, a number is always interpreted as a positive number unless it is preceded by a minus sign punched within its four column field. Never punch a decimal point in an integer card field. The set of integers may be continued on successive integer cards. The unused fields of the last card are left blank.

(6.) Variable format cards. Variable format cards, where used, are always optional. That is, programs which read card data always provide a 'standard' format which the analyst may use and thereby avoid the use of variable format cards. Moreover, control cards, alphanumeric data cards, and integer data cards must always be punched in the particular format specified by the program. The standard format for decimal data cards, however, is often inconvenient for the analyst. The field width may be wider than that needed, or the cards may have already been punched in another format. In such a case - and when permitted by the component program - the analyst may want to state his own format through the use of a variable format card. A variable format card consists of one or more cards on which is punched a legal FORTRAN format statement describing completely the vector to be input or the longest row vector of the matrix to be input (see sections 7 and 8 below and the library version of this Manual on requisite legal F and X format statements). Use only the first 72 columns, but the data cards described by the statement may utilize all 80 columns. Do not punch the word 'format', unless stated differently in the component program description; simply begin with the left parenthesis in column one and end with the right parenthesis.

(7.) Punch as a vector. A vector is an ordered series of elements. Subscripts are often used to denote the ordinal position of the element on the vector. Thus X = (X1, X2, X3,..., XK) is a vector containing K elements. An example of a vector which the analyst might wish to punch is the vector of variable means, i.e., M1,M2,..., MN. In the User's Manual the phrase 'punch X as a vector' means that the successive elements of X are punched into successive fields of successive cards until the last element XK is punched. If any fields remain unused on the last card they are left blank. Thus the phrase 'punch X as a vector on integer cards' would mean that (X1,X2,X3,...,X18) would be punched in columns 1-4, 5-8, 9-12, ..., 69-72 of the first card, (X19,X20,...,X36) In columns 1-4, 5-8, 9-12, ..., 69-72 of the second card, and so on. A similar interpretation would be given the phrase 'punch X as a vector on decimal cards.' thus (X1, X2, ..., X9) would be punched in columns 1-8, 9-16, ..., 65-72 of the first card, (X10,X11, ..., X18) would be punched in columns 1-8, 9-16, ..., 65-72 of the second card, and so on.

(8.) Punch as a matrix. A matrix is a rectangular array of elements. Subscripts are used to denote the ordinal positions of elements in rows and columns. Hence a matrix may be considered an ordered set of vectors, e.g.,

```
     (X11 X12 X13 ... X1C)
     (X21 X22 X23 ... X2C)
X =  (... ... ... ... ...)
     (XR1 XR2 XR3 ... XRC)
```

Thus, in the above, X is a matrix containing R rows and C columns, and XIJ denotes the element in the J'th column of the I'th row. In the BCTRY system, matrices are input as if each row of the matrix were a vector. Thus in the User's Manual the phrase 'punch X as a matrix' means that each row of X is is 'punched as a vector' on successive cards. Thus each row of X begins in column one of a new card, and is punched onto successive cards, leaving blank the unused fields of the last card used by the row. As in the case of vectors, the latter part of phrases like 'punch as a matrix on decimal cards' defines the field width and the way in which the elements are to be punched within any field.

General restrictions on BCTRY programs

The maximum allowable values of parameters are given below. Any exceptions or additions to these restrictions are indicated in the user's descriptions of the component programs.

number of variables	90
number of dimensions (factors)	15
number of defining variables in a key cluster	20
number of subjects	10,000

Error termination

Errors causing the termination of BCTRY execution may be classified into four basic types: (1) errors anticipated by the BCTRY system, (2) errors anticipated by FORTRAN II, (3) errors anticipated by the particular computer installation, and (4) errors which nobody anticipated. The last two types of errors do not generally yield IST restart decks (see user's description of GIVE) and should be taken up with computer installation staff. Termination due to exceeding time or page estimates, and program stops are typical examples of the last two types of errors. Error type (1) and type (2) yield IST restart decks and sufficient printed information to permit the user to locate the error himself.

The errors anticipated by BCTRY yield fairly complete printed messages describing the error and IST restart decks. Most of the commonly made errors are anticipated by BCTRY, e.g., mispunching control cards, failing to place necessary files on IST (see GIST), exceeding maximum values of parameters, etc.

The errors anticipated by FORTRAN II yield printed 'error codes' and IST restart decks. The meanings of the most common error codes are given below.

Statements below are based on FORTRAN II Operations Manual as applied to BCTRY.

CODE	MEANING
HPR 0,1	A format specification is incorrect. It is probably one which is punched on an input card.
HPR 1,1	A character on an input card is inappropriate (such as an alphabetic character appearing where a number is expected), or an attempt was made to print a 'character' in memory which was actually not a legal character at all. In the former case it is definitely a user error; perhaps the input deck is out of order. In the latter case the program is at fault.
HPR 2,1	Same as HPR 1,1
HPR 3,1	Same as HPR 1,1
HPR 4,1	Same as HPR 1,1
HPR 0,3	There was a machine failure while reading a tape. Resubmit job after the machine has been repaired.
HPR 1,3	The information contained on a tape was lost or could not be reliably read. Resubmit job, in the hope that different tapes will be in use.
HPR 2,3	An attempt was made to read more cards than are in the input deck. Check the input setup.
HPR 0,5	Same as HPR 0,3 except it occurred while writing a tape.
HPR 1,5	A tape could not be written on because it contained a defect. Resubmit job.
HPR 2,5	Same as HPR 1,5
HPR 3,5	A tape was written until full and could not contain more information. Perhaps this is due to an enormous amount of raw data (several hundred thousand numbers). Or it is an excessively long run calling on many component programs. Break it up into smaller runs by using GIVE and TAKE.
HPR 0,6	The program referred to a non-existent input-output unit. Program error.

An HPR 1,1 stop for an inappropriate character on a card is by far the most common error termination. The best way to check for this type of error is to print on a listing machine the entire input deck. Although the problem may be with one of the control card, the usual error consists of a non-numeric character being punched somewhere in the data field. Occasionally such an error will be difficult to detect visually (e.g., O's instead of zeros, hyphens instead of minus signs), but usually the error is simply a stray letter where a number should be. To find the error the analyst may simply scan the printed listing of data cards. Another check which should be made is to count the number of data cards. If the data deck is one card short the following control card will be read as a data card. At least seventy-five percent of BCTRY errors can be prevented by making these checks before the problem is run.

The IST restart deck yielded by the above errors includes the contents of the intermediate storage tape at the point where the error termination occurred. The use of this deck is described in the user's description of GIVE. Such a deck permits the user to restart his analysis at the point of termination, after he has corrected the error.

Glossary of general terms

The main purpose of the glossary is to provide a convenient reference for important terms employed in the user's descriptions. The definitions are general and are intended to serve mainly as 'reminders.' A few computer terms not specific to the BCTRY system are defined but the reader must look to other sources for precise definitions.

Access deck - A set of monitor control cards which precede the BCTRY control cards on every run. Their function is to prepare the BC TRY system for operation.

Binary cards - Cards punched in the binary number system. They are not easily interpreted as are BCD cards.

BCD cards - Binary coded decimal cards. The code is a simple one whereby each letter or number is designated by a punch in a particular row of a column. When cards are punched in BCD - as they are on the key punch - the characters can be interpreted on a listing machine or an interpreter in a simple one-character per column fashion. In other words, when cards are output by the computer in BCD you can read them; when punched in binary you cannot.

Component control (CC) cards - Control cards read by a component program. Information punched on CC cards serves to label data and to specify parameters and method choices.

Component programs - All programs monitored by the general executive program, i.e., programs which operate upon data rather than upon other programs. A component program is called by an executive control card bearing the program name, e.g., GIVE, COR3, CC5.

Core - The computer 'memory,' an electronic circuit which represents the data given it. Programs and data must be 'in core' before they can function.

Data storage tape (DST) - Contains identification and raw scores only. All other data are on the IST.

Decimal card format - The FORTRAN format (9F8.4), i.e., up to nine 8-column numbers per card with the decimal point assumed between the fourth and fifth column of each field.

Executive control (E.C.) cards - Control cards which are read by the general executive program. They always consist of a slash in column one followed by a name, e.g., '/COR2', which signals an operation to be performed, e.g., to load the COR2 program into core.

Executive program - See 'General executive program.'

Field - A set of consecutive columns in which a single number is punched, e.g., columns 1-4 might be the field in which the number 25 is to be punched.

File - A set of information written on one section of tape. For example, the IST contains a file of means, a file of standard deviations, etc., each written as a single list on the tape, separated by an 'end of file' marker. 'File' simply denotes that information occurs as a set and implies nothing - in contrast to such terms as 'record' - about the size of the set.

Fixed field name cards - Six-character alphabetic and numeric names punched in the format (9(A6,2x)). That is, on each card there are no more than nine 8-column fields, each of which contains a 6-character name in the first six columns. Names are then separated by two blank columns. No embedded blanks are permitted.

Free field name cards - Cards containing a series of alphabetic and/or numeric names separated by commas and ending with a period. Only columns 1-72 may be used.

General executive program (GEP) - The BCTRY monitor program. GEP reads executive control (E.C.) cards, loads component programs into core, moves tapes, terminates the job, makes error checks, etc.

Integer card format - The FORTRAN format (18I4), i.e., up to 18 4-column fields. Integers are packed right within each field.

Intermediate storage tape (IST) - Contains all information except the raw data which may be of use to more than one component program. In other words, the IST contains most quantities computed by each component program - correlations, communalities, etc. - and most labels input - identification, variable names, etc. See data storage tape for raw data.

Key phrase - An italicized phrase which specifies a type of card input. All key phrases are defined precisely in pages 27-35 and generally in this glossary.

Monitor input tape - Contains all control and data cards. Card input is transferred to this tape instead of being input to the computer directly.

Monitor output tapes - Contain all printed and punched output available to the user.

Restart deck - A set of binary cards containing the contents of the IST, it allows the user to save and input information computed on a previous run.

Shared data tapes - Contain all information output by a component program for further analysis by other programs. See 'Data storage tape' and 'Intermediate storage tape.'

STANALOG programs - All programs which perform primarily statistical analyses, e.g., COR2, DVP, NC2.

Standard - A word often found beside a particular component program control option meaning (1) that the corresponding choice of method or parameter is optional for most purposes and (2) that the option may be obtained by leaving the relevant columns of the control card blank.

Suppressed ('sleeper') variable - A variable which has been excluded from the process of factoring but is included in all other phases of the analysis.

System tape - Contains the executive program and all component programs. When an executive control card is encountered its corresponding program is read from the system tape into the computer.

Title card format (12A6) - The format of the title card identifying the problem. Any alphabetic or numeric characters, using any

User's description of the comment card

Summary

You may wish to have inserted in the printout before or after certain components special information or data that you think is important to have printed at that point in the analysis. You can now insert such material as a permanent printed record by the use of comment cards.

Use of comment

Card input. For each comment, two cards are necessary: the ****
executive and the component cards that contain the material to be ****
inserted. ****
****
(1) Executive control card. Punch '/COMMENT' in cols. 1-8. This ****
executive card, followed by the component cards, can be inserted ****
anywhere just before any other executive card in the control deck, or ****
at the end of the control cards for any component. There can be as ****
many such insertions as you like. ****
****
(2) Component control cards. On as many cards as necessary, ****
punch in cols. 1-72 the material you wish inserted. ****
****
****

Printed output. The material is printed in the form in which it is keypunched.

User's description of general executive operation - GEP

Introduction

The program is designed to initiate and terminate an analysis and to call component programs into operation from the BCTRY system tape, all at the user's direction via executive control cards.

Method

Once GEP is initiated by an access deck, the program continues to read and perform the operations specified by executive control (E.C.) cards until termination occurs. Four basic kinds of operations are performed, depending on the contents of the E.C. card read by GEP.

(1) If the E.C. card contains the code name for initiation (START), the program rewinds all tapes and sets up the intermediate storage tape (IST).

(2) If the E.C. card contains the code name for termination (END), the program rewinds tapes and returns control to monitor.

(3) If the E.C. card contains the code name of a component program GEP reads the component from the system tape and turns control over to the component.

(4) If the E.C. card does not contain a code name that GEP can recognize (i.e., not in any of the above categories), the program prints an error message stating that 'this card is not an executive control card,' prints the offending card, punches an 'IST restart deck' (see user's description of the IST restart program, GIVE, below), and returns control to monitor.

Use

Card input. The program accepts only executive control cards punched with a slash, '/', e.g. '/COR2++' in column 1 and a code name in columns 2-7. Since the E.C. cards which call component programs into operation are described in the user's descriptions of the individual programs, we shall provide instructions for punching only the initiation and termination E.C. cards in this description.

```
                                                                  ****
     (1) Initiation executive control card.  Punch '/START+' in cols.  ****
          1-7.  This card must be the first card to follow the access  ****
          deck.                                                       ****
                                                                      ****
     (2) Termination executive control card.  Punch '/END++' in cols.  ****
          1-7.  This card must be the last card in the input deck      ****
```

Note: as a part of termination, it is good practice to put a GIVE card (q.v.) just before end.

Tape input. Component programs are read from the BCTRY system tape.

Printed output. Primarily, printed output consists of cols. 1-80 of all executive control cards, and time and date of output.

Tape output. The program writes a file on the intermediate storage tape during the initiation process. However, this output is of no concern to the user.

Restrictions. The program is subject to no restrictions of interest to the user.

USER'S DESCRIPTION OF HOW TO USE ''OUTSIDE'' COMPONENTS IN BC TRY RUNS -- TEST

INTRODUCTION

THE BC TRY SYSTEM INCLUDES ONLY COMPONENTS THAT EXECUTE THE INTEGRATED LOGIC OF REDUCING NV VARIABLES TO A SALIENT MEANINGFUL SET OF K GENERAL VARIABLES (V-ANALYSIS) AND OF REDUCING NS DISCRETE INDIVIDUALS TO P CONTRASTING TYPES OF INDIVIDUALS (O-ANALYSIS). HENCE, IT SHOULD NOT FORMALLY INCLUDE ANY COMPONENTS NOT DIRECTLY ADDRESSED TO THESE OBJECTIVES--IT SHOULD NOT BECOME A GENERAL CATCH-ALL MONITOR CONTROLLING DIVERSE STATISTICAL PROCEDURE THAT DO NOT OBVIOUSLY RELATE DIRECTLY AND LOGICALLY TO THIS ANALYTICAL DESIGN.

NEVERTHELESS, IT DOES NOT INCLUDE ALL POSSIBLE COMPONENTS THAT CAN FIT IN WITH THE AVOWED DESIGN. ANALYSTS MAY PROGRAM VERY FITTING COMPONENTS THAT THEY WOULD LIKE TO INCORPORATE IN A BC TRY RUN IN ORDER TO PERFORM A SPECIAL ADDITIONAL ANALYSIS NOT POSSIBLE WITH THE PRESENT SYSTEM. FURTHERMORE, THEY MAY WISH TO INCLUDE IN A BC TRY RUN ANOTHER COMPONENT THAT DOES NOT LOGICALLY FIT IN WITH THE OBJECTIVES OF THE SYSTEM BUT WHICH IS NEVERTHELESS ESSENTIAL TO THEIR OWN RESEARCH OBJECTIVES.

THE COMPONENT TEST PERMITS THE ANALYST TO RUN ANY ''OUTSIDE'' PROGRAM HE WISHES AT ANY PLACE IN HIS REGULAR BC TRY RUN, PROVIDED THE OUTSIDE COMPONENT HAS BEEN EXPRESSLY PROGRAMMED TO BE COMPATIBLE WITH BC TRY. (SEE THE DESCRIPTION OF TEST IN THE PROGRAMMER'S MANUAL).

TEST IS A REGULAR COMPONENT OF BC TRY THAT IS USED IN THE DEVELOPMENT OF THE SYSTEM TO TEST FULLY ANY NEW COMPONENT WHILE IT IS IN OBJECT DECK OR SOURCE DECK FORM AND BEFORE IT IS PUT ON THE SYSTEM TAPE. IT CAN THEREFORE BE EMPLOYED TO INCORPORATE ANY PROGRAM IN DECK FORM PROVIDED THE PROGRAM CAN BE COMPATIBLY CONTROLLED BY GEP. SUCH A PROGRAM CAN UTILIZE ANY OF THE FILES GENERATED BY ANY BC TRY COMPONENT ON THE DATA-SHARING TAPES, IST OR DST, Q.V., AND CAN DEPOSIT RESULTS IN THESE FILES, PROVIDED IT IS CORRECTLY PROGRAMMED TO DO SO.

METHOD

AT THE BEGINNING OF A BC TRY RUN WHILE UNDER MONITOR AND AFTER THE JOB IS INITIATED, THE OUTSIDE PROGRAM IS INPUT ON AN EXTRA SCRATCH TAPE FROM ITS CARDS VIA TAPE. LATER WHEN THE EXECUTIVE PROGRAM ENCOUNTERS A ''/TEST++'' CARD, THE EXECUTIVE PASSES CONTROL TO THE OUTSIDE PROGRAM AND TERMINATES ITS EXECUTION WITH AN ''/END'' CARD OR WHEN AN EXECUTIVE CARD CALLS ANOTHER COMPONENT.

NOTE.. THE USER SHOULD BE ASSURED THAT THE OUTSIDE PROGRAM HAS BEEN EXPRESSLY DESIGNED TO BE COMPATIBLE WITH BC TRY.

USE

FOR THE STARTER DECK WHEN TEST IS USED SEE THE CONSULTANT.

WHEN THE EXECUTIVE PROGRAM ENCOUNTERS A CARD WITH ''/TEST++'' PUNCHED IN COLS. 1-7, IT READS AN INTEGER N FROM COLS. 1-4 OF THE NEXT CARD AND CALLS CHAIN (N, B3). THE PROGRAM IN THE NTH RECORD OF B3 IS THEN THEN PUT INTO OPERATION.

THE FOLLOWING EXAMPLE SHOULD MAKE THE USE OF THIS FEATURE CLEAR

```
*   JOB (OR ID CARDS AS REQUIRED BY THE COMPUTING CENTER)                    ****
*  COL. 7 OR BEYOND ..  ''CHAIN (1,B2)''                                     ****
                                                                             ****
BC TRY STARTER DECK WITHOUT THE ''* DATA'' CARD.                             ****
                                                                             ****
*  COL. 7 OR BEYOND ..  ''CHAIN (1,B3)''                                     ****
                                                                             ****
OBJECT OR SOURCE DECKS WITH APPROPRIATE MONITOR CONTROL CARDS FOR            ****
''PROGRAM X''  TO BE RUN WITH THE SYSTEM                                     ****
                                                                             ****
*  COL. 7 OR BEYOND..  ''CHAIN (2,B3)''                                      ****
                                                                             ****.
AS ABOVE FOR ANOTHER ''PROGRAM Y'' TO BE RUN WITH THE SYSTEM                 ****
                                                                             ****
*  COL. 7  OR BEYOND..  ''DATA''                                             ****
''/START''  CARD TO INITIATE THE EXECUTIVE PROGRAM FOR BC TRY                ****
                                                                             ****
THE REGULAR BC TRY CONTROL CARDS FOLLOWS. WHEN THE OUTSIDE COMPONENT         ****
IS CALLED IT IS TREATED AS THOUGH IT WERE THE REGULAR COMPONENT              ****
WITH THE NAME TEST.  IT IS CALLED AS FOLLOWS..                               ****
                                                                             ****
''/TEST''                                                                    ****
''0001''                                                                     ****
CONTROL CARDS AND DATA CARDS FOR  ''PROGRAM X''                              ****
. . .                                                                        ****
ANY BC TRY COMPONENT                                                         ****
. . .                                                                        ****
''/TEST''                                                                    ****
''0002''                                                                     ****
CONTROL AND DATA CARDS FOR PROGRAM Y                                         ****
. . .                                                                        ****
ETC.                                                                         ****
. . .                                                                        ****
''/END'' OR ANY OTHER BC TRY COMPONENT EXECUTIVE CARD                        ****
```

```
*********************************************************
THE +'S INDICATED IN THE INSTRUCTIONS FOR CONTROL CARD
    PUNCHING STAND FOR BLANK COLUMNS ON THE CARD
*********************************************************
```

User's description of program to generate files on and retrieve them from the intermediate storage tape: GIST

Introduction

The contents of the files of IST are normally routinely input on IST as processed data by the component programs of BCTRY, or automatically output to the programs as needed by them. Program GIST, however, permits the analyst to input initially or to replace at will any file contents or to call directly for the output of any of them on cards.

His direct access to IST through GIST thus enables him to perform a wide variety of analyses not possible by rigid ties of IST to component programs. A few (of many) examples:

Revising a structure analysis. After a standard full-cycle CC (or NC) analysis a revised structure analysis (see CSA or SPAN) can be executed by replacing by GIST the revised set (in file CLUST1 or REFLX1).

Rotation of a reduced set of dimensions. If a principal axes solution has 'overfactored' into trivial dimensions, the analyst can select only the initial most salient dimensions UFACT1 on the salient dimensions only. These can be rotated by GYRO.

V-analysis of published data. Any type of BCTRY V-analysis can be executed on a published correlation matrix by inputting this matrix (in file CORRM1) by GIST. If only factor coefficients have been published the R-matrix can be reproduced by inputting them by GIST (in file UFACT1) and then calling the program FAST.

Analysis of results by 'outside' programs. The output deck of cluster scores described above can be analyzed by any available computer program that operates on scores. Another example: the card output by GIST of the correlation matrix (in file CORRM1) can be analyzed by, say, any regression program.

Note: decks output by GIST are in readable decimal or integer cards.

Method

GIST is designed to input or output any IST file or any set of IST files. Inputting a file consists of (1) reading a file name and an array of information (e.g., a set of means, a correlation matrix, etc.) from BCD cards (see Glossary), and (2) writing this array in the IST file corresponding to the name. Outputting a file consists of (1) reading a file name from a BCD card, and (2) punching the array of information in the IST file corresponding to the name on BCD cards. These operations are performed by means of sets of cards called 'file packages.'

Corresponding to each file are two types of control cards.
(1) The output file package. This package consists of a card with an IST file name in cols. 1-6, causing GIST to punch the contents of the file onto BCD cards. The output may be read by the user (i.e., as decimal or integer cards). They contain identification in cols. 73-80.
(2) The input file package. This package consists of: (A) a card containing an IST file name in cols. 1-6 and a '1' in col. 12 (indicating input), (B) a parameter card, giving the number of rows and/or columns of the array of information to be input, and (C) cards containing the array, causing GIST to write the array on the cards into the IST file corresponding to the file name.

Once GIST is initiated by an executive control card, the program continues to read file packages (input, output, or both) until it encounters a blank card following the last file package.

GIST maintains a list of the names of all legal GIST files (i.e., the files which the program is capable of generating). If the first six columns of the first card in a file pack contain a name not in this list an error message is printed. This failure can occur when the user (1) mispunches a name, (2) asks for an illegal file, (3) provides incorrect parameters, or (4) fails to insert a blank card after the last file package. The error may be traced by determining from printed or punched output the last file package correctly transmitted by GIST.

Use

Card input. Two types of cards are needed.

(1) Executive control card. Punch '/GIST++' in cols. 1-7. ****
****

(2) Component control cards. The description of component ****
control cards in the table below includes a section for ****
each IST file. Each section gives complete information ****
for both the input and the output file packages. ****

While the order in which sections have been placed in the Manual corresponds to the order of the file names used in previous sections, the different file packages for GIST may be input in any order whatsoever. Thus, to use this program description, the analyst should first punch the executive control card and then turn to the section(s) on the IST file(s) in which he is interested. If information on IST is to be output, he punches the card given under 'file package output'. If information is to be input, he punches cards specified under 'file package input.' He may then combine the sets of control cards in any order so long as they are preceded by the executive control card and followed by a blank card.

```
     Variable format input.  Variable format is applicable to input of    ****
data cards only (name cards, title cards, and parameter cards must        ****
be input in the standard format).  To use this feature, insert one        ****
and only one variable format card (see glossary) between what is          ****
described in all IST sections below as card (1) and card (2).  To         ****
signal the program that a variable format card has been used in any       ****
particular set, on the first card punch '1' in col. 16 in addition        ****
to the '1' always punched in col. 12.                                     ****
                                                                          ****
     When a matrix is input, the format statement should describe only    ****
a single row; when the matrix is not rectangular, e.g., the lower         ****
triangle correlation matrix, the format should describe only the          ****
longest row.                                                              ****
                                                                          ****
     In the description of control cards below, cards are described       ****
entirely through the use of 'key phrases.'  The reader should refer       ****
to 'general rules' if he is not yet familiar with these conventions.      ****
the symbols used are:  NS, number of subjects, NV number of variables,    ****
K, number of dimensions or composites.                                    ****


***********************************************************************
Remember a blank card must always follow the last file package.
***********************************************************************


IST files and their input and output file packages

IDFILE:  Data identification.                                             ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'IDFILE' in columns 1-6.                                  ****
     (2)  Blank card (if no more GIST cards follow).                      ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'IDFILE' in columns 1-6 and '1' in column 12.             ****
     (2)  Punch as a title card the identification of the data from       ****
          which the correlation matrix is derived.                        ****
     (3)  Blank card (if no more GIST cards follow).                      ****
                                                                          ****
                                                                          ****
VNAMS1:  Variable-names.                                                  ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'VNAMS1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'VNAMS1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV in cols. 1-4.                     ****
     (3)  Punch on fixed field named cards the names of the NV            ****
          variables.  The alphanumeric names must be packed left with     ****
          no embedded blanks.                                             ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
VSUMS1:  Sums of raw scores.                                              ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'VSUMS1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'VSUMS1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV in cols. 1-4.                     ****
     (3)  Punch as a vector in decimal card format the NV variable sum    ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
MEANS1:  Raw score means.                                                 ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'MEANS1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package                                                   ****
                                                                          ****
     (1)  Punch 'MEANS1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV and NS in cols. 1-4 and           ****
          5-8 respectively.                                               ****
     (3)  Punch as a vector in decimal card format the NV raw score       ****
          means.                                                          ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
STDEV1:  Standard deviations of variables.                                ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'STDEV1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'STDEV1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV in cols. 1-4.                     ****
     (3)  Punch as a vector in decimal card format the NV standard        ****
          deviations.                                                     ****
     (4)  Blank card (if no GIST cards follow).                           ****
```

```
CORRM1:  Correlations.                                                    ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'CORRM1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'CORRM1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV and NS in cols. 1-4 and           ****
          5-8 respectively.                                               ****
     (3)  Punch the lower-triangle correlation matrix in decimal card     ****
          format (9F8.4) as NV-1 vectors with no diagonal elements.       ****
          Thus the only element of vector 1, i.e., row 2, is the          ****
          correlation of variable 1 with variable 2.  If col. 20 of       ****
          card (1) is punched '1', the upper-triangle matrix may be       ****
          punched.  In this latter case, vector one would contain the     ****
          NV-1 correlations of V1 with the NV-1 other variables.          ****
          The last vector would then be the correlation between NV-1      ****
          and NV.                                                         ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
DIAGV1:  Diagonal values.                                                 ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'DIAGV1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'DIAGV1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV in col. 1-4.                      ****
     (3)  Punch as a vector in decimal card format the NV diagonal        ****
          values.                                                         ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
UFACT1:  Unrotated factor coefficients.                                   ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'UFACT1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'UFACT1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with NV and K in cols. 1-4 and            ****
          5-8 respectively.                                               ****
     (3)  Punch as a matrix in decimal card format the factor          ****
          coefficient matrix.  The K rows of this matrix correspond     ****
          to the factors, and the NV columns correspond to the          ****
          variables.                                                    ****
     (4)  Blank (if no GIST cards follow).                              ****
                                                                        ****
                                                                        ****
CLUST1:  Cluster indicators.                                            ****
                                                                        ****
     Output file package.                                               ****
                                                                        ****
     (1)  Punch 'CLUST1' in cols. 1-6.                                  ****
     (2)  Blank card (if no GIST cards follow).                         ****
                                                                        ****
     Input file package.                                                ****
                                                                        ****
     (1)  Punch 'CLUST1' in cols. 1-6 and '1' in col. 12.               ****
     (2)  This card and all cards following are identical with those    ****
          of REFLX1 below excepting the signs of the indices            ****
          are omitted.                                                  ****
                                                                        ****
                                                                        ****
REFLX1:  Reflection indicators.                                         ****
                                                                        ****
     Output file package.                                               ****
                                                                        ****
     (1)  Punch 'REFLX1' in cols. 1-6.                                  ****
     (2)  Blank card (if no GIST cards follow).                         ****
                                                                        ****
     Input file package.                                                ****
                                                                        ****
     (1)  Punch 'REFLX1' in cols. 1-6 and '1' in col. 12.               ****
     (2)  Punch an integer card with K in cols. 1-4.                    ****
     (3)  Punch as a vector in integer card format S(1),...S(K), where  ****
          S(K) is the number of variables defining the I'th cluster.    ****
     (4)  Punch as a vector in integer card format the indices of       ****
          the S(I) variables defining the I'th cluster.  In this        ****
          manner punch K vectors, input in the order S(1),...S(K).      ****
          Variable format is not permitted.                             ****
     (5)  Blank card (if no GIST cards follow).                         ****
                                                                        ****
                                                                        ****
RFACT1:  Rotated factor coefficients.                                   ****
                                                                        ****
     Output file package.                                               ****
                                                                        ****
     (1)  Punch 'RFACT1' in cols. 1-6.                                  ****
     (2)  Blank card (if no GIST cards follow).                         ****
     Input file package.                                               ****
                                                                       ****
     (1)  Punch 'RFACT1' in cols. 1-6 and '1' in col. 12.              ****
     (2)  Punch an integer card with NV and K in cols. 1-4 and 5-8     ****
          respectively.                                                ****
     (3)  Punch as a matrix in decimal card format the rotated factor  ****
          coefficients.  Punch the matrix such that the K rows         ****
          correspond to the ordered factors and the NV columns         ****
          correspond to the ordered variables.                         ****
     (4)  Blank card (if no GIST cards follow).                        ****
                                                                       ****
                                                                       ****
BASIS1:  Correlations among oblique dimensions.                        ****
                                                                       ****
     Output file package.                                              ****
                                                                       ****
     (1)  Punch 'BASIS1' in cols. 1-6.                                 ****
     (2)  Blank card (if no GIST cards follow).                        ****
                                                                       ****
                                                                       ****
     Input file package.                                               ****
                                                                       ****
     (1)  Punch 'BASIS1' in cols. 1-6 and '1' in col. 12.              ****
     (2)  Punch K in cols. 1-4 of an integer card.                     ****
     (3)  Punch as a matrix in decimal card format the correlations    ****
          between the K factors.  This will be a K by K symmetrical    ****
          matrix of ordered factors.  The first element of the first   ****
          row must be the correlation between factor 1 and itself.     ****
     (4)  Blank card (if no GIST cards follow).                        ****
                                                                       ****
                                                                       ****
MEANS2:  Means of the composite scores.                                ****
                                                                       ****
     Output file package.                                              ****
                                                                       ****
     (1)  Punch 'MEANS2' in cols. 1-6.                                 ****
     (2)  Blank card (if no GIST cards follow).                        ****
                                                                       ****
     Input file package.                                               ****
                                                                       ****
     (1)  Punch 'MEANS2' in cols. 1-6 and '1' in col. 12.              ****
     (2)  Punch one decimal card with the mean of the factor scores    ****
          in the first field.                                          ****
     (3)  Blank card (if no GIST cards follow).                        ****
                                                                       ****
                                                                       ****
STDEV2:  Composite score standard deviations.                          ****
                                                                       ****
     Output file package.                                              ****
                                                                       ****
     (1)  Punch 'STDEV2' in cols. 1-6.                                 ****
     (2)  Blank card (if no GIST cards follow).                        ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'STDEV2' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch K in cols. 1-4 of an integer card.                        ****
     (3)  Punch as a vector in decimal card format the K ordered          ****
          standard deviations.                                            ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
WEIGH1:  Weights assigned to component variables in calculating           ****
composite scores.                                                         ****
                                                                          ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'WEIGH1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'WEIGH1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch NV and K on an integer card in cols. 1-4 and 5-8          ****
          respectively.                                                   ****
     (3)  Punch as a matrix in decimal card format the weight of each     ****
          variable in calculating each composite score.  The K rows of    ****
          this matrix will correspond to the ordered composites, and      ****
          the NV columns will correspond to the ordered variables.        ****
     (4)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
                                                                          ****
FSCOR1:  Cluster or factor scores.                                        ****
                                                                          ****
                                                                          ****
     Output file package.                                                 ****
                                                                          ****
     (1)  Punch 'FSCOR1' in cols. 1-6.                                    ****
     (2)  Blank card (if no GIST cards follow).                           ****
                                                                          ****
                                                                          ****
     Input file package.                                                  ****
                                                                          ****
     (1)  Punch 'FSCOR1' in cols. 1-6 and '1' in col. 12.                 ****
     (2)  Punch an integer card with K and NS in cols. 1-4 and 5-8        ****
          respectively.                                                   ****
     (3)  Punch as a matrix in decimal card format the factor scores      ****
          of each subject on each factor.  Thus the NS rows of the        ****
          matrix correspond to subjects and the K columns correspond      ****
          to the ordered composites.                                      ****
     (4)  Blank card (if no GIST cards follow).                           ****
```

GIST also punches, 2SENS1, 2MEANS1, 2STDEV1, COVAR1, REORD1, ****
CORRM2, VAREN1. ****
****

Tape input. Any file which is to be output by GIST will have to be on the intermediate storage tape (IST). There is no tape input special to GIST.

Printed output. When a file package is input, all parameters and data are printed in whatever format is standard for input. Only the parameters of the output file package are printed.

Card output. The card output consists of complete input file packages which can later be used without alteration as file package input through GIST. Thus the reader need only turn to the 'file package input' section of the appropriate file to find the form of the card output. Variable format statements can not be used to specify output format.

In columns 73-80 of the output cards certain identifying data are punched for the user's convenience only: in columns 73-74 an integer is punched which stands for the file's serial position on IST. In columns 75-78 an integer is punched which stands for the row of the matrix output. Columns 79-80 contain the card sequence number, a series of integers beginning with 00 for the first card of each row.

Tape output. GIST outputs no information to IST other than that which is input via file packages nor does GIST 'remove' form IST that information which it prints and punches.

USER'S DESCRIPTION OF PROGRAM TO RESTART THE INTERMEDIATE STORAGE TAPE.. GIVE

INTRODUCTION

AT SOME POINT IN AN ANALYSIS AND AT THE END OF IT, THE ANALYST MAY WISH TO PRESERVE THE RESULTS STORED IN THE FILES OF IST. AN EFFICIENT WAY TO DO SO IS PROVIDED BY THE PROGRAM GIVE THAT 'DUMPS' ALL THE FILES OF IST ON BINARY CARDS. THIS DECK CONSTITUTES A 'RESTART DECK' WHICH ENABLES ONE LATER TO REESTABLISH THE FILES ON IST FOR PURPOSES OF FURTHER ANALYSIS. REESTABLISHING THE FILES IS ACHIEVED BY THE PROGRAM TAKE WHICH UTILIZES THE GIVE RESTART DECK. THE GIVE AND TAKE PROGRAMS THUS AVOID THE CUMBROUS PROCEDURE OF SAVING TAPES CONTAINING RESULTS OF THE ANALYSIS. USER'S DESCRIPTIONS OF THE TWO PROGRAMS ARE GIVEN SEPARATELY BELOW.

THE IMPORTANCE TO THE USER OF UNDERSTANDING AND GENEROUSLY EMPLOYING THE GIVE-TAKE PROCEDURES MAY BE SEEN IN THE FOLLOWING EXAMPLES..

PRESERVING THE ENTIRE IST FILES OF A GIVEN ANALYSIS. ROUTINELY THE ANALYST SHOULD INSERT A GIVE EXECUTIVE CARD JUST BEFORE THE END CARD OF ANY ANALYSIS. HE CAN THEREFORE REWORK ANY SEGMENT HE WISHES WITHOUT DOING THE WHOLE EXPENSIVE ANALYSIS OVER AGAIN. REESTABLISHING THE IST FILE BY TAKE PERMITS THE ANALYST, FOR EXAMPLE, TO REDEFINE HIS CLUSTERS BY REPLACING THE CLUST1 FILE THROUGH GIST BY A REVISED FILE, AND TO REWORK ONLY THE STRUCTURE ANALYSIS (CSA). AGAIN, IN LEAST SQUARES FACTORING (FALS WITH GYRO) THE DIMENSIONS OFTEN NEED REFLECTION IN ORDER TO SECURE SPAN DIAGRAMS THAT ARE EASILY INTERPRETABLE. AFTER TAKE, ONE NEEDS ONLY TO INPUT BY GIST A REPLACING UFACT1 FILE CONTAINING REFLECTED FACTOR COEFFICIENTS, AND THEN TO RECOMPUTE SPAN.

SAVING INITIAL ESTIMATES OF COMMUNALITIES. WHEN FACTORING BY A METHOD THAT REQUIRES COMMUNALITIES (CC5 AND FALS), THE ANALYST MAY COMPUTE INITIAL ESTIMATES BY DVP 60, EXPENSIVE OF COMPUTER TIME IF NV IS LARGE. THESE COMMUNALITIES, SET IN THE DIAGV1 FILE, ARE LOST AFTER THE IST ITERATION WHEN NEW COMMUNALITIES FROM THE FACTOR COEFFICIENTS REPLACE THEM. THEREFORE, IF HE WISHES TO PRESERVE THE INITIAL ESTIMATES IN CASE HE MAY LATER WISH TO REFACTOR THE MATRIX, HE WOULD INSERT A GIVE CARD AFTER THE DVP PROGRAM, YIELDING A RESTART DECK THAT CAN AT ANY LATER TIME REESTABLISH FOR REFACTORING BOTH THE CORRELATION MATRIX AND THE INITIAL COMMUNALITY ESTIMATES.

NOTE.. ALL GIVE DECKS ARE IN BINARY FORM AND ARE NOT, EXCEPT FOR THE IDENTIFICATION FIELD, READABLE.

METHOD

THE IST RESTART DECK MAY BE PRODUCED EITHER BY AN EXECUTIVE CONTROL CARD CONTAINING ''/GIVE'' OR BY AN ERROR TERMINATION (SEE 'GENERAL RULES' ABOVE). TO PRODUCE THE IST RESTART DECK, THE PROGRAM SIMPLY REWINDS THE IST, AND THEN PUNCHES THE CONTENTS OF ALL FILES CURRENTLY PRESENT ON IST. THE PROGRAM PUNCHES THE FILES SUCCESSIVELY ONTO CARDS, SO THAT THE FIRST FILE PUNCHED IS USUALLY IDFILE, AND THE LAST FILE PUNCHED IS ALWAYS FINISH, THE 'END OF LOGICAL TAPE' FILE. EACH CARD CONTAINS THE CONTENTS OF THE FILE PUNCHED IN THE BINARY MODE (I.E., NOT READABLE BY THE USER), AND THE NAME OF THE FILE AND A CARD SEQUENCE INDICATOR PUNCHED IN THE BCD MODE (I.E., READABLE BY THE ANALYST).

THE FINISH FILE IS AUTOMATICALLY WRITTEN AS THE LAST FILE ON IST, HENCE THE USER NEED NOT CONCERN HIMSELF WITH IT. THE FILE IS MENTIONED HERE ONLY BECAUSE IT CAN BE USED TO SEPARATE TWO OR MORE IST RESTART DECKS.

THE IST MAY BE RESTORED IN ONLY ONE WAY, I.E., BY AN EXECUTIVE CONTROL CARD CONTAINING ''/TAKE'' FOLLOWED BY A RESTART DECK FROM GIVE. THE PROCEDURE IS AS FOLLOWS.. THE IST IS REWOUND AND ERROR CHECKED. IF THE TAPE HAS NOT BEEN INITIATED BY A ''/START'' CARD DURING THE PRESENT RUN, AN ERROR TERMINATION OCCURS (SEE 'GENERAL RULES'). OTHERWISE, THE FILES PRESENT IN THE RESTART DECK FROM GIVE ARE ADDED SUCCESSIVELY TO THOSE PRESENT ON THE IST. IF THE IST ALREADY CONTAINS FILES PRESENT IN THE RESTART DECK THE CURRENT CONTENTS OF THESE FILES ON IST ARE REPLACED BY THE CONTENTS OF THE FILES IN THE RESTART DECK.

USE OF GIVE

THE RESTART DECK IS PRODUCED BY ERROR TERMINATIONS OR THE USER CAN OBTAIN A RESTART DECK WHENEVER HE WISHES BY USING AN EXECUTIVE CONTROL CARD. THE RESULTING RESTART DECK IS THE SAME REGARDLESS OF WHETHER IT WAS OBTAINED BY AN ERROR TERMINATION OR BY A CONTROL CARD.

CARD INPUT. EXECUTIVE CONTROL CARD.. PUNCH ''/GIVE++'' IN ****
COLS. 1-7. NO OTHER CONTROL OR DATA CARDS ARE READ. ****

TAPE INPUT CONSISTS OF ALL FILES WHICH CURRENTLY EXIST ON THE INTERMEDIATE STORAGE TAPE, INCLUDING FINISH, THE FILE WHICH IS AUTOMATICALLY LAST ON THE TAPE AND WHICH INDICATES THAT NO MORE FILES ARE ON IST.

PRINTED OUTPUT CONSISTS OF A LIST OF THE NAMES OF THE IST FILES WHICH ARE ON IST WHEN GIVE BEGINS ITS OPERATIONS AND WHICH ARE INCLUDED IN THE RESTART DECK OUTPUT BY THE PROGRAM.

THE PROGRAM PRODUCES NO TAPE OUTPUT.

PUNCHED OUTPUT CONSISTS OF THE IST RESTART DECK CONTAINING THE SUCCESSIVE FILES READ BY GIVE FROM THE INTERMEDIATE STORAGE TAPE BEGINNING WITH AN IDENTIFICATION CARD. THE FORMAT OF THESE CARDS IS AS FOLLOWS..

COLUMNS	CONTENTS
1-72	THE DATA CONTAINED IN THE IST FILE PUNCHED IN THE BINARY MODE (I.E., NOT EASILY READ BY THE USER).
73-78	THE NAME OF THE IST FILE (WHOSE CONTENTS APPEAR IN COLS. 1-72), PUNCHED IN THE BCD MODE (I.E., EASILY READ BY USER WHEN INTERPRETED). THESE NAMES ARE THE ONES DESCRIBED EARLIER, I.E., IDFILE, CORRM1, ETC.
79-80	THE CARD SEQUENCE INDICATOR FOR EACH FILE, PUNCHED IN THE BCD MODE (I.E., READABLE WHEN INTERPRETED), AN ALPHABETIC INDICATOR IS USED, I.E., FOR A GIVEN FILE THE FIRST CARD CONTAINS 'AA' IN COLS. 79-80, THE SECOND CARD CONTAINS 'AB', AND SO ON.

USE OF TAKE

THE RESTART DECK FROM GIVE OBTAINED EITHER BY AN ERROR TERMINATION OR BY A CONTROL CARD MAY BE USED TO RESTORE THE IST BY TAKE. THE PROCEDURE IS THE SAME FOR EITHER DECK.

CARD INPUT. AN EXECUTIVE CONTROL CARD AND THE DECK OF RESTART ****
CARDS OBTAINED FROM GIVE OR MADE UP TO CORRESPOND TO A GIVE DECK ARE ****
REQUIRED. ****
****

(1) EXECUTIVE CONTROL CARD. PUNCH ''/TAKE++'' IN COLS. 1-7. ****
NO OTHER CONTROL CARDS ARE READ BY THE PROGRAM. IT SHOULD BE NOTED ****
THAT THE ABOVE EXECUTIVE CONTROL CARD MUST OCCUR AFTER THE ''/START'' ****
EXECUTIVE CONTROL CARD AND BEFORE THE ''/END'' EXECUTIVE CONTROL CARD ****
FOR ANY GIVEN RUN. AS A GENERAL RULE THE ''/TAKE'' CARD SHOULD BE ****
PLACED IMMEDIATELY AFTER THE ''/START'' CARD.

****
(2) DATA CARDS. IST RESTART DECK FROM GIVE. THESE CARDS ARE ****
PLACED IMMEDIATELY BEHIND THE ''TAKE'' EXECUTIVE CONTROL CARD. THE IST ****
RESTART DECK IS INPUT EXACTLY AS IT IS OUTPUT BY THE GIVE COMPONENT ****
HOWEVER, BEFORE INPUTTING THE DECK THE USER IS WELL ADVISED TO INTER- ****
PRET COLS. 73-80 OF THE CARDS AND CHECK FOR THE FOLLOWING TYPES OF ****
COMMON ERRORS.. (A) THE DECK CONTAINS EXTRANEOUS CARDS, E.G., CARD ****
INPUT FROM GIST, BLANK CARDS, ETC. IF THIS ERROR OCCURS THE USER ****
SHOULD REMOVE ALL CARDS NOT PROPERLY LABELED IN COLS. 73-80. (B) THE ****
DECK CONTAINS TWO OR MORE IST RESTART DECKS FROM GIVE. SEPARATE IST ****
RESTART DECKS MAY BE IDENTIFIED BY THE FACT THAT THE FIRST FILE IN ****
ALMOST ALL RESTART DECKS IS THE IDFILE AND THE LAST FILE IN ALL RESTART ****
DECKS IS THE FINISH FILE. WHEN THIS TYPE OF ERROR OCCURS THE USER ****

SHOULD DETERMINE WHICH RESTART DECK HE WISHES TO USE FROM HIS PRINTED ****
OUTPUT AND FROM THE ORDINAL POSITIONS OF THE RESTART DECKS. (C) THE ****
DECK IS OUT OF ORDER. THE DECK MAY BE RESTORED TO ITS CORRECT ORDER BY ****
USING THE FILE NAME AND THE CARD SEQUENCE INDICATOR, WHICH ARE PUNCHED ****
IN COLS. 73-80 AND COLS. 79-80 RESPECTIVELY (SEE USER'S DESCRIPTION OF ****
THE GIVE UNIT ABOVE). ****

THE PROGRAM DOES NOT READ ANY INFORMATION FROM TAPE.

PRINTED OUTPUT. THE NAMES OF THE IST FILES READ FROM THE RESTART CARDS AND PLACED ON IST ARE PRINTED.

TAPE OUTPUT. THE FILES READ FROM THE RESTART DECK FROM GIVE ARE WRITTEN ON IST AS DESCRIBED IN THE METHOD SECTION. IT SHOULD BE NOTED THAT THE PROGRAM PRODUCES NO OTHER TAPE OUTPUT. THUS, THE PROGRAM RESTORES ONLY IST. IN ORDER TO RESTORE THE DATA STORAGE (DST), THE ANALYST MUST USE THE DATA PROCESSOR PROGRAM (DAP).

```
********************************************************
THE +'S INDICATED IN THE INSTRUCTIONS FOR CONTROL CARD
    PUNCHING STAND FOR BLANK COLUMNS ON THE CARD
********************************************************
```

User's description of data processor program: DAP (including DPRINT)

Introduction

This component prepares the data storage tape (DST) from a deck of cards containing the raw score matrix of NS scores of objects on NV variables. The tape is then used by other component programs which operate on the score matrix, such as a correlation program (e.g., COR2) or a cluster or factor score program (FACS). DAP permits one to identify each variable by name, to reorder the variables and to reflect any variable. It incorporates an extensive system of error checking. It accommodates missing data, and computes means and standard deviations.

Method

The initial processing of the data cards includes (A) reading them in, (B) checking the parameters NV and NS, the format statements, variable names and any transformations, and (C) writing on the data storage tape (DST) the title, NV, NS, and MISS (missing data). The data cards are then read and error checked in blocks, the variables are transformed if options to reorder or reflect them have been taken, the sums of scores and of their squares are calculated and output on the data storage tape. Finally, means and standard deviations are computed and output on the intermediate storage tape (IST).

Caution with dichotomous data: when the variables are dichotomous, like test items, the analyst should be sure that each shows variation. No correlational analysis should include a variable with a standard deviation of zero, i.e., 100 percent 'pass' (or 'fail').

Crucial check on input by DPRINT: in order to be sure that a score matrix is correctly input on DST, the user should always call for a printout of the score matrix input to DST. Called immediately after DAP, the component DPRINT reads and prints the score matrix exactly as it is stored on DST and utilized by other programs (COR2, COR3, FACS) (see the end of the Use section).

Use (if you are using the July 1965 version of BCTRY, see DAP2)

Card input. In addition to the executive control card, six ****
component control and data cards (and sets of cards) are involved in ****
the use of DAP. Some of the component control cards are not required. ****
****
(1) Executive control card. Punch '/DAP+++' in cols. 1-7. ****
****
(2) Component control cards. These cards contain an identifying ****
title, certain parameters of the data, and control ****
information. ****
****
Card A. Identification of data. This card contains the title to ****
be associated with the printed output of programs using the raw data, ****
and is punched in a title card format. ****

```
     Card B.  Parameters and control constants.  This card contains      ****
the number of variables, the number of observations and control          ****
information.  It is an integer card, as follows:                         ****
                                                                         ****
                                                                         ****
             COLUMNS          CONTENTS OF CARD B                         ****
                                                                         ****
               1-4     The number of variables, NV.                      ****
                                                                         ****
               5-8     The number of observations, NS.                   ****
                                                                         ****
                       (Ordinarily, DAP counts the number of             ****
                       observations appearing in the data                ****
                       cards and uses NS for a check.  However,          ****
                       if NS is not punched (i.e., cols. 5-8 are         ****
                       blank), the checking is skipped.)                 ****
                                                                         ****
               9-12    Controls the format used to read the data         ****
                       cards.                                            ****
                       '0'.  The format is input by the user (see Card   ****
                       C below).  Card sequence checks are not made.     ****
                                                                         ****
                       L, a positive number, indicates the field         ****
                       width (number of columns used) for each           ****
                       score.  Card sequence checks are made.            ****
                                                                         ****
               13-16   Specifies the existence of missing obser-         ****
                       vations by setting an indicator miss.             ****
                                                                         ****
                       'O'.  There are missing data.  A blank field      ****
                       (a score of -0) on a data card is interpre-       ****
                       ted as a missing observation.                     ****
                                                                         ****
                       '1'.  There are no missing data.  All obser-      ****
                       vations are complete.  In this case a blank       ****
                       field (a score of -0) on a data card will         ****
                       cause processing to be terminated with an         ****
                       error message.                                    ****
                                                                         ****
                       '2'.  There are no missing data.  All obser-      ****
                       vations are complete.  A blank field ( a          ****
                       score of -0) on a data card is interpreted        ****
                       as a legitimate score of zero (+0).               ****
                                                                         ****
     Cards C.  Format for reading data cards.  These cards must be       ****
included in the input deck if and only if cols. 9-12 of Card B contain   ****
'0'.  'INFORMAT' must be punched in cols. 1-8.  In cols. 9-80 punch      ****
a legal FORTRAN format statement (excluding statement number and         ****
'format', but including the outside parentheses) giving the format of    ****
all cards belonging to a subject.  Use only F and X conversion           ****
characters.  The format statement may be continued onto more cards,      ****
using cols. 9-80 with 'INFORMAT' always in cols. 1-8.  A maximum of      ****
eight cards may be used.  (Do not repeat the left parenthesis when       ****
continuing to a second card.)                                            ****
```

```
     Optional control cards D, E, F.  The following cards may be used     ****
by the analyst to specify the names of the variables, reordering           ****
of the variables, and/or reflections of some variables (i.e., reversing    ****
the signs of all observations of some variables).  None of these          ****
three types of control cards are required for any control option and       ****
may be left out entirely.                                                  ****
                                                                           ****
                                                                           ****
     Cards D.  Names of variables.  These cards specify an alphanumeric    ****
name of up to 6 characters for each variable.  This name is associated     ****
with the variable throughout the succeeding analysis and makes possible    ****
a much more readable printed output - especially if some attention is      ****
paid to forming names with high mnemonic value.  First, punch a card       ****
with 'VARIABLE+NAMES' in cols. 1-14.  Follow this card with a set of       ****
cards containing a list of NV names, punched as a vector on free field     ****
name cards.  A name may consist of from 1 to 6 characters, chosen from     ****
the 36 alphabetic and numeric characters.  The 6 characters must be        ****
packed left, with no blanks.  Within a field characters after the firs     ****
6 are ignored.  The names are formed into a list by separating them        ****
with commas and ending them with a period.  Punch this list in cols.       ****
1-72 on as many cards as required.  Do not split a name between two        ****
cards.  The sequence of these names must correspond to the order of        ****
appearance of the variables on the data cards.  The variable names         ****
cards may be omitted altogether.  Omit the 'VARIABLE+NAMES' cards also     ****
In this case the following names are generated automatically:  V001++,     ****
V002++, V003++, etc.  However, the reordering and reflection options       ****
are then unavailable.                                                      ****
                                                                           ****
                                                                           ****
     Cards E.  Reordering of variables.  These optional cards cause        ****
a reordering, or permutation, of the variables immediately after they      ****
are read from the data cards.  In all succeeding operations they           ****
appear in the new order and are associated with the correct name.         ****
Punch a card with 'REORDER' in cols. 1-7.  Follow this card with the       ****
same list of variable names as above, and punched by the same rules,       ****
but in the new order.  If this option is not desired, omit all Cards       ****
E described in the preceding paragraph.                                    ****
                                                                           ****
                                                                           ****
     Cards F.  Reflection of variables.  These optional cards cause        ****
the sign of a given variable to be changed immediately after it is         ****
read from the data cards.  In all succeeding operations the variable       ****
appears in this form.  Punch a card with 'REFLECT' in cols. 1-7.           ****
Follow this card with a list of variable names, punched by the above       ****
rules, which specifies the variables to be reflected.  If this option      ****
is not desired, omit all Cards F.                                          ****
                                                                           ****
                                                                           ****
                                                                           ****
     Data cards.  The data cards must be preceded by a card punched        ****
'DATA' in cols. 1-4 and followed by a card punched 'END+++' in cols.       ****
73-78 ('+' means 'blank column').  Data are punched by subject, each       ****
subject beginning on a new card.  If L is not 0 (i.e., cols. 9-12          ****
```

of Card B are not '0'), cols. 73-78 of each data card must be punched ****
with a six-character subject name or serial number and cols. 79 and 80 ****
must contain a sequence number giving the order of the cards with the ****
subject. If cols. 73-78 are not the same on all of a subject's cards, ****
or if the sequence number does not increase from one card to the next, ****
an error message is printed and the program will not continue. ****
****
The data must begin in col. 1 and continue in sequential fields ****
of width L until col. 72 is reached (or col. 70 if L = 5 or 7). If ****
necessary, continue on another card. ****
****
If L = 0, there is no error-checking of subject names or card ****
sequence numbers and, indeed, they need not be present. The 'INFORMAT ****
cards must specify a format which covers all of the cards present in ****
the input deck for a subject. ****

The program does not read any information from IST.

Printed output is self-explanatory. It includes a listing of all non-data cards in the input deck, the subject names, and the sum, mean, and SD of each variable. Printed output volume will vary from 2 to 20 pages.

Tape output. Tape output to IST and DST is as follows.

To intermediate storage tape (IST). The following information is written on IST for all control options (unless otherwise indicated).

INFORMATION	IST FILE
Identification of the data	IDFILE
Names of the NV variables	VNAMS1
Sums of the NV variables	VSUMS1
Means of the NV variables	MEANS1
Standard deviations of the NV variables	STDEV1
The sample sizes of each variable (for missing data only)	VAREN1

To data storage tape. The identification of the data, the number of variables NV, the missing data indicator, MISS, and the raw scores are all written on DST in the format shown below. In the following, let NS denote the number of subjects:

File 1. Heading file.
Record 1. Identification of data, 12 words written in the BCD mode.
Record 2. The same 12 words of identification written in the binary mode.
Record 3: Two words: NV and MISS.

File 2. The raw scores
The raw scores are written as NS records of NV words each, in the binary mode. Thus record I contains the

scores of subject I on the NV variables, where
I = 1, 2, ... NS.

The following restrictions on parameters and control constants should be noted:

The number of variables, NV: min 2, max 90.
The number of subjects, NS: min 2, max 9,999
The field width on data cards, l: min 0, max 9

Use of DPRINT (see Method section above).

Card input. The only card input required is the executive control card. Punch '/DPRINT' in cols. 1-7.

Output. The only output is the printed score matrix giving the raw scores on DST of the NV variables listed by the NS subject indices (ONAMS). There is no other output.

Note. The user should call DPRINT every time he inputs the data deck of cards via DAP, and check it for correctness. When DAP is called for FACS, however, he can avoid the DPRINT check provided he has in his files the raw correlations between cluster scores from CSA or NCSA. In this case if these correlations check with those given in FACS, then his card input to DAP for FACS is correct.

User's description of DAP2

Summary

The new DAP2 outputs the names of subjects onto printouts, punched cards, and to the new IST file, ONAMS1. It permits the user to specify the column positions of subjects' names on the data cards, or it generates unique names. It checks the number of data cards per subject, and counts the number of subjects NS.

In DAP2 each type of control is now identified by a 'name' intended to communicate the meaning of the control. DAP2 is virtually identical with DAP for the 'normal mode', i.e., when the number of variables is 90 or less, or for an O-analysis on 90 subjects or less. Under this normal mode there are, as in DAP, four main control options on control card B: NV, NS, LFIELD, and MISS, the rest of the card being blank. The last seven control options, from MTDAP to MAXVAL, are mainly relevant to the 'BIGNV mode', which will be operational when programming of PICK is complete in Fall, 1965. Even then, most users will use blank columns for these control options.

Use of DAP2

Follow the use section of DAP except as noted below.

Card input.

(1) DAP executive card. Punch '/DAP2++' in cols. 1-7. ****
****

(2) Component control cards. ****
****

Card A. IDFILE, identification of data on a card in title card format. ****
****
****
****

Card B. Parameters and controls. ****
****

COLUMNS CONTENTS OF CARD B ****
****

1-4 NV, number of variables. ****
****

5-8 NS, number of subjects (observations). Two new options are in DAP2: ****
****
****

'0' or blank. Just before the END card of the data set, a dummy subject must be inserted whose subject name is all dollar signs. DAP2 counts up to the dummy (standard). ****
****
****
****
****

NS. Punch the number of subjects. DAP2 checks this NS. ****
****

```
             9-12      LFIELD, length of field in format control.  No     ****
                       new options are added, but when field, L, is       ****
                       between 1 and 9, then see LABOPT (cols. 25-28)     ****
                       and LOCID (cols. 29-32).  DAP2 supplies missing    ****
                       ONAMS.  There is, however, a special option for    ****
                       the rare user with binary data on tape.  It is,    ****
                       '-1' binary, i.e., data are on binary tape         ****
                       (see notes on binary at end of Use).               ****
                                                                          ****
             13-16     MISS.  Last sentence in DAP now reads, blanks      ****
                       are given actual scores of +0.                     ****
                                                                          ****
Note.  The rest of Card B is new but can be blank for the normal mode,    ****
i.e., NV=90 or less, and if NS exceeds 90, no O-analysis is planned.      ****
                                                                          ****
             17-20     MTDAP, a special magnetic tape for DAP2            ****
                       required by the BIGNV mode.                        ****
                                                                          ****
                       '0' or blank.  Normal mode, with NV not greater    ****
                       than 200 and NS below 10,000.                      ****
                                                                          ****
                       '1'.  BIGNV mode, with NV not greater than 2000    ****
                       NS below 10,000, and where the output is to the    ****
                       new supply tape, MTSUP, and to the new IST file    ****
                       IPFILE.                                            ****
                                                                          ****
             21-24     MTIN, an optional auxiliary magnetic tape for      ****
                       input when NV exceeds 200, and NS exceeds 500.     ****
                       (see notes on MTIN at the end of the Use           ****
                       section).                                          ****
                                                                          ****
                       '0' or blank.  MTIN tape is not used (standard)    ****
                                                                          ****
                       '1'.  MTIN tape is used.                           ****
                                                                          ****
             25-28     LABOPT, optional checking of the ONAMS labels      ****
                       of the subjects.  Used only when LFIELD, cols.     ****
                       9-12 is a length, L, between 1 and 9.              ****
                                                                          ****
                       '0' or blank.  Checks ID field and card            ****
                       sequence (standard).                               ****
                                                                          ****
                       '1'.  Does not check ID field and card sequence    ****
                                                                          ****
                       '2'.  Checks card sequence only.                   ****
                                                                          ****
                       '3'.  Checks ID field only.                        ****
                                                                          ****
             29-32     LOCID, location of ID field.  Used when cols.      ****
                       9-12 is a length, L, between 1 and 9.              ****
                                                                          ****
                       '0' or blank.  ID field is in cols. 73-78 of       ****
                       data card and card sequence is in cols.            ****
                       79-80 (standard).                                  ****
```

```
                    '1'.  ID field is in cols. 1-6, sequence number    ****
                    in cols. 7-8.                                      ****
                                                                       ****
                    '-1'.  No ID fields in on data card.  DAP2         ****
                    generates ONAMS.                                   ****
                                                                       ****
          33-36     INFORMAT card count check.                         ****
                                                                       ****
                    '0' or blank.  No check is made (standard).        ****
                                                                       ****
                    NXC,  the number of cards per subject.  DAP2       ****
                    checks to see that precisely NXC cards will be     ****
                    read by INFORMAT for each subject.  NXC is         ****
                    definitely required if you have 100 or more        ****
                    cards per subject in the data deck.                ****
                                                                       ****
          37-40     KPACK, optional data packing in BIGNV mode ('1'    ****
                    in cols. 17-20).  Can be used only when data       ****
                    are integers.  Used to reduce tape reading and     ****
                    writing time, with no effect on computations.      ****
                                                                       ****
                    '0' or blank.  No packing (standard).              ****
                                                                       ****
                    '1'.  Data are positive integers, and will be      ****
                    packed.  The largest score must be not greater     ****
                    than MAXVAL (cols. 41-48).                         ****
                                                                       ****
                    '-1'.  Data are signed integers and will be        ****
                    packed.  The largest absolute score must be not    ****
                    greater than MAXVAL.                               ****
                                                                       ****
          41-48     MAXVAL, optional data packing in the BIGNV mode    ****
                    ('1' in cols. 17-20) and KPACK in cols. 37-40 i    ****
                    '1' or '-1'.  Punch the largest absolute number    ****
                    in the data deck.  If MAXVAL is left blank,        ****
                    then when KPACK equals 1, MAXVAL is set to 4000    ****
                    but when KPACK equals -1, MAXVAL is set to 2000    ****
                                                                       ****
     Card C.  INFORMAT cards.  Required when LFIELD differs among the  ****
variables or, if the same, exceeds 9 columns.                          ****
                                                                       ****
     A maximum of 20 cards may now be used.  If ONAMS are on data card ****
then insert 'AW' in the format statement in the columns of the data    ****
cards where the ONAMS are punched.  You can have only one 'AW', and    ****
'W' must be from 1 to 6.                                               ****
                                                                       ****
                                                                       ****
     Card D.  VNAMS cards, for inputting the names of variables.  Same ****
as in DAP but in the next to last sentence replace V001++, etc., by    ****
V00001, etc.  Then add the following new paragraph:                   ****
                                                                       ****
     Part variable list.  If you have a long list of variables most    ****
of which you want DAP2 automatically to name but some of which you     ****
want to assign special names to, punch 'PART+VARIABLE+LIST' in cols.   ****
```

1-18, and then follow this card by cards which contain a vector in ****
free field format of the special names to be input. Follow each name ****
by a comma, then by its ordinal position, and then by a comma. In the ****
last entry the last comma is replaced by a period. Insert these cards ****
before the data deck and data cards. ****
****
****

Cards E. Same as in DAP. ****
****
****

Cards F. Same as in DAP. ****
****
****

Cards G. (New) ONAMS cards for inputting the names of subjects. ****
to input ONAMS and have them punched on the data cards, insert before ****
the data card and data deck, a card punched 'SUBJECT+NAMES' in cols. ****
1-13, and follow it by cards with the NS ONAMS punched as a vector ****
in free field format. The ONAMS cards are optional. ****
****

Part subject list. If some ONAMS are missing from the data cards ****
punch 'PART+SUBJECT+LIST' in cols. 1-17, then follow this card by card ****
with a vector in free field format containing the missing ONAMS, each ****
of which is followed by a comma, then by its ordinal position, then by ****
a comma. In the last entry the last comma is replaced by a period. ****

The remainder of DAP is the same as DAP2 except as follows:

Printed output. Add the following: VAREN and ONAMS1 are now printed.

Tape output. Add the following: IPFILE and ONAMS1 are also output to IST.

Restrictions. Add to 1) number of variables: maximum is 2000 in BIGNV analysis, 2) number of subjects: maximum is 9,999.

Notes

Binary, Card B, cols. 9-12, option '-1'. This option, intended for the expert, is designed so that a previous data processor can put the score matrix in a suitable binary format for DAP2 but yet not require that the binary to BCD conversion take place, a process that may cause numerical problems. The 'END' card is not required with binary input.

MTIN, Card B, cols. 21-24. When the card input becomes too large, the bulkiness of the card input can be avoided by putting the data deck on tape with a card to tape processor such as an off-line 1401. The deck that is taped consists of the data deck, the dummy subject (if subject counting is done by DAP2), and the standard DAP2 END card ('END' in cols. 73-75). The DAP2 deck consists of all the control cards up to and including the 'DATA' card. The tape resulting from the card to tape procedure is mounted on symbolic tape MTXINI (at Berkeley FORTRAN logical tape 6(b5)). In control card B the parameter MTIN indicates the use of this auxiliary tape.

User's description of correlation program for complete data: COR2

Introduction

The object of COR2 is to calculate the means and standard deviations of all NV variables of the study, and the Pearson product moment correlation, R, between all pairs of the NV variables. Observations must be available on the data storage tape (usually via DAP) on all NS individuals on all NV variables.

The correlation matrix from COR2 is stored as CORRM1 on IST, hence any correlation matrix already calculated can be input in this file by GIST (q.v.).

Use

Card input. The only card input required is the executive control card. Punch '/COR2++' in cols. 1-7. ****
****

The intermediate storage tape. The program follows DAP and reads the following files from IST:

IDFILE	identification
VNAMS1	variable names
VSUMS1	variable sums
MEANS1	variable means

The data storage tape. The raw scores are read from the binary DST prepared by either DAP or EUCO.

The following files are output to intermediate storage tape (IST)

STDEV1	standard deviations
CORRM1	correlations

Besides general BCTRY restrictions, COR2 will not allow missing data.

User's description of correlation and covariance program for missing and complete data: COR3

Introduction

COR3 is more general (and slower) than COR2, permitting the calculation of covariances in addition to product moment R's. Of special moment is the use of COR3 in calculating R's among all variables when the number of scores on different variables may not be equal, due to missing data on some individuals. The program provides sample moment or, alternatively, unbiased estimates of the means and SD's.

Method

The formulas for R, covariance, means, and SD are standard for sample moment or unbiased estimates. The R's and covariances are compiled only for the cases common to the paired variables, but the means and standard deviations are calculated for all complete N's of each variable or for the matched pairs of objects for each R.

Use

Card input. Two cards are required, an E.C. card and one component control card.

(1) Executive control card. Punch '/COR3++' in columns 1-7.

(2) Component control card. This control card contains method and printing specifications. It is punched in integer card format and will be blank if all standard options are chosen (see following chart):

COLUMNS		CONTENTS
1-4	'0'.	To compute matched standard deviations and save on IST (standard).
	'1'.	Otherwise.
5-8	'0'.	To compute and save covariances (standard).
	'1'.	Otherwise.
9-12	'0'.	To compute and save correlations (standard).
	'1'.	Otherwise.
13-16	'0'.	To print matched standard deviations (standard)
	'1'.	Otherwise.
17-20	'0'.	To print covariances (standard).
	'1'.	Otherwise.
21-24	'0'.	To print correlations (standard).
	'1'.	Otherwise.
25-28	'0'.	For sample moment estimates (standard).
	'1'.	For unbiased estimates.
29-32	'0'.	For no replacement of missing data by the mean (standard).
	'1'.	For such replacement.

33-36	'0'.	To indicate presence of missing data.	****
	'1'.	To indicate data are complete.	****
	Blank	to use indication given in DAP as to missing data (standard).	**** ****
37-40	'0'.	To print matched means (standard).	****
	'1'.	Otherwise.	****
41-44	'0'.	To print matched NS's (standard).	****
	'1'.	Otherwise.	****
45-80		must be blank.	****

Tape input. The program follows DAP and reads the following files from the intermediate storage tape (IST):

IDFILE - identification
MEANS1 - means of individual variables (unmatched)
NVAMS1 - variable names
VAREN1 - sample size of each variable (unmatched)

Printed output. The data identification, a complete summary of methods used, and variable names are always printed. Correlations, covariances, matched means, matched NS's and matched standard deviations are printed if called for on the component control card. Since different quantities are printed as separate matrices and arrays, the user may wish an additional printout from the component program by tandem use of RLIST (which prints an array of statistics for each pair of variables).

Tape output. The following information is output to the intermediate storage tape (IST): matched standard deviations (2STDEV1), covariances (COVAR1), correlations (CORRM1), matched means (2MEAN1), and matched NS's (2SENS1).

User's description of program to reorder and delete variables of the correlation matrix: REDE

Introduction

If the analyst desires a printout of the raw correlation matrix organized for publication, for other presentation, or for study, he may employ REDE to reorder the row and column variables of the matrix, delete any variables from it, and reflect any variables in it.

Method

Since the procedure is essentially one of rearranging and/or reflecting elements of a matrix, the method may best be understood by way of an example, as follows:

Let R be a correlation matrix representing the intercorrelations among five variables, with all correlations positive.

```
                           VARIABLES
                1       2       3       4       5

     V 1    R(1,1.  R(1,2)  R(1,3)  R(1,4)  R(1,5)
     A
     R 2    R(2,1)  R(2,2)  R(2,3)  R(2,4)  R(2,5)
     I
     A 3    R(3,1)  R(3,2)  R(3,3)  R(3,4)  R(3,5)
     B
     L 4    R(4,1)  R(4,2)  R(4,3)  R(4,4)  R(4,5)
     E
     S 5    R(5,1)  R(5,2)  R(5,3)  R(5,4)  R(5,5)
```

Suppose we wish to reorder with variable 5 first, 4 second, 3 third, 2 fourth, reflect 5 and delete 1. Then the following control cards would produce the correctly reordered matrix shown below, each line indicating a new card with punches beginning in column 1.

```
'/REDE'
'0005'
'0004'
'0003'
'0002'
```
Blank card
```
'0001'
```
Blank card

The result as printed is:

```
                      VARIABLES

               -5        4        3        2
       V
       A -5   R(5,5)  -R(5,4)  -R(5,3)  -R(5,2)
       R
       I  4  -R(4,5)   R(4,4)   R(4,3)   R,4,2)
       A
       B  3  -R(3,5)   R(3,4)   R(3,3)   R(3,2)
       L
       E  2  -R(2,5)   R(2,4)   R(2,3)   R(2,2)
       S
```

Use

```
     Card input.  Three types of cards are required, an executive        ****
control card and two sets of component control cards.                   ****
executive control card.  Punch '/REDE++' in columns 1-7.                ****
                                                                         ****
     Component control cards.  Two sets of control cards are required:   ****
Cards A which indicate the new order of variables in the matrix, and    ****
Cards B, which list variables to be deleted.  A blank card must follow   ****
each set.  Cards B are optional, but the second blank card is required   ****
nonetheless.                                                             ****
                                                                         ****
     Cards A for new order (required).  (I) For each variable to         ****
be included in the reordered matrix, punch a card with the variable     ****
index in columns 1-4.  Indices should be packed right with a minus sig   ****
column one when the variable is to be reflected (e.g., the card for th   ****
3rd variable in the original matrix, reflected, would be '-003').       ****
Now order the cards so that each index's position in deck A correspond   ****
to its desired position in the reordered matrix.  (II) The last card    ****
must be blank.                                                           ****
                                                                         ****
     Cards B for deleted variables.  (I) For each variable to be         ****
deleted from the reordered matrix (in each variable not included in     ****
Cards A) punch a card with the variable index in columns 1-4.  If no    ****
variables are deleted, Cards B will contain no index cards.  (II)       ****
Insert a blank card whether or not Cards B are used.  Thus, if all      ****
variables are used, the last two component control cards will be blank  ****
```

Tape input. The program reads only from the intermediate storage tape IDFILE (identification), CORRM1 (correlations) and VNAMS1 (variable names).

Printed output. The correlation matrix is printed in the manner specified by the control cards. Columns and rows are labeled by the original variable name.

Tape output. REDE produces no changes in IST or DST.

User's description for listing statistics on correlations form missing data: RLIST

Introduction

The purpose of RLIST is to provide a rearranged printout of values computed by COR3. Although COR3 provides a printout of all values computed, each type of quantity is printed in a separate matrix. For certain purposes it is convenient to have in one place all statistics for each pair of variables, i.e., the R, matched means, matched SD's, etc.

BC RLIST provides a printout of a missing-data correlation matrix computed by bc COR3 such that each correlation coefficient is printed on a separate line. Associated with each coefficient are the names of the variables intercorrelated, matched N, matched means, and matched standard deviations. The program does no computations and has as its only purpose the user's convenience. Except for the executive control card, all input is from IST.

Use

Executive control card. Punch '/RLIST+' in columns 1-7. This is the only control card required by the program. ****
****

The following information must be on the intermediate storage tape (IST):

```
     IDFILE  data identification.
     VNAMS1  variable names (optional).
     VAREN1  sample sizes (of separate variables).
     2SENS1  matched NS's.
     2MEAN1  matched means.
     2STDV1  matched standard deviations.
     CORRM1  correlations.
```

The printed output is self-explanatory.

RLIST makes no changes in IST or DST.

User's description of suppressed variable analyses: SLEP1, SLEP2.

Introduction

There are two main purposes in using this suppression ('sleeper') program. The first is to suppress experimentally dependent variables, i.e., those that include other of the NV variables as physical components. Such variables are likely to generate communalities over unity ('Heywood cases') in factoring. The second is to implement an experimental design in which some variables are suppressed from factoring by CC5, NC2, FALS, in order to see the portion of their variance accounted for by the dimensions factored from the active ('unslept') variables. SLEP1 is the component that suppresses the designated variables. SLEP2 completely reactivates the suppressed variables after factoring is completed and recomputes all factor statistics on all NV variables in the customary 'summary' form.

Method of SLEP1

After the correlation matrix is computed (by COR2, COR3, or input), it is reordered by allocating the suppressed variables to the end rows and columns positions in the R-matrix. Only the forepart of the matrix containing the unsuppressed variables is then made available to the tandem factoring component (CC5, NC2, or FALS).

Use of SLEP1

Card input to SLEP1. Two types of cards are needed. ****
****
(1) Executive control card. Punch '/SLEP1+' in cols. 1-7. ****
****
(2) Component control cards. Punch as a vector in free field name ****
card format the names of the variables to be suppressed. ****
(Remember that unless variable names are input by the analyst DAP ****
assigns the names: 'V001,' 'V002,' etc.) ****

Tape input to SLEP1. The following files are read from IST.

IDFILE	identification
VNAMS1	variable names
CORRM1	correlations

SLEP1 prints a list of the new and old locations of the variables in the correlation matrix and a list of the suppressed and non-suppressed portions of the correlation matrix.

SLEP1 outputs the following files to intermediate storage tape (IST):

VNAMS1	variable names.
CORRM1	correlations among non-suppressed variables.
REORD1	list of new variable locations.
CORRM2	correlations among all variables.

Method of SLEP2

After the tandem factoring procedure (CC5, NC2, or FALS) is complete on the unsuppressed matrix generated by SLEP1, SLEP2 computes the factor coefficients of the variables suppressed by SLEP1, and then entirely reconstitutes the original R-matrix in its originally ordered form. Finally, the complete 'summary' factor statistics are printed for all NV variables in their reordered order, and the appropriate tape output is placed on IST in the original order of the variables.

Use of SLEP2

Card input to SLEP2. Only one card is needed. ****
****

Executive control card. This is the only control card read by ****
SLEP2. Punch '/SLEP2+' in columns 1-7. ****

The following files are read from intermediate storage tape (IST):

IDFILE	identification.
VNAMS1	variable names.
CORRM1	correlations among non-suppressed variables.
REORD1	list of new variable locations.
CORRM2	correlations among all variables.
UFACT1	unrotated factor coefficients.
CLUST1	cluster indicators (optional).
REFLX1	reflection indicators (optional).

SLEP2 prints the factor coefficients, communalities, etc., of all the original NV variables in their reordered positions. The fact that the variables have been restored to their original positions on IST is also printed.

The following files are output to intermediate storage tape (IST)

VNAMS1	variable names.
CORRM1	correlations in the original order.
DIAGV1	diagonal values.
UFACT1	unrotated factor in the original order.
CLUST1	cluster indicators based on original order (optional)
REFLX1	reflection indicators based on original order (optional).

'Optional' files will be output when SLEP2 follows NC2 or CC5.

User's description of the diagonal values program: DVP

Introduction

DVP computes or selects a set of values to be inserted as the diagonal elements, or 'self-correlations', of the NV variables in the principal diagonal of the correlation matrix. The set usually desired are estimates of the communalities though other types of values such as reliability coefficients, or unity can be substituted.

Method (see DVP in Tryon(1964a,b) and SMIS, example 3)

Use

```
     Card input.  Two types of cards are needed: an E.C. card and        ****
one component control card.  One special use requires two component      ****
control cards.                                                           ****
                                                                         ****
     (1) Executive control card.  Punch '/DVP+++' in columns 1-7.        ****
                                                                         ****
     (2) Component control cards.  A single control card suffices for    ****
all methods except PF, which reads an additional card.                   ****
                                                                         ****
     Card A.  This card specifies methods to be used, parameters         ****
for certain of the methods, and printing options.  The card is punched   ****
in integer card format except for columns 13-16, which contain decimal   ****
number as follows:                                                       ****
                                                                         ****
      COLUMNS       CONTENTS OF CARD A                                   ****
                                                                         ****
        1-4      Punch here one of the integers listed below to          ****
                 select the main method of computing                     ****
                 communalities.                                          ****
                                                                         ****
                 '30'  QF                                                ****
                 '40'  B                                                 ****
                 '41'  MOD B                                             ****
                 '50'  HIGH R                                            ****
                 '52'  TRIADS                                            ****
                 '60'  PF                                                ****
                 '70'  ZERO                                              ****
                 '71'  UNITY                                             ****
                                                                         ****
        5-8      In the case of QF, punch here another integer           ****
                 from the list to select a preliminary                   ****
                 method,  otherwise leave blank.                         ****
                                                                         ****
        9-12     Used by QF method only.  Punch the number of            ****
                 predictor variables to be used with each variable       ****
                 to compute the communality estimate.  Restriction:      ****
                 ten variables at most can be used.  If these            ****
                 columns are left blank, the number of predictors        ****
                 will be either ten or one less than half the number     ****
                 of variables in the matrix, whichever is smaller.       ****
        13-16    Used by QF method only.  Punch a decimal             ****
                 number which is the convergence criterion for        ****
                 QF (normally '.010').  The decimal point             ****
                 must be punched.                                     ****
                                                                      ****
        17-20    Used by QF method only.   '1' for printing           ****
                 of the successive estimates of communality in        ****
                 iteration.  Estimates are not printed if             ****
                 this field is blank.                                 ****
                                                                      ****
        21-24    Blank.                                               ****
                                                                      ****
        25-28    Used by B and MOD B methods only.  '1'               ****
                 causes the squared index of proportionality for each ****
                 pair of variables to be printed.  The field is       ****
                 left blank if this printing is not desired.          ****
                                                                      ****
                                                                      ****
     Card B.  This card is required by PF method only, as follows:    ****
                                                                      ****
      COLUMNS       CONTENTS OF CARD B                                ****
                                                                      ****
        1-4      The number of reference variables NS(RV), not        ****
                 counting the variable whose communality they         ****
                 determine.  Limitation:  the condition NS(RV) must   ****
                 be between 3 and 9 inclusively, otherwise NS(RV)     ****
                 is set to 9 or (NS(RV)-1)/2, whichever is smaller.   ****
                                                                      ****
        5-8      The number of iterations desired, NS(I).  Limitation: ****
                 NS(I) must be between 1 and 10 inclusively.  If NS(I) ****
                 is zero or blank it is set to 4, if NS(I) is         ****
                 greater than 10, it is set to 10.                    ****
```

Intermediate storage tape. The following information is read from IST.

data identification	IDFILE
correlation matrix	CORRM1

The printout contains a complete description of methods and parameters chosen by the analyst and a list of diagonal values. In the case of the high R method, the ten highest R's for each variable are printed in their rank order.

The only file written on the intermediate storage tape is DIAGV1.

Introduction to factoring (independent dimensional analysis)

The objective of factoring is to reduce the NV observed variables to a small number of independent composites of the NV variables that reproduce (A) the intercorrelations among the NV variables (the off-diagonal elements of the R-matrix given in COR2 or COR3) and (B) specified portions of their variances, usually their communalities (the diagonal elements of the R-matrix computed by DVP).

All forms of factoring are variants of the same general method of independent dimensional analysis (Tryon, 1959,1964a). The method is a sequence of 'regions of decisions' that define the logic of dimensional analysis. Each factoring variant is simply a special sequential pattern of decisions in this logic.

BCTRY includes three component programs of factoring: CC5, NC2, and FALS. As a group they permit the user to perform all the main forms of cluster and factor analysis. The specific pattern of decisions that forms each of the three main component programs is described in its own methods section of its user's description below. It is the selection of a given program and of the particular pattern of options taken within the program that defines each of the special types of cluster and factor analysis.

The 'preferred' solution. The CC program has the greatest generality, being capable of performing all types of factoring except least squares and averoid. To the user who wishes a blind, empirical factoring procedure and one which by use of GIVE cards permits him to explore a variety of alternative types of factoring of his data, a standard key cluster CC solution is the preferred solution. This standard solution is made quite simple for the user by the fact that only blank component control cards are necessary (q.v.). Preceded by DVP 50 (or 60), and followed by CSA and SPAN, with blank control cards in all of them, this standard key cluster solution provides a complete full cycle V-analysis, including orthogonal and oblique structure analysis.

Orthodox factor analysts are likely to urge principal axes factoring (PFA in FALS) as the 'preferred' solution (e.g., Harman, 1960, p. 179) because of its least squares variance maximizing feature. Actually, this method has narrow usefulness. The results from it are normally quite unintelligible; its 'factors' must be rotated; and it has other difficult features as indicated in the Method section of the user's description of PFA in FALS.

Refactoring to improve a solution. If the initial factoring by CC5, NC2, or FALS should give an unsatisfactory solution, an improved solution is usually possible (on the GIVE input).

(1) When residuals are too large. Call '/FAST', 'residual'. then follow fast by DVP50, and continue the factoring of the residuals by whichever factoring program you wish (NC2, CC5, FALS).

(2) When communalities are too large or exceed 1.00 ('Heywood cases'), (A) repress experimentally-dependent variables, if any, by SLEP1. (B) CC5 or FALS: iterate factoring only once. (C) CC5 or NC2: examine the correlations (or residuals) among the definers of each dimension, and in presetting definers for refactoring delete definers with trivial residuals, or a whole dimension if the residuals among its definers are trivial. A variable with too large a communality is often one relatively highly collinear with the definers of a given dimension but not included as a definer. Locate the dimension on which such a variable shows a sudden high jump in partial communality and add it as a preset definer to the dimension in the preset rerun of the solution.

Most important: in the beginning of the 'Summary', check carefully the list of dimension-definers for overlappers and delete any overlaps from the later dimensions in which they appear a second time as definers.

(3) When improved sets of pivotal dimension-definers can be selected. Following CC5 or NC2, study CSA and SPAN and be ready to rerun on preset definers if an improved set is obvious. Reruns are inexpensive. The importance of preset refactoring is also emphasized in O-analysis. For the way it is utilized there, see EUCO, the section titled '(4) Locating the general O-types by factoring and structure analysis.'

User's description of cumulative communality key cluster analysis: CC5

Introduction

This general program of independent dimensional analysis (factoring) permits a selection of a subset of variables as the defining variables of each dimension, computes factor coefficients of all variables on each dimension and a variety of other statistics relating to factoring. The analysis starts with predetermined diagonal values in the correlation matrix.

In standard form it selects the most independent sets of defining variables that are collinear (have proportional R's), cumulates partial communalities as factoring proceeds, terminates factoring on the dimension at which the sum of communalities converge. The program is usually followed by the program CSA and then SPAN, which graphs the results of the CC5 analysis.

Options on control cards also permit a variety of other forms of multidimensional analysis: centroid (CENT) factoring, square root (PV) factoring, factoring with unities or reliability coefficients as diagonal values, bifactor analysis, many forms of rational analysis, and several varieties of object cluster and inverse factor analysis.

Method

This general program of independent dimensional analysis (factoring) permits a selection of a subset of variables to be the definers of each of K dimension of a problem. Each dimension is defined as an independent 'structured' domain under model I (see CSA), the consequence of which is that communalities (DIAGV1) as diagonal elements are required in the correlation matrix. Factor coefficients are computed by simple sums.

Options in CC5 provide an opportunity to perform a wide variety of analyses among which are:

(1) Empirical key cluster analysis. The program itself selects the definers of the K dimensions.

(2) Preset key cluster analysis ('multiple group factoring'). The analyst inputs the definers, selected on rational grounds (RC analysis) or as a revision of a prior CC run (PC analysis).

(3) Preset dimension analysis. The analyst presets the number K of dimensions desired but lets the program select the definers.

(4) Pivot variable (PV) analysis ('square root factoring'). Each dimension is defined by a single variable. CC5 is the only BC TRY program that performs PV analysis.

(5) Centroid (CC-CENT) analysis. Each dimension is defined by all NV variables, as in traditional Thurstone factoring, or by the most general set, as in salient centroid factoring.

(6) Bifactor analysis. The first dimension is defined by a general centroid, the remaining by collinear subsets, as in traditional Spearman-Holzinger analysis.

(7) Sleeper analysis. The suppressing of selected variables during factoring but a later activating of them in order to compute all factor statistics on them and to include them in later analyses is achieved by linkage with SLEP (q.v.).

(8) Ordered and designed analyses. The sequential linking of analyses is performed, as in tandem CC analyses, i.e., ordering and sleeping blocks of variables in a sequence of runs according to a rational experimental design.

(9) Total variance and reliable variance factoring. Factoring is performed with unities or reliabilities, respectively, as diagonal values.

(10) Hierarchical ('higher-order') analysis. CC5 is followed by a set of tandem CSA analyses progressively combining the most collinear oblique cluster composites.

(11) CC5 is a preliminary analysis to oblique structural analysis (CSA), to a geometric configurational analysis (SPAN), to comparing the dimensions with those found in other groups (COMP), and to O-analysis (FACS and EUCO).

The decision pattern of CC5

Since all forms of factoring represent a particular pattern of decisions across the different regions of decisions that embody the general case of independent dimensional analysis (Tryon, 1959), the particular pattern that generates CC factoring may be described under these regions, as follows:

(1) Correlation matrix to be reproduced by factoring. This matrix must be made available from either COR2 or COR3.

(2) Diagonal values to be reproduced by factoring. These values are usually approximated by high R or PF communalities (see DVP), though reliabilities or unities may be chosen (see DVP).

(3) The defining variables of each dimension. For any independent dimension, these are a selected subset of the NV variables designated as V1, V2,...,VI. The analyst can select them in four ways.

(A) Empirically-derived mutually-collinear subset. The program selects as a pivot variable the one with highest variance of squared R's, then adds additional definers most collinear with it and with each other. Criteria of mutual collinearity are employed, and if a given subset fails to meet these criteria, a second selection is made. If after four 'tries' no subset meets the criteria, factoring terminates.

(B) Preset. The analyst expressly designates the definers of each of the K dimensions.

(C) A subset of a single definer of each dimension. The analyst presets it or it is selected empirically by one of several optional criteria.

(D) Total set of NV variables (max NV=20) the most salient general subset. The analyst designates the most general by either of two criteria.

(4) Reflection of defining variables. The submatrix of defining variables, used in the calculation of factor coefficients, is systematically reflected. For S equal to or less than 10, the reflection that makes the sum optimal is the final reflection pattern chosen. For S greater than 10, the refection pattern is selected on first power R's.

(5) Factor coefficients and partial communalities. CC5 defines dimensions under model I, hence communalities are required in all calculations.

(A) Unaugmented. Factor coefficients are calculated by the simple sum formula, and the partial communalities are simply their squares.

(B) Augmented. The 'normalized' values are also calculated.

(6) Residuals. After factoring each dimension the residuals are calculated.

(7) Number of dimensions (terminating criteria). The number is preset by the analyst or empirically determined by one of two criteria.

(A) Factoring is terminated when a preset per cent of the initial estimates of total communalities (the diagonal values) is reproduced by factoring.

(B) Factoring terminates when a preset per cent of total squared R's (the off-diagonal elements) are reproduced by factoring.

(8) Number of iterations. After the first factoring is completed, the factoring procedure is iterated on the same dimension but uses the new values of communalities derived from the first factoring. Reiteration terminates on the iteration which, at the option of the analyst is:

(A) the iteration preset by him;

(B) the iteration that yields converged communalities at an

indicated decimal point;
(C) the 30th iteration in the case of non-convergence by the 30th try.

(9) Sleeper option. If the analyst wishes to suppress any variable or block of variables to prevent their being definers of a dimension, he calls SLEP1 after calculating the correlation matrix. By Calling SLEP2 after CC5 analysis, the sleepers are activated and all finally iterated factor statistics are computed on them.

Use

Card input. One E.C. card and two component control cards are ****
needed. ****
****

(1) Executive control card. Punch '/CC5+++' in columns 1-7. ****
****

(2) Component control cards. Ordinarily the program requires ****
only two component control cards, both of which will be blank when ****
standard values and parameters are used. Additional cards are ****
necessary for one of the control options. ****
****

Card A. Dimensions, pivots, definers, presetting, iterations, ****
residuals. This integer card specifies the analyst's decisions on ****
what options or criteria are to be used in the analysis, as follows: ****
****

COLUMNS AND CONTENTS OF CARD A ****
****

1-4 Number of dimensions. ****
'1' by communality exhaustion criterion (standard). ****
'2' by sum of squared R exhaustion criterion. ****
'3' by communality increment criterion ****
'4' preset by analyst (in columns 17-20) or PV or ****
salient centroid (cols. 9-12). ****
****

5-8 Pivot variable selection. ****
'1' select variable with highest variance of squared ****
R (standard). ****
'2' the key variable selection method chosen also ****
determines the pivot variable (cols. 9-12). ****
****

9-12 Selecting defining variables of each dimension. ****
****

Collinear subset (key cluster): ****
'1' variables most collinear with a single pivot ****
variable. ****
'2' most mutually collinear subset (standard). ****
****

Preset: ****
'3' selected by analyst via input cards. ****
'4' selected from IST as the defining subset of ****

variables produced by a preceding NC analysis. ****

```
        Pivot variable (PV) (if not preset):                        ****
        '5' the single pivot variable having the highest            ****
              residual communality.                                 ****
        '6' the single pivot variable having highest sum of         ****
              squared correlations.                                 ****
        '7' the single pivot variable having highest variance       ****
              of squared correlations.                              ****
                                                                    ****
        Salient centroid.  (If NV=20 or less, all NV variables      ****
        are routinely selected as definers).  When NV exceeds 20    ****
        the selected set of definers of each dimension are:         ****
        '8' the 20 variables having highest variances of squared    ****
              correlations.                                         ****
        '9' the 20 variables having highest sums of squared         ****
              correlations.                                         ****
                                                                    ****
13-16   Reflecting the submatrix of defining variables of each      ****
        dimension.                                                  ****
        '1' reflection is optimal if number of definers is 10 or    ****
              less, otherwise by first-power method (standard).     ****
                                                                    ****
17-20   Number of preset dimensions (when cols. 1-4 punched '4').   ****
        If the number of dimensions is to be preset, punch the      ****
        desired number.  If not, punch a '0' (standard).  Note      ****
        that the selection of preset definers via control cards     ****
        C (determined by a '3' in columns 9-12 of control card      ****
        A) requires that the number of dimensions be preset here.   ****
                                                                    ****
21-24   Number of iterations.  Punch '0' to allow                   ****
        iterations to be stopped by satisfaction of the commun-     ****
        ality difference criterion (punched in columns 17-24 of     ****
        Card B).  But if a fixed number of iterations is desired,   ****
        punch the number in this field.  Four iterations are        ****
        usually sufficient (standard).                              ****
                                                                    ****
25-28   Not used.  Leave blank.                                     ****
                                                                    ****
29-32   Maximum number of successive pivot variables to be tested   ****
        as pivots of subsets that meet PSQ criterion (punched       ****
        in Card B, cols. 25-32).  Punch '4' (standard) for a        ****
        maximum of 4, or punch whatever specific maximum is         ****
        desired.                                                    ****
                                                                    ****
33-64   Not used.  Leave blank.                                     ****
                                                                    ****
65-68   Printout of residual matrix after each dimension.  To       ****
        obtain this printout - which may be voluminous - punch      ****
       '1'.  To suppress it, punch a '0' (standard).                ****
                                                                    ****
69-72   Printout of the factoring on each successive iteration.     ****
        To obtain this (voluminous) printout, punch '1'.  To        ****
        suppress it, punch '0' (standard).                          ****
```

```
     73-80   Not used.  Leave blank.                                     ****
                                                                          ****
     Card B:  criterion values.  This card specifies criterion values    ****
for terminating factoring and iteration and for accepting definers      ****
of dimensions.  Punch all values in decimal card format.  If an         ****
option is not being used and its criterion is therefore not applicable  ****
leave it blank on the card.  Standard values are given in parentheses.  ****
The contents are as follows:                                            ****
                                                                          ****
COLUMNS AND CONTENTS OF CARD B                                          ****
                                                                          ****
     Accepting more than four definers of a dimension:                  ****
     1-8     Index of proportionality (PSQ) criterion for accepting      ****
             more than four defining variables of a dimension (0.81).   ****
             See columns 9-12, Card A, '1' or '2'.                       ****
                                                                          ****
     Terminating factoring:                                             ****
     9-16    Exhaustion criterion to be met by the last (Kth)           ****
             dimension (.92).  See columns 1-4 of Card A, '1' or        ****
             '2'.                                                        ****
                                                                          ****
     Terminating iteration:                                             ****
     17-24   Communality difference criterion for terminating iteration ****
             (0.01).  See columns 21-24, Card A, '0'.                    ****
                                                                          ****
     Accepting the most collinear subset of four definers of a          ****
     dimension:                                                         ****
     25-32   Index of proportionality (PSQ) criterion for accepting the ****
             second, third, and fourth defining variables of a          ****
             dimension (0.4).  See cols. 9-12, Card A, '1' or '2'.       ****
                                                                          ****
     33-80   Not used.  Leave blank.                                     ****
                                                                          ****
                                                                          ****
     Card C.  Specifying definers of preset dimensions.  These cards    ****
specify defining variables.  They are needed only when the defining     ****
variables of each dimension are input by the analyst, i.e., when        ****
columns 9-12 of Card A contain a '3' and columns 17-20 of Card A        ****
contain the preset number of dimensions.  The indices of the variables  ****
defining dimension 1 must be punched in the first card, those defining  ****
dimension 2 on the second card, etc., for all dimensions.  The          ****
indices are punched in a format identical to the integer card format    ****
except that columns 73-80 may also be used.  Do not punch a negative    ****
sign before any preset definer; punch its variable number only.         ****
After SLEP1, the indices of preset definers of CC, NC, CSA, NCSA are    ****
the reordered ordinal numbers as printed in SLEP1.                      ****
                                                                          ****
     The following files are input from the intermediate storage tape:

          IDFILE     Identification.
          CORRM1     Correlations.
          DIAGV1     Diagonal values.
          REFLX1     Reflection indicators (only if columns 9-12 of
                     Card A contains a '4').
```

Printed output. Appropriate titles and headings are printed with all printed statistics and information.

Tape output. The following files are written on IST:

DIAGV1	Diagonal values.
UFACT1	Unrotated factor coefficients.
CLUST1	Cluster indicators (without reflection indicators)
REFLX1	Reflection indicators.

Besides the general restrictions of the system, the number of defining key variables of a dimension may not exceed 20.

User's description of non communality key cluster analysis: NC2

Introduction

The program performs a form of key cluster analysis virtually identical to that of CC5 except that the factor coefficients are calculated without the use of diagonal elements. Initial estimates of communality may be used to establish a criterion to terminate factoring, but the diagonal elements in the correlation matrix are ignored in the factoring. Factoring is not reiterated. NC2 includes averoid factoring as a special case.

Limitations of NC factoring. Experience has shown that NC2 may give a poor solution unless NV is large and the variables show substantial correlation. (See factoring above, last section on 'refactoring.') When the variables are test items, whose inter-R's are usually quite low, use CC5 with estimated communalities form DVP (high). Generally, however, NC2 is excellent as a preliminary (and often final) factoring procedure and is especially suited to factoring the EUCO matrix in O-analysis (q.v.).

Method

NC factoring is so similar to CC factoring that only the points in which it deviates from CC will be given here. These deviations result from the defining of each NC dimension (or 'factor') under model II as a composite score on a domain of variables from which the observed defining variables of the dimension are a representative sample. The result is that no individual communalities for the variables are required as diagonal values in the correlation matrix being factored.

Without communalities in diagonal cells, NC factoring becomes a high speed factoring procedure, for the more time-consuming DVP programs necessary for close initial approximations to communalities (e.g., PR values) are avoided, and no iterations of factoring to convergence of communalities, as in CC and FALS factoring, are made. But empty diagonal cells in the submatrix of defining variables of an NC dimension introduce a hazard, namely, that the mean inter-R's or residuals between the definers may approach zero or become negative, whence factor coefficients become indeterminate or imaginary. Hence, NC factoring is most suitable in problems where only the most salient dimensions of high generality are desired.

Referring to the 11 general kinds of analysis listed in the introduction of CC factoring, all can be performed by NC analysis except:

(4) Pivot variable analysis.
(9) Total variance or reliable variance factoring.
(11) For oblique structure analysis one should usually follow NC

analysis by CSA and not by NCSA in order to avoid indeterminate or imaginary oblique factor coefficients.

Referring also to the 9 general regions of decision listed in the CC description, the particular decision pattern which generates CC factoring, all apply to NC factoring except:

(2) That a DVP program (usually high R) is used only to form a criterion by which to terminate factoring.

(3c) That single variable definers cannot be selected in NC analyses.

(8) That no iterations of NC factoring are possible (except by calling up CC after NC).

Use

```
     Card input.  One E.C. and two component control cards are needed.    ****
                                                                           ****
     (1) Executive control card.  Punch '/NC2+++' in cols. 1-7.            ****
                                                                           ****
     (2) Component control cards.  The two control cards described         ****
below contain quantities required by NC2 for such things as selecting      ****
key variables, terminating factoring, etc.  Most of these quantities       ****
have a 'standard value' which NC2 sets automatically if the field is       ****
left blank.  Thus to obtain a completely 'standard' NC2, both              ****
component control cards would be left blank.                               ****
                                                                           ****
     Card A:  Dimensions, presetting, residuals, definers.  This card,      ****
punched in integer card format, specifies the analyst's decisions on       ****
what options or criteria are to be used in the analysis.                   ****
                                                                           ****
     COLUMNS AND CONTENTS OF CARD A                                        ****
                                                                           ****
     1-4     Determination of the number of dimensions.                    ****
             '1'.  By communality exhaustion criterion (standard).         ****
             '2'.  By sum of squared R exhaustion criterion.               ****
             '3'.  By communality increment criterion.                     ****
             '4'.  Preset by analyst (the number of dimensions must be     ****
                   given in cols. 9-12), or salient centroid (col. 17-     ****
                   20).                                                    ****
                                                                           ****
     5-8     Calculating and printing communality exhaustion statistic     ****
             for terminating factoring.                                    ****
             '1'.  Dimensionality is to be determined by method other      ****
                   than communality exhaustion (i.e., presetting).         ****
                   Statistics are neither computed nor printed.            ****
             '2'.  Statistics are computed and printed even if they ar     ****
                   not used to determine dimensionality (standard).        ****
     9-12    Number of preset dimensions.                              ****
             If the number of dimensions is to be preset, punch the    ****
             desired number.  If not, punch a '0' (standard).  Note    ****
             that the selection of defining variables of dimensions vi ****
             control cards C (determined by a '3' in cols.  17-20 of   ****
             control card A) requires that the number of dimensions be ****
             preset.                                                   ****
                                                                       ****
     13-16   Printing of residual correlation matrix.                  ****
             '0'.  Residuals printed only after last dimension         ****
                   (standard).                                         ****
             '1'.  Residuals printed after every dimension.            ****
                                                                       ****
     17-20   Method of selecting key defining variables of each        ****
                   dimension.                                          ****
                                                                       ****
          Collinear subset (key cluster).                              ****
             '1'.  Variables most collinear with pivot variable.       ****
             '2'.  Most mutually collinear subset (standard).          ****
                                                                       ****
          Preset.                                                      ****
             '3'.  Selected by analyst via input control cards C.  See ****
                   Cards C below.                                      ****
                                                                       ****
          Salient centroid (if NV=20 or less), all NV variables are    ****
          definers.                                                    ****
             '8'.  All variables ordered by variance of squared        ****
                   correlations and highest 20 taken as defining key   ****
                   variables (averoid method) of each dimension.       ****
             '9'.  All variables ordered by sum of squared correlation ****
                   and highest 20 taken as defining variables of each  ****
                   dimension.                                          ****
                                                                       ****
     21-24   Maximum number of successive variables to be tested as    ****
             pivots of subsets that meet PSQ criterion (punched in Car ****
             B, cols.  33-40).  Punch '4' (standard) for a maximum of  ****
             4, or punch whatever specific maximum is desired.         ****
                                                                       ****
     25-72   Not used.  Leave blank.                                   ****
                                                                       ****
     Card B:  criterion values.  This specifies criterion values       ****
needed in the analysis.  The values are to be punched in decimal card  ****
format.  'Standard' criterion values are given below in parentheses.   ****
If an option is not being used and its criterion is therefore not      ****
applicable, leave it blank on the card.                                ****
                                                                       ****
COLUMNS AND CONTENTS OF CARD B                                         ****
                                                                       ****
     Accepting more than four definers.                                ****
     1-8     Index of collinearity (PSQ) criterion value for accepting ****
             more than four defining variables of a dimension (0.81).  ****
             See cols. 17-20 of Card A, '1' and '2'.                   ****
     Terminating factoring:                                              ****
     9-16    Dimensionality criterion value to be met by the last        ****
             dimension (0.92) for either communality or sum of           ****
             R-squared exhaustion criteria.  See cols. 1-4 of Card A,    ****
             '1' or '2'.                                                 ****
                                                                         ****
     17-24   Dimensionality criterion value to be met by last dimension  ****
             (0.95) for the communality increment criterion.  See cols   ****
             1-4 of Card A, '3'.                                         ****
                                                                         ****
     25-32   Dimensionality criterion value for a sum of R-squared test  ****
             which takes precedence over all other tests (0.999).  This  ****
             is to prevent unnecessary factoring.                        ****
                                                                         ****
    Accepting the most collinear subset of four definers:                ****
     33-40   Index of collinearity criterion for accepting the second,   ****
             third, and fourth defining variables of a dimension (0.4)   ****
             See cols. 17-20 of Card A, '1' or '2'.                      ****
                                                                         ****
     41-72   Not used.  Leave blank.                                     ****
                                                                         ****
     Cards C.  Dimension-definer cards.  These cards are needed only     ****
when the defining variables of each dimension are to be input by the     ****
analyst (that is, when cols. 17-20 of Card A contain a '3').  The        ****
indices of the defining variables are punched in a format exactly like   ****
the integer card format, except that cols. 73-80 may be used if          ****
necessary.  Do not punch a negative sign before any preset definer;      ****
punch only its variable number.  After SLEP1, the indices                ****
of preset definers of CC, NC, CSA, NCSA, are the reordered ordinal       ****
numbers as printed in SLEP1.                                             ****
```

The following data are read from the intermediate storage tape (IST):

IDFILE	Identification.
CORRM1	Correlations.
DIAGV1	Diagonal values (if communalities are used as a means of determining dimensionality).

Printed output is self-explanatory.

NC2 writes the following files on the intermediate storage tapes (IST):

DIAGV1	Diagonal values.
CLUST1	Cluster indicators (without reflection indicators).
UFACT1	Unrotated factor coefficients.
REFLX1	Reflection indicators.

Besides the general restrictions of BCTRY, there is a restriction that the maximal number of defining variables of a dimension is 20.

User's description of principal axes, canonical, and augmented factoring: FALS (PFA, CANON, AUG)

Introduction

The three optional methods that comprise FALS are identical in purpose to CC5 factoring except in the following two major particulars. (1) Like centroid factoring (CC5-CENT), they are all special cases of defining each dimension by the total set of all NV variables; and (2) instead of calculating factor scores by simple sums, they employ a least squares solution. They differ from each other largely in the fashion in which the least squares factor coefficients are computed.

The restriction of defining each dimension by the total set precludes PFA, CANON, and AUG factoring (like CENT factoring) from performing the following types of analyses: subset analysis, square root (PV) factoring, bifactor analysis, ordered, and hierarchical analyses, and an oblique solution (see CSA). Furthermore, like CENT, after the first dimension, all later dimensions are bipolar, and in real problems many are imaginary (i.e., producing negative squared factor coefficients). To be interpretable, PRIN, CANON, and AUG dimensions usually must be rotated either orthogonally (as by quartimax or varimax given in GYRO) or obliquely.

Analyses that can be performed by these components are the following: presetting the dimensionality, designed analysis through the use of SLEP, total and reliable variance factoring, and a geometric description via SPAN. O-analysis by FALS-EUCO, as tandem programs, uses regression scores from FACS. Subsequent comparative analyses can be made by use of COMP or SIMRO.

The distinctive feature of principal axes of PFA factoring is that it is formulated to maximize the sums of squared factor coefficients on successive dimensions, and it can factor any symmetric matrix whether or not the elements are in the R metric. Canonical (CANON) or CFA factoring computes factor coefficients as least squares regression weights that maximize the multiple R between the dimensions and the variables, and also, being a maximum likelihood solution provides, under certain restrictive assumptions, for a chi-square test of the significance of residuals after any preset dimension. Augmented (AUG) or AFA factoring is PFA factoring applied to the augmented correlation matrix, i.e., factoring in 'common factor space' (also called 'alpha factor analysis' by Kaiser and Caffrey (1965).

Method

The basic formulation of PFA is a set of complex characteristic equations whose roots, called 'Eigenvalues,' when properly scaled, give the sum of squared factor coefficients on each dimension; corresponding to each Eigenvalue of a given dimension are the 'Eigenvectors' which, when scaled, are the factor coefficients of the variables. This formulation results in the need for extensive iteration in order to converge on final values - usually time-consuming even on a computer, though our routine HOW is a fast one. Any advantage from maximizing factor coefficients may be lost through the necessity of initially estimating communalities.

Canonical factor analysis formulates the K dimensions as 'criteria,' the NV tests as 'predictors,' and the factor coefficients are the multiple regression weights of the dimensions on variables. The first dimension is defined by a pattern of weights that maximizes the squared multiple R (a least square solution) between the dimension and the variables. From the first residuals, the second dimension is similarly defined, and so on to the Kth dimension, which can be selected as the last one deemed to be 'significant,' i.e., yield a distribution of residuals classed as significant on a chi-square test.

If the analyst wishes to perform a PFA analysis, as described above, on the augmented correlation matrix, the component AUG executes the analysis. Since the communalities of the variables are used to augment the raw correlation matrix, which is then factored, each variable is set to have equal variance to be partitioned by factoring the augmented matrix. This means that the augmented R's of variables of very low communality are given equal weight (namely, unity) in the factoring to those of high. Good estimates of communalities are therefore quite essential in this analysis, otherwise poor estimates of augmented R's would result, and the findings would become difficult to comprehend.

Use

Card input. One E.C. Card and four component control cards are needed.

(1) Executive control card. Punch '/FALS++' in cols. 1-7.

(2) Component control cards. These cards specify the method to be used, provide control over iteration, and set the methods and parameters used to estimate the number of dimensions. The standard options are given in parentheses. When the standard option differs for the three methods three separate standard options are given. Blank columns yield standard options.

Card A: method control card. This card specifies which of the three methods of factoring (PFA, CFA, AFA) are to be used.

COLUMNS AND CONTENTS OF CARD A

1-3	'PFA'. Use principal factor analysis, or PRIN.
	'CFA'. Use canonical factor analysis, or CANON.
	'AFA'. Use augmented factor analysis, or AUG.

Card B: iteration control card. This card provides the user with complete control over the number of times dimensionality will be estimated, and for each estimate of dimensionality the number of times communality will be estimated by refactoring. Card B is punched in an integer card format.

COLUMNS AND CONTENTS OF CARD B

1-4 Specifies the maximum number of times dimensionality is t be estimated empirically by the program. No more than four estimates can be made, hence the contents are betwee zero and four.
- '0'. The dimensionality is not estimated empirically. Instead the number of dimensions is preset on Card C.
- '1'. The dimensionality is estimated once empirically (standard).
- '2','3','4'. The number of dimensions is estimated two, three, or four times respectively, unless two successive estimates are equal, in which case no further estimates are made.

5-8 Specifies the maximum number of communality iterations to be performed, using either
(1) the number of dimensions preset by the analyst when cols. 1-4 of Card B contain '0' or
(2) the number of dimensions obtained from the first empirical estimation when cols. 1-4 of Card B contain '1', '2', '3', or '4'. (PFA standard: '1'), (CFA standard: '1'), (AFA standard: '1').

9-12 Specifies the maximum number of communality iterations to be performed using the number of dimensions obtained from the second empirical estimate of communality. If cols. 1-4 of Card B contain a number less than two, leave blank (standard).

13-16 Specifies the maximum number of communality iterations using the number of dimensions obtained from the third empirical estimate. If cols. 1-4 of Card B contain a number less than three, leave blank (standard).

17-20 Specifies the maximum number of communality iterations using the number of dimensions obtained from the fourth empirical estimate. If cols. 1-4 of Card B contain a number less than four, leave blank (standard).

Card C: preset dimensionality. This card contains either (1) the number of dimensions preset by the analyst (when cols. 1-4 of Card B contain '0' or (2) an upper limit on the number of dimensions estimated by empirical methods (when cols. 1-4 of Card B contain a number greater than '0'). The value should be punched in cols. 1-4 in an integer card format (see Glossary). (Standard: 15, or one half the total number of variables, whichever is smaller). Blank columns yield standard options. In CANON, however, the user should preset the

```
dimensionality;  a prior NC analysis will give a good estimate.           ****
                                                                          ****
     Card D:  criteria for empirical estimates of dimensionality.         ****
This card permits the user to specify the empirical method or methods     ****
to be used to estimate the number of dimensions and the criterion         ****
values associated with each method.  The methods mentioned below are      ****
all discussed in the program design.  If more than one method is used,    ****
the method yielding the smallest dimensionality takes precedence.  Car    ****
D should be punched in a decimal card format.  Blank columns yield        ****
standard values.                                                          ****
                                                                          ****
     COLUMNS AND CONTENTS OF CARD D                                       ****
                                                                          ****
     1-8     Minimal Eigenvalue or minimum communality method.  If a      ****
             decimal number greater than zero is punched, the estimate    ****
             of the dimensionality is the number of dimensions whose      ****
             Eigenvalues (or sums of partial communality over all         ****
             variables) are greater than the decimal number.  Punch       ****
             '0' if you do not wish to use this method.  (PFA             ****
             standard:  .0; CFA standard:  1.33; AFA standard:            ****
             0.90).                                                       ****
                                                                          ****
     9-16    Communality exhaustion method.  If a decimal number          ****
             between zero and one is punched, the estimate of the         ****
             dimensionality is the smallest number of dimensions which    ****
             exhaust the proportion of total communality specified by     ****
             the decimal number.  Punch '0' to suppress this method.      ****
             (PFA standard: 0.95, CFA standard: 0.0, AFA                  ****
             standard: 0.99).                                             ****
                                                                          ****
     17-24   Residual correlation method.                                 ****
             If a decimal number between zero and one is punched the      ****
             dimensionality is the smallest number of dimensions for      ****
             which the proportion of the original off-diagonal sum S o    ****
             squared correlations present in the residual matrix does     ****
             not exceed the value of decimal number.  Punch '0' to        ****
             suppress this method  (standard: 0.0).                       ****
                                                                          ****
     25-32   Reproduced correlation method.                               ****
             If a decimal number between zero and one is punched, the     ****
             dimensionality is the smallest number of dimensions that     ****
             reproduces all of the mean squared original correlation      ****
             except for the proportion specified by the decimal number    ****
             Punch '0' to suppress this method (standard: 0.005).         ****
```

Tape input. The program reads the identification file (IDFILE), correlation matrix (CORRM1), and diagonal values (DIAGV1) from the intermediate storage tape. These files are required input for all control options.

Printed output. Before the factoring is begun the program prints out a description of the analysis to be performed, including the values of all control constants and criteria. For each factoring a number of statistics are printed, with appropriate labels.

For each estimation of the number of dimensions, the new estimate is printed, as is a comment noting which of the above seven conditions resulted in the estimate.

After factoring has been terminated, a summary of the analysis is printed with appropriate labels. The summary includes the final Eigenvalues, Eigenvectors, factor coefficients, augmented factor coefficients, and cumulative partial communalities.

Tape output. FALS output to IST is the same for all control options. The final communalities are written in DIAGV1 and the final factor coefficients are written in UFACT1.

In addition to the usual BCTRY system restrictions the following maximum allowable values should be noted:

Maximum number estimates of dimensionality	4
Maximum number of factorings, that is, the sum of the contents of cols. 1-20 of Card B.	50
Initial number of preset dimensions	15

Important: always call '/FAST' with its subcomponent 'residual' after FALS. Otherwise you will not have a record of the residual matrix after FALS factoring. Also, be sure to call 'GIVE' after FALS. The reason is that, in order to get good SPAN diagrams, some of the FALS dimensions may have to be reflected by inputting UFACT1 via GIST with reversed signs on these dimensions and then recalling SPAN.

User's description of program to compute residuals and the reproduced correlation matrix: FAST

Introduction

An analyst usually tests the degree to which the results of a particular factoring procedure fit the actual correlation matrix by either of two methods, or by both: he reproduces the observed R's from the factor coefficients, or he computes the residuals, i.e., the differences between observed and reproduced R's. Though one or both methods are routinely part of the CC5 and NC2 factoring programs of BC TRY, a printout of the matrices is not provided by FALS. FAST is an independent component that provides such printouts. If an analyst wishes to continue factoring on the residuals from a prior analysis, FAST sets them on IST (CORRM1) for this purpose. It is also useful in providing these statistics on published factor analysis, where such findings are not usually found. He does so by inputting the appropriate files on IST by GIST (i.e., IDFILE, UFACT1, CORRM1).

Use

```
     The program requires two control cards which must be input in the    ****
following order:                                                          ****
                                                                          ****
     (1) Executive control card.  Punch '/FAST++' in columns 1-7.         ****
                                                                          ****
     (2) Component control card.                                          ****
     This card provides two control constants:  the first determines      ****
what statistic will be calculated;  the second controls the output of     ****
this statistic.                                                           ****
                                                                          ****
     Residual matrix (without diagonal entries).                          ****
                                                                          ****
     COLUMNS AND CONTENTS                                                 ****
                                                                          ****
     1-16    'Residual'  (begin punching in col. 1).                      ****
                                                                          ****
     17-20   '0'.  The residual matrix is printed only after the last     ****
                   dimension (standard).                                  ****
             '1'.  The residual matrix is printed out for each            ****
                   dimension.                                             ****
             '2'.  The residual matrix is printed only after the last     ****
                   dimension and is written on IST as CORRM1.             ****
                                                                          ****
     Reproduced matrix.                                                   ****
                                                                          ****
     COLUMNS AND CONTENTS                                                 ****
                                                                          ****
     1-16    'Reproduced' (begin punching in col.1).                      ****
                                                                          ****
     17-20   '0'.  The reproduced matrix is written on the intermediate   ****
                   binary storage tape as the CORRM1 file.                ****
             '1'.  The matrix is not written IST.                         ****
```

Important: recall that depositing either matrix on CORRM1 removes its prior contents. If you want to save them, call '/GIVE' before '/FAST'.

Tape input. The following data are read from the intermediate storage tape (IST) for the control options indicated:

DATA INPUT	FILE	CONTROL OPTION
Identification	IDFILE	All options
Factor coefficients	UFACT1	All options
Correlation coefficients	CORRM1	When cols. 1-16 contain 'residual'

Printed output consists of the identification obtained from IST and the matrix requested.

Tape output. The following information is written as the CORRM1 file on IST for the following control options only. All other control options produce no tape output.

DATA OUTPUT	FILE	CONTROL OPTIONS
Residual correlations	CORRM1	Cols. 1-16 contain 'residual' and cols. 17-20 contain '2'.
Reproduced correlations	CORRM1	Cols. 1-16 contain 'reproduced' and cols. 17-20 contain '0'.

User's description of least squares key cluster analysis: LEAKEY

Introduction

A general solution to key cluster analysis can be obtained in the least squares sense. This solution has not been programmed for BCTRY at the time of this edition of the User's Manual. However, a variant of and an approximation to the general solution may be obtained by using the already programmed components of BCTRY.

Two types of analyses are possible. The first is an orthogonal factoring of the correlation matrix with the successive components being defined by a fixed set of pre-determined clusters. The second is an oblique direct (i.e., unrotated) solution.

Method

The method of LEAKEY is an iterative use of the components of BCTRY. At present the calculations are dependent on predefined clusters. The clusters may be predefined by any one of a number of methods. The least squares theorist will probably want to use an initial principal axes solution for the entire set of variables to select the clusters. This is done by inspecting the rotated solution and the SPAN diagrams of the principal axes analysis. Each dimension is represented by a subset of variables as mutually collinear as possible with each cluster as independent of the others as possible.

Other techniques may be used to select the clusters. For example an empirical key cluster analysis by CC5 provides excellent clusters.

Each of the predefined clusters is used in turn to define the principal axis (singular) associated with the cluster. For the first axis the submatrix of intercorrelations for the first cluster is analyzed for the largest principal axis. The correlations of this principal axis and all of the variables in the entire matrix are calculated. This set of correlations is the vector of correlations of the first cluster-axis with the observed variables. The residual matrix of intercorrelations is obtained by subtracting the inner-products of the cluster-axis correlations from the original correlation matrix. The sub-matrix of intercorrelations of the variables of the second cluster in the residual matrix is analyzed similarly, and so on for all of the clusters, always using the current residual matrix. The result is a set of orthogonal principal cluster-axes.

A direct oblique solution may be obtained by successively evaluating the principal cluster-axes for each cluster, using the original intercorrelations of the variables for each axis.

Use

LEAKEY is a compound and as such requires a deck of executive and component control cards for other components of the BCTRY system. The following outline is intended to be suggestive only and not to give the specific details of card punching. Two routines are involved, one for the orthogonal solution, the other for the oblique solution.

A. Orthogonal LEAKEY. After initiating BCTRY and establishing the correlation matrix and other IST files used by SLEP the control deck is as follows:

```
/SLEP1
The control cards for SLEP1 are punched so as to
suppress all variables excepting those of pre-
defined cluster number 1.
/DVP
0071
/FALS
PFA
The control cards for PFA are punched so as to
iterate on one factor only.
/SLEP2
/FAST
Residual
/SLEP1
The control cards for SLEP1 are punched so as to
suppress all variables excepting those of pre-
defined cluster number 2.
/DVP
0071
/FALS
PFA
The control cards for PFA are punched so as to
iterate on one factor only.
/SLEP2
/FAST
Residual
. . .
. . .
. . .
/SLEP1
The control cards for SLEP1 are punched so as to
suppress all variables excepting those of the last
predefined cluster.
/DVP
0071
/FALS
PFA
The control cards for PFA are punched so as to
iterate on one factor only.
/SLEP2
```

```
/FAST
Residual
/DVP
0071
/FALS
Standard control cards for PFA based on squared
correlation absorption criterion.
```

The final FALS analysis is used to check the adequacy of the principal cluster-axes to account for the entire correlation matrix.

B. Oblique LEAKEY. After initiating BCTRY and establishing the correlation matrix and other IST files used by LEAKEY the control deck is set up in the same cycles as for orthogonal LEAKEY except that the /FAST cards are not included. The successive applications of SLEP and FALS are to the original correlations. The first cycle is given by:

```
/SLEP1
The control cards for SLEP1 are punched so as to
suppress all variables excepting those of pre-
defined cluster number 1.
/DVP
0071
/FALS
PFA
The control cards for FALS are punched so as to
iterate on one factor only.
/SLEP2
/SLEP1
The control cards for SLEP1 are punched so as to
suppress all variables excepting those of pre-
defined cluster number 2.
. . .
. . .
. . .
```

The final FALS analysis on the entire matrix is omitted. The final matrix is the original matrix.

User's description of oblique cluster structure analysis (model I): CSA

(Card changes of new CSA2 are inserted at the end of this description.)

Introduction

The main objective of CSA is to provide a complete statistical description of the correlational properties of oblique cluster domains ('factors') defined as 'structured' domains under model I. The subset of variables that defines each cluster is usually selected by a prior key cluster factoring component (CC or NC analysis), but may also be derived rationally. This objective is known in orthodox factor analysis as a direct 'rotated oblique factor solution.' The program also includes the component CLUR which gives an oblique unifactor 'structure' and reorders the raw correlation matrix by clusters. See Tryon (1964a,1964b) for an exhaustive account of CSA.

Method

In using CSA, one defines a score on a domain (a 'factor') under model I as a 'structured' composite on a component set of variables, each set itself composed of many variables strictly collinear with one of the observed variables. This definition requires the use of communalities of the observed variables estimated from DVP or from a prior factoring.

If it is desired to preset the clusters used by CSA, as in the analysis of rational clusters without prior factoring, the rational groups can be input through GIST on file REFLX1.

A CSA analysis on a published factor analysis can be performed by inputting the factor coefficients in file UFACT1 via GIST; the correlation matrix used by CSA can then be generated from them by CSA. Cluster indicators would be input by GIST.

Use (if you are using the July 1965 version of BCTRY, see CSA2)

```
     Card input.
     (1) Executive control card.  Punch '/CSA+++' in columns 1-7.        ****
                                                                         ****
     (2) Component control cards.  Two control cards are required.       ****
The first provides the user with control of input and output.  The sec   ****
ond provides a criterion for the clustered correlation analyses.         ****
                                                                         ****
     Card A.  Punch the following control constants on an integer        ****
card:                                                                    ****
                                                                         ****
     COLUMNS AND CONTENTS OF CARD A                                      ****
                                                                         ****
     1-4  Determination of the correlation matrix to be analyzed.        ****
             '0'.  Analyze matrix of observed correlations among         ****
                   variables (standard).                                 ****
             '1'.  Analyze matrix of correlations reproduced from        ****
                   factor coefficients.                                  ****
```

5-8 Calculation and printing of clustered correlation matrix. ****
'0'. The matrix is calculated and printed. ****
'1'. The matrix is not calculated (especially in ****
O-analysis with large N). ****
'2'. The matrix is not printed. ****
****

9-12 Calculation and printing of augmented rotated factor ****
coefficients. ****
'0'. The matrix is not calculated or printed (standard). ****
'1'. The matrix is calculated and printed. ****
****

13-16 Calculation and printing of augmented correlation matrix. ****
'0'. The matrix is not calculated or printed (standard). ****
'1'. The matrix is calculated and printed. ****
****

Card B. This card contains , the lower limit of communality ****
for the clustered correlation analysis. Variables whose communalities ****
are below this limit are ignored in the clustered correlation analysis ****
HLOW should be punched in columns 1-8 using a decimal card format. Th ****
Standard value of HLOW is 0.20. ****

Tape input. The following information is input from the intermediate storage tape (IST):

FILE NAME		CONTROL OPTIONS FOR WHICH FILE IS REQUIRED INPUT
IDFILE	Identification	All options
REFLX1	Reflection indicators	All options
CORRM1	Correlation matrix	When Card A columns 1-4 contain '0'.
DIAGV1	Communalities	When Card A columns 1-4 contain '0'.
UFACT1	Factor coefficients	When Card A columns 1-4 contain '1'.

Printed output. The printed output is self-explanatory.

Tape output. The following information is placed on IST for all control options:

DESCRIPTION OF CONTENTS	FILE NAME
Rotated oblique factor matrix	RFACT1
Correlations among rotated factors	BASIS1

Note: when unities are set as diagonal values (by DVP71) before calling CSA, all correlations are raw R's. CSA is thus useful for a standard item analysis in test construction.

User's description of CSA2

Summary

There are three important additions to CSA: 1) determining the degree of 'generality' of each oblique cluster ignoring the other clusters, 2) providing specific data on how to increase the reliability of each cluster by including additional definers of it, 3) computing the mean intercorrelation of the definers of each cluster with other variables unifactorially allocated to it. The meaning of these innovations is given in Tryon (1964a, Sect. 11, 12, 13). Other changes suppress the printing of certain statistics (see Use).

Use of CSA2

The innovations do not require the use of any control cards options, so the Use section of CSA applies directly to CSA2, except for the following two changes that repress the statistics indicated unless otherwise opted by the user:

5-8 CLUR, calculating and printing the clustered correlation matrix.

'0'. Form and print the matrix (this option is no longer standard).

'2' or blank. Do not form or print the matrix (standard).

21-24 Correlations of raw cluster scores with cluster domains.

'0' or blank. Do not calculate the correlations.

'1'. Calculate and print the correlations.

User's description of oblique non-communality cluster structure analysis (model II): NCSA

(Card changes of new NCSA2 are inserted at end of this description.)

Introduction

The main objective of NCSA is identical with that of CSA except that the oblique cluster domains ('factors') of NCSA are defined under model II by 'representative' samples of variables. Hence, communalities are not used by NCSA.

A special use of NCSA is in test-construction where the analyst seeks knowledge of the correlational structure of cluster scores on rational 'subscales' drawn from a large pool of test-items. The statistics provide a more powerful comprehensive form of item and subscale analysis than is usually available. When rational clusters have a very low average correlation, the fact is signaled by absurd findings, and if negative by a program stop. As presently programmed, NCSA should not be used when different clusters have overlapping variables for the results are in error. NCSA can take clusters formed by as many as 60 variables.

Use (if you are using the July 1965 version of BCTRY, see NCSA2)

```
     Card input.                                                   ****
                                                                   ****
     (1)  Executive control card.  Punch '/NCSA++' in columns 1-7. ****
                                                                   ****
     (2)  Component control cards.  These cards specify the methods ****
used in the analyses and may also specify the key cluster variables ****
used to define each dimension.                                     ****
                                                                   ****
     Card A.  This card determines whether the observed or reproduced ****
correlation matrix is used, whether the clustered correlation matrix is ****
formed, and the method of inputting the defining variables of each ****
cluster.  The contents are as follows:                             ****
                                                                   ****
     COLUMNS AND CONTENTS OF CARD A                                ****
                                                                   ****
     1-4     Determine the correlation matrix to be analyzed.      ****
             '0'.  Analyze the matrix of observed correlations among ****
                   variables (standard).                           ****
             '1'.  Analyze the matrix of correlations reproduced from ****
                   the factor coefficients.                        ****
                                                                   ****
     5-8     The calculation and printing of the clustered         ****
             correlation matrix.                                   ****
             '0'.  Form and print the matrix.                      ****
             '1'.  Do not form or print the matrix (standard).     ****
```

9-12 Determine the method of specifying the number of clusters ****
and the defining variables of each cluster. ****
'0'. Input from intermediate storage tape (standard). ****
'1'. Input by the analyst (see control cards B). ****
****

Cards B. These cards are necessary only when columns 9-12 of Car ****
A contain a '1'. They specify the defining variables of each ****
cluster to be input. In the following, let K be the number of cluster ****
let NI be the number of variables defining cluster I, and let SI be ****
the reflection indicators of the variables which are to define cluster ****
I, that is, SI = I(I,1), I(I,2),..., I(I,NI) where I(I,J) is the index ****
of variable J in cluster I. I(I,J) should be negative if variable J i ****
to be reflected in cluster I, otherwise I(I,J) should be positive. ****
****

Control cards B are all integer cards (see Glossary) punched ****
as follows: the first card contains K, the number of clusters punched ****
in columns 1-4. The second card contains the number of variables ****
defining each of the K clusters, SI,S2,...,SK, punched as a vector ****
on integer cards (see Glossary). ****
****

The third and succeeding cards contain the sets of indices for ****
the K clusters, S1, S2,...,SK; each of these successive sets is punched ****
as a vector on integer cards. ****

Tape input. The following information is input from the intermediate storage tape (IST):

NAME	CONTENTS	CONTROL OPTIONS FOR WHICH FILE IS REQUIRED INPUT
IDFILE	Identification	All options
CORRM1	Correlation matrix	When cols. 1-4 contain '0'.
UFACT1	Unrotated factor coefficients	When cols. 1-4 contain '1'.
REFLX1	Reflection indicators	When cols. 9-12 contain '0'.

Printed output includes the data identification, clusters used, rotated oblique factor coefficients, correlations among oblique dimensions, domain validities, and correlations among cluster scores.

Tape output. The following information is written on IST for all control options: the rotated oblique factor coefficients as RFACT1 and the correlations among the rotated oblique dimensions as BASIS1.

NCSA is subject to the BCTRY system restrictions with one exception. When the cluster indicators are read from cards by NCSA, the maximum number of variables in a cluster is 60. However, when the cluster indicators are read from IST, the maximum number of variables in a cluster is 20.

It should also be noted that NCSA is not designed for the case where the same variable is used to define more than one cluster. Whenever such a definer occurs NCSA simply deletes all calculations but does not cause an error termination.

User's description of NCSA2

Summary.

The three additions described above for CSA2 are also incorporated in NCSA2. The meaning of them is given in Tryon (1964a, sections 9, 10, 11). The clustered matrix is now suppressed.

Use of NCSA2.

5-8 Calculating and printing the clustered correlation matrix.

'0'. Form and print the matrix (this option is no longer standard).

'2' or blank. Do not form or print the matrix (standard).

User's description of spherical analysis: SPAN

(Card changes in SPAN2 are inserted at end of this description.)

Introduction

The central object of V-analysis, recall, is to describe individual differences in NV variables by a reduced set of K general composities of them that can be meaningfully interpreted. The first dimension derived by key cluster factoring (CC and NC) is usually a meaningful composite of collinear definers, but later dimensions, being perforce orthogonal, rarely 'pass through', or are directly measures of, collinear subsets of variables. For this reason, by using the component CSA, we center on the oblique groups of variables by which each of the dimensions has been defined. In short, orthogonal dimensions are not usually meaningful, oblique usually are. Unfortunately, however, even if the NV variables should in fact fall nicely into a good cluster structure, there are arbitrary conditions built into the factoring procedures that may not result in the obvious clusters being exactly selected as the critical defining subsets that go into CSA as the oblique clusters whose structural relations are analysed there. In some problems, indeed, no clear cluster structure may in fact exist, so that the defining subsets may be quite arbitrarily selected by the factoring component. What is needed is to see the structural pattern of collinear relations among the variables as a whole - the total configuration. For when the configuration is visually examined, arbitary happenstance features in cluster selection by the factoring method become immediately apparent. The analyst will examine the configuration, revise the cluster selection if necessary, and rerun the analysis, certainly the CSA part of it.

The problem is complicated by the fact that the total configuration can be geometrically represented exactly only if we factored on NV dimensions, for in the NV dimensions it is invariant with respect to the method of factoring - subset or total set. But a distinctive feature of factoring is that as it proceeds, additional dimensions describe a configuration that departs less and less from that given by all NV dimensions. The practical solution therefore is to select the salient set of K dimensions that describes the salient configuration - one that departs only trivially from the fill exact one. To achieve this objective of configuration representation is the purpose of 'spherical analysis', programmed by the component SPAN.
Taking the K dimensions derived by key cluster factoring, SPAN selects 'subspaces' in these dimensions in this fashion: the first subspace describes the configuration with maximal saliency, another is then added with next highest saliency, and so on. SPAN does this for variables that are 'sufficiently' described in single dimensions, then in '2-dimensional subspaces' (in planes), and in '3-dimensional subspaces' (in spheres). For each of the spheres, a printout of its portion of the total configuration is provided. In most problems, even though the dimensionality is so large that the configuration would be theoretically represented only in a 'hypersphere' (a sphere of more than three dimensions), the analyst can, nevertheless, by cross-referencing the visually-presented fractions of the total configuration given in the 'minimally sufficient 3-dimensional

subspaces' (i.e., in the printout spheres) 'see' the total configuration, for usually 'marker' variables or clusters carry over from one sphere to another to aid in the cross-referencing perceptual act.

Method (for the simple geometry of SPAN, see Tryon, 1964a)

Surface model. - The geometric model on which the configuration is presented is the generalized sphere. This model is appropriate because any variable V can be plotted as a point on such a sphere because the sum of its squared augmented factor coefficients (taken as coordinates of a Cartesian frame) equals 1.00. As a surface point variable V refers to the variable-domain CV.

From the analytic geometry of the situation, the central angle between any two variables, VI and VJ, represents the correlation between their variable-domains. That is, their correlation is the cosine of their central angle, or the 'inner products' of their augmented factor coefficient.

Stated quite simply, this means that the spatial configuration or layout of all NV variables on the sphere depicts their correlations or their central angles as indexed by their cosines, for the arc between any two variables on the surface is a linear function of the central angle between them. Thus, par excellence the surface configuration is a map that depicts visually all the intercorrelations between the variables; it is a visual representation of the complete table of augmented correlations among all NV variables (the 'common factor correlations'), hence it describes the relations between the variables 'in common factor space', i.e., with respect to what is common or general in them.

Inner model. - The raw correlations among the variables are not perceptually represented in this model. Each observed variable V is represented as a point in the inner model of the sphere at a distance H (the square root of its communality) on a vector from the origin to the surface point C, its variable-domain; indeed, the inner point is located there by plotting it from its unaugmented factor coefficients (derived by factoring) treated as coordinates on the K axes. But the configuration of the NV observed variables in the inner modes does not depict their common factor R's. It is for this reason that the past practice of orthodox factor analysts of plotting variables by their factor coefficients and 'rotating axes' in this space is a misleading enterprise. Only the surface configuration represents relationships among the variables, not the inner configuration.

Positive manifold. - On a printout sphere of SPAN the positions of the termini of the three factored dimensions that define it are clearly indicated as points X, Y, and Z, and around each of these axes, or as nearby as possible considering that the dimensions must be orthogonal, are crowded the defining variables of it. Non-defining variables also described in this 'subspace' are also shown. Lines drawn between the X, Y, and Z axes (dots given on the printout show where the lines go) define a spherical triangle each side of which is a plane bounded by two of the dimension-axes. If all variables depicted on the sphere lie within the triangle they are said to reveal a 'positive manifold', i.e., have positive factor coefficients on the three dimensions. In order best to see the configuration in one place on the sphere, SPAN reflects any variables which, if by so doing, the point that represents it lies nearer the center of the spherical triangle. Variables that are not reflected are plotted by the symbol '0', those reflected by 'R'. The configuration is thus visually 'centered'.

'Rigid' rotation. - After such reflection (the sphere is rigidly rotated so that the centroid of all the points on the printout page is directly in the line of the eye. From this vantage point the configuration can best be perceptualized. Rigid rotation means that the sphere is rotated, not the factored dimensions X, Y, and Z (the program GYRO does that).

Physical model. - Since analysts with poor space perception may have difficulty in 'seeing' the configuration on the printouts, it is often desirable to transfer the configuration to a physical globe, such as a plastic balloon. The procedure is quite simple. Since the coordinates of each variable on the three rigidly rotated dimensions are listed in the printout, each variable marked 'V' or 'R' can be plotted on the globe from these coordinates. The termini of the three rigid axes are first designated 90 degrees apart on the globe. Then a semi-circular collar is prepared on which is written the cosines of angles from 0 degrees to 90 degrees. Since the printed coordinates are in fact the cosines of the angles between a variable and a given axis, with this circular ruler each point can be located by triangulation.

Selection of the minimally-sufficient subspaces in which to describe the configuration.

Just as in factoring we select the salient number of dimensions K to describe the configuration, so also for the given K we depict the configuration progressively with maximal saliency or 'sufficiency'. The problem is quite crucial, especially when the dimensionality exceeds three, for no configuration can be 'seen' on a hypersphere. The solution is to fractionate the total configuration into 'subspaces' in which only those variables are represented whose communalities are accounted for in a given subspace at a high optional level of sufficiency, S (designated as CL in SPAN). If we let S be less than 1.00 then there usually exists a minimal set of subspaces in which all or substantial parts of the total configuration can be visually represented. It is, for example, not uncommon to be able to depict the total configuration on a hypersphere of, say, six dimensions in one sphere at a sufficiency level of, say, .8 or .9.

The computer does this logic job (it is basically not computational) with great speed. At a sufficiency level of, say, .8, it first allocates all variables that meet this level on single dimensions i.e., 1-dimensional subspaces. That dimension with the maximal number of variables allocated to it is printed as 'set 1' then the dimension that includes the maximal number of variables not allocated to the first set becomes the second printed set, and so on. The individual dimensions are thus ordered by saliency with which they account for the NV variables at the specified sufficiency level. Usually, only several such sets are sufficient, though many variables may be missing from them. Then 2-dimensional sets are analogously tested, and ordered by saliency. More variables, perhaps all NV of them, may be described by a small number of 2-dimensional sets, or planes. Finally, 3-dimensional sets, or spheres, are similarly tested. Usually, all NV variables, or nearly all, are included in one or another of the spheres, ordered by saliency, usually small in number compared to the total number of sets possible. These are the sets that are printed by SPAN.

Methods of factoring in relation to configuration analysis

Key cluster (collinear subset) factoring is the form that optimally permits the description of the configuration, because it is especially tailored to project axes into clusters in the configuration, or to set them in planes or 'hyperplanes'. Total set-factoring by centroid, principal axes, or canonical methods is quite inferior in this regard because they project axes into centers of the total or total residual configuration, where in fact few variables may be located. Orthogonal rotation of them (see GYRO) may improve configurational analysis, especially if performed by quartimax rotation, which tends maximally to pass axes near positions at which key cluster factoring directly sets them. Varimax is usually quite poor for configurational analysis by SPAN, especially when the configuration is rather oblique, for varimax axes in this case are often far removed from the configuration itself.

Selection of the final set of oblique clusters from the configuration

With the salient configuration before him, the analyst can finally select the final set of oblique cluster of variables by which to score individuals (by FACS). SPAN usually reveals that the dimensions from key cluster factoring will have passed as near to collinear subsets as is possible, subject to the condition of their being orthogonal. But these defining subset need not always be accepted. The analyst will normally wish to add to oblique clusters some variables that have not been routinely accepted as definers. The 'unifactor' allocation of variables to clusters provided by CLUR in the CSA program will frequently be useful here. He may, of course, wish to reject some machine-selected definers, especially those at the 'edge' of a cluster).

'Simple structure'. - Orthodox factor analysts interested in the concept of 'simple structure' (Thurstone, 1945, ch. 14) will find SPAN tailored to their purposes. SPAN reveals an orthogonal or oblique simple structure if all the variables lie sharply (at a high sufficiency level) in planes or hyperplanes with a positive manifold - Such a structure will be quite apparent im the SPAN spheres and in the statistics on minimal subspaces. The cluster analyst on the other hand, will usually not expect to discover such a structure, nor be unduly impressed by it if it indeed appears. He is interested not in dimensions that bound the configuration but rather in the observable clusters of variables within it. He would tend not to assign to a dimension the status of an 'underlying causative factor'. On domain sampling principles, he sees the configuration simply as a layout of a sample set of variables drawn from domains of behavior circumscribed by the sphere or hypersphere. The shape of the configuration may, indeed, suggest to him that other variables might in a future experiment be measured to fill out the configuration in some meaningful way; if so, all to the good. But the main use of the configuration is to identify clusters of variables by which to measure and compare individuals in a meaningful and efficient (i.e., reductionist) way. The configuration describes correlation (i.e., generality of individual differences) and not causation. To no place in the configuration can one point and say: 'here lies an "underlying factor", a "functional unity", a "vector of mind"'. Such a belief is ex cathedra to multidimensional analysis itself.

Use (if you are using the July 1965 version of BCTRY, see SPAN2)

Card input. ****
(1) Executive control card. Punch '/SPAN++' in columns 1-7. ****
****

(2) Component control card. The program requires only one ****
control card specifying the source of the factor coefficients to be ****
used, the maximum subspace dimensionality, and the minimum saliency ****
value of variables. Blank fields yield standard options. The content ****
are as follows: ****
****

COLUMNS AND CONTENTS OF CONTROL CARD ****
****

1-4 Specifies the source of factor coefficients used in the ****
analysis. The contents are punched in the integer card ****
format (see Glossary). ****
'1'. Use unrotated factor coefficients (UFACT1) from IST ****
(standard). ****
'2'. Use rotated factor coefficients (RFACT1) from IST. ****
****

5-8 NF, the maximal dimensional-subspace to be considered by ****
the program. Thus, the program will select subspaces ****
consisting of 1,2, ..., NF dimensions. The possible ****
contents are whole numbers between one and K, the total ****
number of dimensions, punched in the integer card format. ****
The standard value is '3', i.e., 3-dimensional subspaces ****
(spheres). ****
****

9-12 CL, the minimal communality criterion value. A variable ****
is 'projected' in a subspace only if the proportion of ****
its communality exhausted by the subspace is greater than ****
or equal to CL. The contents are decimal numbers between ****

zero and one, punched in the decimal card format (see ****
Glossary). The standard value of CL is .80. ****
****
13-16 NSTART, the minimal subspace desired (standard is 3). ****
****
17-80 Ignored by this program. ****

Tape input. The following files must be input via the intermediate storage tape (IST):

IDFILE Identification
UFACT1 Unrotated factor coefficients (required if columns 1-4 of component control card contains a '1' or blank).
RFACT1 Rotated factor coefficients (required if columns 1-4 contain a '2').

Printed output. In addition to the title page and problem identification, the program prints the augmented factor coefficients (coordinates), partial communality, and total communality for all variables in every subspace. Variables projected in a given subspace at sufficiency level NF are indicated by arrow heads 'V' or 'R' appearing over their coordinates in the space. Finally the percent of total communality in each subspace is printed. For 3-dimensional subspaces the coordinates on the rigidly rotated axes are printed.

In the case of the minimal sets of 3-dimensional subspaces the program plots the projected variables on the unit sphere, using the coordinates printed in the tables. The procedure is as follows: first, the coordinates' axes are reflected so that most of the variables will lie around the center of the printed sphere; second, the variables are plotted on the unit sphere, using '0' for an unreflected variable and 'R' for reflected variables. Decimal points are used as dots to outline the first octant (the spherical triangle) of the original reference axes derived by factoring. It should be noted that these transformations are merely changes in arbitrary axes, and the distances among the variable points are not affected.

SPAN has no IST or DST output.

All BCTRY restrictions apply.

Caution: after writing in the VNAMS of variables on a given sphere, check to see if any are on the under-hemisphere. These would be variables in the table of three coordinates marked 'V' whose Z coordinate is negative, or those marked 'R' whose Z is positive.

Revision to
user's description of SPAN

Summary

This new revision of SPAN introduces the 'spotter', a feature of each sphere that labels the variables on it by their VNAMS (or ONAMS) and also by their input ordinal positions. The names of the variables are also attached to their coordinates. You can now communicate directly with the component and ask for any specified subspaces you wish. Printing is drastically reduced by giving routinely only spherical subspaces, though you can opt for any additional types of subspaces you wish. Markers now designate particular variables in each sphere that have low communality. In BIGNV procedures where you form a composite subspace from different samples of variables in V-analysis (or of subjects in O-analysis), an invaluable aid is now the capability of calling for rigid rotation of a given spherical subspace in the different samples on their common dimension definers.

Use of SPAN2

Follow the Use section of SPAN in the Manual, except as noted below:

```
     (1)  Executive control card.  Punch '/SPAN2+' in cols. 1-7.         ****
                                                                         ****
     (2)  Component control cards.  Only one control card, Card A, is    ****
still required unless you desire to specify which particular subspaces   ****
you wish to have printed, in which case these are designated on a        ****
second control card, Card B.  The main control card A has been           ****
entirely restructured as follows:                                        ****
                                                                         ****
     COLUMNS AND CONTENTS OF CARD A                                      ****
                                                                         ****
             COLUMNS          CONTENTS                                   ****
                                                                         ****
             1-4       Mode of selecting subspaces (integer format).     ****
                                                                         ****
                       '1' or blank.  The SPAN2 program selects all the  ****
                       subspaces (standard).                             ****
                                                                         ****
                       '2'. User selects all the subspaces by control    ****
                       cards B.                                          ****
                                                                         ****
                       '3'. SPAN2 selects as in '1' but user also        ****
                       designates some subspaces by control cards B      ****
                       that SPAN2 should not miss.                       ****
                                                                         ****
                       Note:  user designation of subspaces in '2' and   ****
                       '3' overrides designations by the controls belo   ****
                       in cols. 9-12, 13-16, 17-20.                      ****
```

```
5-8       Source of orthogonal factor coefficient         ****
          (integer card format).                          ****
                                                          ****
          '1' or blank.  From factoring, i.e., UFACT1 fil ****
          (standard).                                     ****
                                                          ****
          '2'.  From orthogonal (never oblique) rotation  ****
          GYRO, i.e., RFACT1 file from varimax or         ****
          quartimax.                                      ****
                                                          ****
          Note:  if you revise dimension-definers and     ****
          wish a new SPAN, rerun a preset CC or NC        ****
          factoring on the new definers so that you go    ****
          on into SPAN2 with new factor coefficients on   ****
          orthogonal dimensions (i.e., UFACT1).           ****
                                                          ****
9-12      NSTART, the lowest dimensional subspace         ****
          desired (integer card format).  Punch an integer ****
          number from 1 to 15 denoting the lowest         ****
          dimensional subspace desired.  SPAN2 analyzes   ****
          all subspaces starting with this number up to   ****
          NFIN, specified in cols. 13-16, just below.  If ****
          NSTART is left blank the standard is set at 3.  ****
                                                          ****
13-16     NFIN, the highest dimensional subspace desired  ****
          (integer card format).  Punch an integer number ****
          from 1 to 15 denoting the highest dimensioned   ****
          subspace desired.  SPAN2 finishes analyzing     ****
          subspaces defined by this number of dimensions. ****
          If left blank the standard is set at 3.         ****
                                                          ****
17-20     MINEW, minimal number of 'new' variables that a ****
          subspace must include if it is to be printed    ****
          (integer card format).  To illustrate:  the     ****
          integer '3' would mean that a new triplet is    ****
          sufficient to justify printing a subspace, i.e. ****
          subspaces defined by less than 3 new variables  ****
          are deleted.  If left blank, the standard is    ****
          set at 3.                                       ****
                                                          ****
21-28     CLOPR, the lowest percentage of communality of  ****
          a variable that will admit it to a subspace     ****
          (decimal card format).  A variable is 'projected ****
          into a subspace only if the proportion of its   ****
          communality exhausted by the subspace is greater ****
          than or equal to CLOPR.  Punch the decimal value ****
          of CLOPR.  If left blank, the standard is set a ****
          '+++.8000'.                                     ****
                                                          ****
29-35     MINCOM, the lower bound of a communality below  ****
          which a variable is marked for potential        ****
          exclusion from a printed subspace (decimal      ****
          card format).  A variable listed in a spotter   ****
          is given an asterisk if the absolute value of   ****
```

```
                  its communality is below the punched decimal      ****
                  value of that lower bound.  If left blank, the    ****
                  standard is set at '+++.1000'.                    ****
                                                                    ****
        36-40     NOTATE, a physical rotation of each sphere        ****
                  (see Method) not determined by all variables on   ****
                  the sphere but only by dimension-definers.        ****
                                                                    ****
                  '0' or blank.  Rotation determined by all         ****
                  variables on the spheres (i.e., those marked      ****
                  'V' or 'R' in the table of coordinates).          ****
                                                                    ****
                  '1'.  Rotation determined by dimension-definers   ****
                  only (i.e., those in REFLX1 file).                ****
                                                                    ****
                                                                    ****
     Card B.  Designation of desired subspaces (necessary if either ****
mode '2' or '3' is punched in cols. 1-4) (integer card format of    ****
4-column fields).  For each subspace desired, keypunch on a separate ****
card the dimensions that define it, e.g.,                           ****
                                                                    ****
     '0001' specifies the single axis defined by dimension 1.       ****
     '00010003' specifies the plane defined by dimensions 1 and 3.  ****
     '000100030004' specifies the sphere defined by dimensions 1, 3, 4.  ****
     '0001000300040007' specifies the hypersphere defined by dimensions  ****
        1, 3, 4, 7.                                                 ****
                                                                    ****
     The maximal number of cards cannot exceed 90.                  ****
                                                                    ****
     Important:  when Cards B are used, always follow them by a blank ****
card.  If no Cards B are used, no blank card should be used.        ****
                                                                    ****
                                                                    ****
```

The rest of the Use section is the same as in the User's Manual, except under 'printed output' add 'a spotter is now printed for each sphere.'

User's description of programs to rotate axes: GYRO

Introduction

The object of GYRO is to provide in one general component alternative methods of rotating the dimensions derived by factoring. Some form of rotation is necessary when the correlation matrix is factored by total-set factoring, namely, by centroid factoring (CC5 CENT), or by FALS factoring, principal axes (PRIN), canonical (CANON), or augmented (AUG). Such rotations are known as 'derived' solutions (Harman, 1960, part III). The quartimax and varimax programs of GYRO are modified versions of the published programs of Cooley and Lohnes (1962).

Use

```
     Card input.                                                       ****
     (1) Executive control card.  Punch '/GYRO++' in columns 1-7.      ****
                                                                       ****
     (2) Component control cards.  The program requires three control  ****
cards specifying the method of rotation, and the criteria for          ****
terminating iteration.                                                 ****
                                                                       ****
     Card A.  This card specifies the method of rotation to be used.   ****
                                                                       ****
     COLUMNS             CONTENTS AND MEANING                          ****
                                                                       ****
     1-6     'QRTMAX'   Use the quartimax method.                      ****
             'VARMAX'   Use the varimax method.                        ****
                                                                       ****
     Card B.  This card contains the maximum number of iterations to b ****
performed, punched in columns 1-4 in an integer card format.  The      ****
standard value is 10.                                                  ****
                                                                       ****
     Card C.  This card contains the convergence criterion for         ****
iteration termination, punched in columns                              ****
                                                                       ****
     1-8, using a decimal card format.  The standard values are .001   ****
          (for quartimax) and .0000001 (for varimax).                  ****
```

The program requires both the identification (IDFILE) and the unrotated factor coefficients (UFACT1) from the intermediate storage tape (IST).

The printed output includes a description of methods used, a summary of iteration, the converged rotated factor coefficients, the augmented rotated factor coefficients, and partial communalities.

Tape output. The program writes the converged rotated factor coefficients on the intermediate storage tape in the RFACT1 file on IST, and an identity matrix in BASIS1.

USERS' DESCRIPTION OF BIGNV (ENCOMPASSING SAMPLER AND MERGER)

INTRODUCTION

BIGNV PROCEDURES ENABLE YOU TO PERFORM A V-ANALYSIS ON PROBLEMS WITH UP TO 2000 VARIABLES (AND O-ANALYSIS ON SUBJECTS UP TO 10,000 CASES). THESE OBJECTIVES ARE ACHIEVED IN SPITE OF THE FACT THAT THE COMPONENTS OF BC TRY HAVE A LIMIT IN ONE COMPUTER RUN OF HANDLING NOT MORE THAN NV=90 VARIABLES. THE RESTRICTION IS SURMOUNTED BY THE SAMPLING AND MERGING FEATURES OF BIGNV.

METHOD

BROADLY SPEAKING, HERE ARE THE STEPS.. IN THE FIRST STAGE OF BIGNV ANALYSIS, THE FULL SUPPLY IS DIVIDED INTO STRATIFIED RANDOM SAMPLES OF 90 VARIABLES, USING SPECIAL COMPUTER PROCEDURES CALLED SAMPLER. IN THE SECOND STAGE, THE CLUSTER STRUCTURE OF EACH SAMPLE IS THEN DETERMINED BY REGULAR BC TRY COMPONENTS. IN THE THIRD STAGE, ANOTHER SPECIAL SET OF COMPUTER PROCEDURES CALLED MERGER COMBINES THE STRUCTURES OF THE SAMPLES TO FORM THE OVERALL STRUCTURE OF THE FULL SUPPLY. IN SHORT, SAMPLER BREAKS DOWN THE SUPPLY AND MERGER BUILDS IT UP AGAIN, BUT IN THE PROCESS THE CLUSTER STRUCTURE OF THE SUPPLY HAS BECOME REVEALED. THE FOURTH AND FIFTH STAGES ARE CONCERNED WITH THE DESCRIPTION OF THE STRUCTURE OF THE FULL SUPPLY AND THE PREDICTABILITY OF IT.

THE FIRST STAGE SAMPLER COMPOUND CONSISTS OF SEVERAL COMPONENTS, THE PROGRAMMING OF WHICH SHOULD BE COMPLETED BY FALL, 1965. IN THE INTERIM, A SURROGATE IS AVAILABLE IN BERKELEY CALLED PRIEST AND TRYST. ELSEWHERE USERS MUST DRAW RANDOM SAMPLES BY HAND (SEE SAMPLER). THE SECOND STAGE REGULAR BC TRY COMPONENTS THAT PERFORM THE CLUSTER ANALYSES OF THE SAMPLES ARE, OF COURSE, PRESENTLY AVAILABLE. THE THIRD STAGE MERGER COMPOUND CONSISTS OF THE FOLLOWING THREE SEPARATE COMPONENTS NOW FULLY PROGRAMMED AND AVAILABLE.. SETCOPY IS A SUBEXECUTIVE THAT ACTIVATES A SPECIAL TAPE THAT IS TO RECEIVE THE RESULTS ON THE SAMPLES. COPY WRITES THE RESULTS FOUND ON EACH SAMPLE AS THEY ARE COMPUTED. PRANK MERGES THE RESULTS OF ALL SAMPLES STORED ON THE TAPE INTO ONE INTEGRATED DESCRIPTION OF STRUCTURE OF THE FULL SUPPLY.

NOTE.. FOR A WORKED-OUT EXAMPLE OF BIGNV, SEE TRYON, STEIN AND CHU (1965).

USE OF BIGNV

A DETAILED OUTLINE OF THE STAGES OF BIGNV PROCEDURES IS PRESENTED BELOW. UNDER THE STEPS WITHIN THESE STAGES, THE REQUIRED TANDEM SEQUENCES OF BC TRY COMPONENTS ARE GIVEN. THE SYMBOL SAMPLER MEANS THE PROCEDURES OF DRAWING SAMPLES, WHETHER BY THE COMPOUND NOW BEING

PROGRAMMED, OR BY ITS BERKELEY SURROGATE (PRIEST AND TRYST), OR BY USER INPUT.

STAGE 1. SAMPLER PROCEDURES.--BREAKING DOWN THE SUPPLY INTO RANDOM SUBSAMPLES, AND FROM THEM DETERMINING THE PIVOTAL DIMENSIONS OF THE FULL SUPPLY OF VARIABLES.

A. DIVIDING THE FULL SUPPLY OF VARIABLES INTO STRATIFIED RANDOM SAMPLES.

USE SAMPLER(DESIGNING THE SAMPLES)-SETCOPY FOR COMMUNALITIES (DIAGVI).

B. DETERMINING THE COMMUNALITIES OF ALL THE VARIABLES IN THE SAMPLES.

ON EACH SAMPLE.. USE SAMPLER(DRAWING THE SAMPLE) -DAP2-COR2-DVP41-COPY(FOR DIAGVI).

ON THE MERGER TAPE OF SAMPLES.. USE PRANK FOR RANKING COMMUNALITIES (DIAGVI).

C. DETERMINING THE PIVOTAL DIMENSIONS AND THEIR DEFINERS.

1) PERFORMING A CLUSTER ANALYSIS OF A SAMPLE COMPOSED OF THE MOST GENERAL VARIABLES. USE SAMPLER(SELECTING VARIABLES WITH HIGHEST DIAGVI VALUES)-DAP2-COR2-DVP41-CC-CSA2-SPAN2.

2) HIERARCHICAL CONDENSATION OF CLUSTERS, IF NECESSARY, TO FORM THE FINAL DIMENSIONS AND THEIR DEFINERS, I.E., COMBINING DEFINERS OF CORRELATED CLUSTERS AND ELIMINATING TRIVIAL CLUSTERS. EXECUTE BY HAND.

STAGE 2. PRE-SET CLUSTER ANALYSES ON SAMPLES.--CLUSTER ANALYSING SUCCESSIVE SAMPLES OF 90 VARIABLES.

A. DESIGNING THE SUCCESSIVE SAMPLES IN DESCENDING ORDER OF GENERALITY (COMMUNALITY).

USE SAMPLER(ALLOCATING VARIABLES TO SAMPLES BY MAGNITUDE OF DIAGVI VALUES)-SETCOPY(FOR DIAGVI, RFACTI).

B. RUNNING CLUSTER ANALYSES OF THE SUCCESSIVE SAMPLES. FOR EACH SAMPLE..

USE SAMPLER(SELECTING THE SAMPLE FROM THE ABOVE DESIGN)-DAP2-COR2-DVP41-PRESET CC-CSA2-SPAN(ON DEFINERS)-COPY(FOR RFACTI, DIAGVI).

C. TESTING THE SUFFICIENCY OF PIVOTAL DIMENSIONS ON THE SUCCESSIVE SAMPLES.

MAKE A STATISTICAL STUDY, PRESENTLY BY HAND, OF THEIR RESIDUAL MATRICES THAT ARE PRINTED IN THE CC COMPONENT.

STAGE 3. MERGER PROCEDURE.--FORMING THE SALIENT, COMPOSITE STRUCTURE OF THE SUPPLY FROM THE SAMPLES.

A. REJECTING VARIABLES FROM THE COMPOSITE THAT HAVE TRIVIAL COMMUNALITIES.

EXCLUDE VARIABLES WITH TRIVIAL DIAGV1 VALUES (SAY, BELOW .1).

B. FORMING THE COMPOSITE SPAN OF THE NON-TRIVIAL VARIABLES.

FORM A COMPOSITE SPAN FROM THE SEPARATE SPANS OF THE SAMPLES IN STAGE 2B.

C. FORMING A COMPOSITE CSA OF THE NON-TRIVIAL VARIABLES.

FROM COPYS OF STAGE 2B, CALL PRANK TO MERGE THE OBLIQUE FACTOR COEFFICIENTS ON EACH DIMENSION (RFACTI) WITH TAG-ALONG DIAGVI.

STAGE 4. DESCRIPTION.--DESCRIBING THE CLUSTER STRUCTURE OF THE FULL SUPPLY.

A. DECIDING ON THE FINAL MOST NEARLY INDEPENDENT PIVOTAL CLUSTERS PLUS ANY OTHER MEANINGFUL DEPENDENT CLUSTERS.

USE THE COMPOSITE SPAN AND CSA FROM STAGE 3B,C.

B. PERFORMING SECTOR ANALYSES TO AID IN THE FINAL DECISIONS, I.E., PROJECT ALL ITEMS OF EACH SECTOR OF SPECIAL INTEREST INTO A PRESET CLUSTER ANALYSIS USING THE PIVOTAL DIMENSIONS OF STAGE 2B.

USE SAMPLER(SELECTING THE SECTOR SAMPLE PLUS PIVOTS)-DAP2-COR2-DVP41-PRESET CC-CSA2-SPAN2 ON DEFINERS).

C. DESCRIBING A REPRESENTATIVE CLUSTER STRUCTURE BY A SINGLE ABRIDGED CLUSTER ANALYSIS OF ALL OBLIQUE CLUSTERS.

INCLUDE IN EACH CLUSTER ITS BEST DEFINERS, FROM STAGE 4A, AND FORM ONE SAMPLE OF 90 SUCH DEFINERS, THEN USE SAMPLER (SELECTING THE SAMPLE)-DAP2-COR2-DVP41-PRESET CC-CSA2-SPAN2.

D. CONCEPTUALIZING EACH OBLIQUE CLUSTER.
FROM A CONTENT ANALYSIS OF THE VARIABLES THAT BEST DEFINE EACH CLUSTER AND FROM THEIR CLUSTER STRUCTURE DESCRIBED IN C, B, AND A, CONCEPTUALIZE IT.

STAGE 5. PREDICTION.--FROM THE PIVOTAL OBLIQUE CLUSTERS, PREDICTING SCORES ON ANY DEPENDENT CLUSTERS OR ON ANY 'OUTSIDE' VARIABLES OF INTEREST THAT WERE NOT INCLUDED IN THE CLUSTER ANALYSIS. USE SMIS (EXAMPLE 1) TO CALCULATE THESE LINEAR MULTIPLE PREDICTIONS.

O-ANALYSIS.--FOLLOWING THE BIGNV ANALYSIS OF VARIABLES, YOU WOULD NORMALLY CALL FACS IN ORDER TO COMPUTE SCORES ON SUBJECTS FOR AN O-ANALYSIS. YOU WOULD THEN CALL RSCAT FOR THE PURPOSE OF STUDYING THE LINEAR AND CURVILINEAR RELATIONS BETWEEN THE OBLIQUE CLUSTERS.

THE PROCEDURES OF O-ANALYSIS PROPER DIRECTLY UTILIZE BIGNV. HERE IS A SUCCINCT SUMMARY OF THE FIVE STAGES OF BIGNV IN O-ANALYSIS AS THEY ARE AUGMENTED BY SPECIAL COMPONENTS DESCRIBED IN THE O-ANALYSIS SECTION OF THE MANUAL.

IN STAGE 1, SAMPLER IS USED TO DRAW STRATIFIED SUBSAMPLES OF 90 SUBJECTS FROM THE FULL SUPPLY OF NS CASES. THE PROCEDURES OF EUCO ANALYSIS (Q.V.) REQUIRE CONVERTING THE EUCLIDEAN DISTANCE MATRICES TO CORRELATION-MATRICES ON THE SUBSAMPLES IN ORDER TO DETERMINE THE DIMENSIONALITY OF THE FULL SUPPLY.

IN STAGE 2, PRESET EUCO-ANALYSES ARE PERFORMED ON THE SUCCESSIVE SAMPLES OF SUBJECTS, SUPPLEMENTED BY OMARK ANALYSIS (Q.V.).

IN STAGE 3, MERGER AIDS IN RECOMBINING THE SAMPLES TO FORM CORE O-TYPES IN THE SUPPLY.

IN STAGE 4, DESCRIPTION OF THE COMPLETE O-CLUSTER STRUCTURE OF THE FULL SUPPLY OF NS SUBJECTS IS AIDED BY EUFIT AND OSTAT(Q.V.).

IN STAGE 5, PREDICTION IS BY THE NON-LINEAR PREDICT PROCEDURES (Q.V.).

REALISTIC BIGNV.--THE PRESENT UPPER LIMITS OF 2000 VARIABLES AND 10,000 SUBJECTS APPEAR NOT TO BE FEASIBLE LIMITS ON COMPUTERS WITH TAPE SIZES AND SPEEDS COMPARABLE TO THOSE OF IBM 7094. BY THE TIME THE SAMPLER PROCEDURES ARE PROGRAMMED, REALISTIC LIMITS ON NV AND NS WILL BE KNOWN. PRESENTLY WE FIND THAT COMPUTING A V-ANALYSIS AND AN O-ANALYSIS BY IBM 7094 ON 600 VARIABLES AND 1000 SUBJECTS WITHOUT MISSING DATA WOULD REQUIRE ABOUT 3 HOURS, BUT WITH MISSING DATA ABOUT 5 HOURS.

USERS' DESCRIPTION OF SAMPLER
(PART OF BIGNV PROCEDURES)

INTRODUCTION.

THE PURPOSES OF SAMPLER ARE TO EXECUTE THE SAMPLING PROCESSES OF BIGNV PROCEDURES. RECALL THAT WHEN THE FULL SUPPLY OF VARIABLES, NV, IS VERY LARGE, YOU CANNOT DIRECTLY DISCOVER THE RESULTS OF A V-ANALYSIS ON THE FULL SUPPLY. YOU MUST CONFINE YOUR COMPUTER RUNS TO SAMPLES OF NOT MORE THAN 90 VARIABLES, THE RESULTS OF WHICH WILL LATER BE SYNTHESIZED FOR THE FULL SUPPLY BY MERGER. WHEN YOU PERFORM AN O-ANALYSIS ON MORE THAN 90 SUBJECTS YOU ALSO NEED SAMPLER FOR PRECISELY THE SAME KIND OF SAMPLING FROM THE FULL SUPPLY OF SUBJECTS, NS. SAMPLER IS THE COMPOUND OF COMPONENTS THAT DESIGNS AND DRAWS SUCH SAMPLES. DEPENDING ON THE STRATA-STRUCTURE OF THE SUPPLY, AND ON THE STAGE OF THE BIGNV ANALYSIS, THE SAMPLES THAT ARE DRAWN FROM THE FULL SUPPLY WILL BE EITHER SIMPLE RANDOM SAMPLES, OR THEY MAY BE STRATIFIED RANDOM, OR EXPRESSLY DESIGNATED SAMPLES, OR COMBINATIONS OF THESE THREE TYPES OF SAMPLING.

'IN O-ANALYSIS, SAMPLER SELECTS SUBSAMPLES OF SUBJECTS FROM THE FULL SUPPLY OF O-NAMES AT SUCCESSIVE STAGES OF THE ANALYSIS PARALLEL WITH THOSE OF V-ANALYSIS. THE SUBSUPPLIES OF O-NAMES FROM WHICH SAMPLES ARE DRAWN CAN, HOWEVER, BE TAGGED BY PATTERNS OF STRATA AS WELL AS BY SINGLE STRATA WITHIN A CLASSIFYING VARIABLE.

METHOD.

IF YOU LOOK BACK AT THE FIVE STAGES OF BIGNV ANALYSIS AS THEY ARE DETAILED IN THE USE SECTION OF BIGNV, YOU WILL DISCOVER THAT THE METHOD OF SAMPLING FROM THE SUPPLY DEPENDS ON THE PARTICULAR DESIGN OF THE SAMPLE NEEDED AT THE GIVEN STAGE. THUS IN STAGE 1, STEP A, IN WHICH THE FULL SUPPLY IS DIVIDED INTO SAMPLES IN WHICH ESTIMATES OF COMMUNALITIES OF THE VARIABLES ARE CALCULATED, A STRATIFIED RANDOM SAMPLE IS DESIRABLE. DETERMINING THE COMMUNALITY OF A VARIABLE REQUIRES THAT SEVERAL REFERENCE VARIABLES MOST COLLINEAR WITH IT BE USED (SEE DVP). THEREFORE, THE SAMPLE OF 90 VARIABLES IN WHICH IT IS EMBEDDED SHOULD BE ONE IN WHICH ALL VARIABLES OF THE SUPPLY HAVE EQUAL OPPORTUNITY TO BE CHOSEN AS REFERENTS. RANDOM SAMPLING FROM STRATA INTO WHICH THE VARIABLES MAY BE MEANINGFULLY SEPARATED ASSURES SUCH EQUALITY. THE FULL SUPPLY MUST BE CAST INTO SUCH STRATIFIED SAMPLES, I.E., SUB-SUPPLIES OF VNAMS. THE LAST SAMPLE TO BE SELECTED FROM THE SUPPLY MAY NOT COME OUT EXACTLY TO 90 VARIABLES, HENCE TO ASSURE RANDOMNESS OF THE SAMPLE, A STRATIFIED RANDOM SAMPLE OF THE VARIABLES ALREADY USED IN PRIOR SAMPLES MUST BE DRAWN IN ORDER TO FILL OUT THE SAMPLE. AFTER THE SAMPLE DESIGN IS COMPLETED, INCLUDING THE SPECIFICATION OF THE VARIABLES THAT COMPOSE EACH SAMPLE, THE NEXT STEP B OF STAGE 1 CONSISTS OF ACTUAL COMPUTER RUNS ON EACH SAMPLE. IT IS AT THIS STEP THAT THE ACTUAL OPERATION OF THE DRAWING OF THE SAMPLE FROM THE SUPPLY IS MADE.

THIS EXAMPLE SHOWS THE PROCESSES REQUIRED IN SELECTING A SAMPLE. THE STRATIFYING CHARACTERISTICS OF THE VARIABLES IN THE SUPPLY MUST BE OBJECTIVELY DEFINED, OR 'TAGGED' IN ORDER TO ACHIEVE STRATIFICATION. WE MAY CALL THIS PROCESS THE 'TAG' PHASE. THEN THE DIFFERENT SUBSAMPLES MUST BE DEFINED AND THEIR VARIABLES NAMED BY A DESIGN THAT ASSURES THAT ALL TAGGED CLASSES OCCUR IN EACH SAMPLE BY THE SAMPLING RATIO EXISTING IN THE FULL SUPPLY AND ARE TO BE DRAWN UNDER THE CONDITIONS OF RANDOM SAMPLING. WE MAY CALL THIS PROCESS THE 'PICK' PHASE. FINALLY, ON THE BASIS OF THE ABOVE SAMPLING DESIGN SET BY TAG AND PICK, A PARTICULAR SAMPLE MUST BE ACTUALLY DRAWN FROM THE FULL SUPPLY. CALL THIS PROCESS, PRY. THE PROGRAMMING OF THESE THREE PROCESSES WILL REQUIRE COMPONENTS WHICH WE MAY THEREFORE TERM 'TAG', 'PICK', AND 'PRY'.

THESE THREE PROCESSES MAY WORK OUT DIFFERENTLY IN SAMPLING AT OTHER STAGES OF BIGNV ANALYSIS. THUS, IN STAGE 1, STEP C, IN WHICH A SAMPLE OF 90 VARIABLES WITH THE HIGHEST COMMUNALITIES IS REQUIRED, THE PROCESSES SELECT ONLY ONE SAMPLE WHICH IS NOT RANDOM BUT 'FORCED'. IN STAGE 2, A, B, STILL ANOTHER DESIGN IS REQUIRED IN WHICH THE SAMPLING PROCESSES DRAW SAMPLES IN EACH OF WHICH A SET OF CONSTANT PIVOTAL DIMENSION-DEFINERS IS 'FORCED', AND THE REST OF THE VARIABLES ARE NOT RANDOM BUT TAGGED BY THE SIZE OF THEIR COMMUNALITIES. IN STAGE 4, B, EACH SAMPLE OF A 'SECTOR' CONTAINS THE CONSTANT PIVOTS PLUS A 'FORCED' SET OF VARIABLES THAT COMPOSE THE GIVEN SECTOR. IN STAGE 4, C, THE WHOLE SAMPLE IS A FORCED COMPOSITE OF OBLIQUE CLUSTERS CONSISTING OF THEIR 'BEST' DEFINERS.

WHEN SAMPLER IS FINALLY PROGRAMMED, THE ORDINARY USER OF THE SYSTEM WILL NOT NEED TO INVOLVE HIMSELF IN THE COMPLEXITY OF DRAWING SUCH SAMPLES. SINCE THE DIFFERENT KINDS OF REQUIRED SAMPLES FALL INTO A FEW STANDARD TYPES, IT IS INTENDED TO PROGRAM EACH STANDARD TYPE AS A 'MACRO'. ALL THE USER WILL THEN NEED TO DO IS TO DESIGNATE THE TYPE OF MACRO HE NEEDS AT A PARTICULAR STAGE IN THE ANALYSIS, AND THE APPROPRIATE TAGGING, PICKING, AND PRYING OUT THE SAMPLES WILL BE EXECUTED BY SAMPLER ITSELF.

SAMPLER DOES, OF COURSE, HAVE A GENERAL PURPOSE USE BEYOND THAT OF BIGNV ANALYSIS. THE ELASTICITY OF THE PROGRAMS THAT EXECUTE THE PROCESSES OF SAMPLING PERMIT ANY KINDS OF SAMPLING OF VARIABLES OR OF OBJECTS FOR ANY PURPOSE.

USE OF SAMPLER.

UNTIL SAMPLER IS PROGRAMMED, THE FORM OF THE CONTROL CARDS OF THE COMPONENTS THAT WILL COMPOSE IT IS NOT YET KNOWN.

IN BERKELEY, THE ABOVE PROCESSES OF SAMPLING CAN PRESENTLY BE EXECUTED BY CERTAIN REPROGRAMMED IHD PROGRAMS WHOSE MODIFIED FORMS ARE NAMED PRIEST AND TRYST. FOR THE USE OF THEM SEE THE CONSULTANT.

AT CENTERS OUTSIDE BERKELEY, THE USER WILL EXECUTE THE PROCESSES OF SAMPLER 'BY HAND'. IN V-ANALYSIS HE CAN FORM THE PROPER SAMPLES

REQUIRED AT THE VARIOUS STAGES OF BIGNV ANALYSIS BY THE USE OF THE FORMAT-CONVERSION OPTION INFORMAT IN DAP2. SINCE INFORMAT CAN TAKE NO MORE THAN 20 CARDS, THE USER MAY BE ABLE TO MANAGE A SUPPLY OF NOT MORE THAN, SAY, 400 VARIABLES. IF HE HAS MANY MORE, HE WILL NEED TO USE TABULATING EQUIPMENT TO TRANSFER ONTO SAMPLE DECKS THE PARTICULAR VARIABLES NEEDED IN EACH SAMPLE. IN O-ANALYSIS, THE PRINCIPLES OF SAMPLER ARE MORE EASILY EXECUTED BY HAND THAN IN V-ANALYSIS. USING FACS, THE USER SIMPLY CALLS FOR THE OUTPUT OF FSCOR1 CARDS OF HIS FULL SUPPLY OF SUBJECTS. THEN, AFTER HE HAS FIGURED OUT THE SAMPLE DESIGN, HE NEEDS ONLY TO DRAW THE CARDS OF THE SUBJECTS THAT COMPOSE THE SAMPLE, AND INPUT THESE CARDS INTO EUCO. RANDOM DRAWS CAN BE ACHIEVED EITHER BY THE USE OF A TABLE OF RANDOM NUMBERS OR BY CARD SHUFFLING., IN THE COMPUTER COMPONENTS, RANDOM SELECTION IS ACHIEVED BY A PSEUDO-RANDOM NUMBER GENERATOR.

USERS' DESCRIPTION OF MERGER (SETCOPY, COPY, PRANK) (PART OF BIGNV PROCEDURES)

INTRODUCTION

MERGER IS A COMPOUND DESIGNED TO RECOMBINE ANY OR ALL IST FILES THAT ARE COMPUTED ON THE DIFFERENT SAMPLES OF VARIABLES SELECTED BY SAMPLER INTO MASTER LISTINGS ON THE FULL SUPPLY OF VARIABLES (OR SUBJECTS). THE MASTER LISTINGS ARE PRINTED BY THE ORDINAL NUMBERS OF THE VARIABLES, WITH THEIR VNAMS, OR BY THE RANKS OF THEIR FILE VALUES IN THE FULL SUPPLY FROM WHICH THE SAMPLES ARE DRAWN. A SPECIAL TYPE OF MASTER 'TAG-ALONG' LISTING SUCH AS IN CSA THE VECTOR OF COMMUNALITIES AS A TAG-ALONG AFTER THE RANKED OBLIQUE FACTOR COEFFICIENTS IS ALSO PROVIDED. IN EACH IST FILE, VALUES IN A SPECIFIED UPPER PERCENTAGE OF ALL CASES OR ABOVE A SPECIFIED THRESHOLD VALUE CAN ALSO BE STARRED BY ASTERISKS.

METHOD

MERGER IS A COMPOUND CONSISTING OF THREE NEW COMPONENTS.. (1) SETCOPY IS A SUBEXECUTIVE THAT ACTIVATES A SPECIAL TAPE, MTDEX, FOR THE INPUT OF IST FILES OF THE SAMPLES THAT ARE TO BE MERGED. (2) COPY WRITES THE FILES OF EACH SAMPLE ON MTDEX AS SPECIFIED BY SETCOPY. (3) PRANK PERFORMS THE ACTUAL MERGING OF EACH OF THE FILES ON MTDEX AND PRINTS THEM ACCORDING TO OPTIONS SUPPLIED BY THE USER, UNDER CERTAIN SPECIFICATIONS OF STARRING THE VARIABLES AS DIRECTED IN SETCOPY.

MERGER IN O-ANALYSIS.--THOUGH THE FOLLOWING USE SECTIONS ARE WRITTEN TO APPLY IN V-ANALYSIS TO THE RECOMBINING OF IST FILES OF VARIABLES INTO FILES OF THE FULL SUPPLY OF NV VARIABLES, THE SECTIONS EQUALLY APPLY IN O-ANALYSIS TO MERGING OF IST FILES OF SUCCESSIVE RANDOM SAMPLES OF SUBJECTS INTO FILES OF THE FULL SUPPLY OF NS SUBJECTS. RECALL THAT IN O-ANALYSIS THE DATA CARDS OF EACH SAMPLE OF 90 SUBJECTS ARE DRAWN FROM THE FULL SUPPLY OF NS SUBJECTS EITHER BY HAND OR BY SAMPLER, AND ARE INPUT AS A SAMPLE INTO FACS. THIS SCORING COMPONENT OUTPUTS THE CLUSTER SCORES TO THE COMPONENT EUCO WHICH CONVERTS THE SAMPLE OF 90 SUBJECTS TO 90 'VARIABLES' BY TREATING THE COLUMNS OF EUCLIDEAN DISTANCES OF SUBJECTS AS RAW DATA VARIABLES WHICH IT OUTPUTS AS SUCH TO THE DATA STORAGE TAPE, DST. AT THE SAME TIME, EUCO TRANSFORMS THE 90 LABELS IN THE ONAMS1 FILE TO 90 IN THE VNAMS1 FILE. FROM THEN ON, SUBJECTS ARE TREATED AS VARIABLES WHOSE IST FILES ARE MANIPULATED BY MERGER AS DESCRIBED HERE.

LIMITATIONS ON IST FILES.--SINCE MERGER IS DESIGNED TO RECOMBINE VARIABLES OR SUBJECTS-AS-VARIABLES, IT MERGES ONLY IST FILES WITH NV ROWS. EXCLUDED FROM MERGING ARE FILES WITH LESS THAN NV ROWS (E.G., MEANS2) AND FILES OF NON-ALGEBRAIC DATA (E.G., IDFILE, REFLX1). THE FILES ON WHICH YOU MAY APPLY MERGER ARE THIS LIMITED LIST..

VECTORS	RECTANGULAR MATRICES	SYMMETRIC MATRICES
VSUMS1	UFACT1	CORRM1
MEANS1	RFACT1	COVAR1
STDEV1	WEIGH1	
DIAGV1		

USE OF SETCOPY

THE COMPONENT SETCOPY SETS UP THE SPECIAL STORAGE TAPE, MTDEX, ON WHICH THE SPECIFIED IST FILES OF THE DIFFERENT SAMPLES OF VARIABLES ARE TO BE WRITTEN BY COPY AND MERGED BY PRANK.

IMPORTANT.. IN BERKELEY NO SPECIAL MOUNTING OF A TAPE FOR MTDEX TAPE IS NECESSARY UNLESS YOU WISH TO SAVE THE MTDEX TAPE, IN WHICH CASE YOU MUST SUPPLY AN EXTRA REEL TO THE OPERATOR WITH A REQUEST TO MOUNT IT ON A5 ('LOGICAL 5') AND SAVE IT. IN OTHER COMPUTER CENTERS, IT MAY ALWAYS BE NECESSARY TO MOUNT A REEL FOR THE SYMBOLIC TAPE, MTDEX (SEE THE LOCAL BC TRY CONSULTANT FOR THE LOGICAL/PHYSICAL UNIT).

```
     CARD INPUT.  THREE CARDS ARE REQUIRED BY SETCOPY.                   ****
                                                                         ****
     (1)   SETCOPY EXECUTIVE CARD.  PUNCH '/SETCOPY' IN COLS. 1-8.       ****
           IMPORTANT.. INSERT THIS EXECUTIVE CARD ONLY ONCE AND THEN     ****
           IMMEDIATELY AFTER THE '/START' CARD OF THE FIRST SAMPLE.      ****
                                                                         ****
     (2)   COMPONENT CONTROL CARDS.  THERE ARE TWO COMPONENT CARDS.      ****
           CARD A 'INITIALIZES' (ACTIVATES) THE MTDEX TAPE WITH RESPECT  ****
           TO THE METHOD BY WHICH INFORMATION WILL GO ON IT, AND CARD    ****
           B SPECIFIES THE NATURE OF THE PARTICULAR IST FILES THAT       ****
           WILL BE COPIED ON IT.                                         ****
                                                                         ****
     CARD A.  THE ORIGINAL METHOD BY WHICH THE FULL SUPPLY OF VARIABLES  ****
HAS BEEN DIVIDED INTO SAMPLES TO BE INPUT ON THE STORAGE TAPE, MTDEX.    ****
                                                                         ****
                                                                         ****
             COLUMNS          CONTENTS                                   ****
                                                                         ****
             1-6        METHOD OF DIVIDING THE FULL SUPPLY AND           ****
                        INPUTTING IT TO MTDEX BY COPY.                   ****
                                                                         ****
                        BLANK.  THE DIVISION OF THE FULL SUPPLY INTO     ****
                        SAMPLES HAS BEEN MADE AND RECORDED ON THE MTDEX  ****
                        TAPE BY THE PROGRAM PICK (CURRENTLY AT BERKELEY  ****
                        THE PROGRAMS PRIEST AND TRYST, REVISED IHD       ****
                        PROGRAMS, ARE SUBSTITUTED FOR PICK).             ****
                                                                         ****
                        'CLEAN'.  THE DIVISION IS INITIATED ON ONE       ****
                        COMPUTER RUN BY THE USER INPUTTING THE IST FILES ****
                        OF THE SAMPLES BY DECKS OF THEIR GIVE CARDS      ****
                        (SEE COPY).                                      ****
```

CARD B. FILES CARDS, SPECIFYING THE IST FILES THAT ARE TO BE MERGED INTO A SINGLE MASTER LIST BY PRANK LATER. IF LEFT BLANK, A STANDARD SET OF FILES WILL BE MERGED. BUT IF YOU WANT SPECIFIC FILES, A SEPARATE CARD B IS REQUIRED FOR EACH FILE. IF YOU WANT IST FILES THAT ARE VECTORS OR RECTANGULAR MATRICES THE FILE CARDS WILL NORMALLY ONLY REQUIRE THE NAMES OF THE FILES AND WILL OTHERWISE BE BLANK. THERE ARE TWO CARD B OPTIONS..

(1) BLANK CARD B (STANDARD). THE FILES TO BE STORED BY COPY AND MERGED BY PRANK ARE THE FOLLOWING.. MEANS1, STDEV1, DIAGV1. (SEE THE APPENDIX TO MERGER FOR STARRING OF PRINTED VALUES, ESPECIALLY, MODE, COLS. 9-12 IN APPENDIX.

(2) PUNCHED CARDS B. THE PARTICULAR IST FILES TO BE STORED BY COPY AND MERGED BY PRANK ARE EXPRESSLY DESIGNATED BY THE USER. ONE CARD IS REQUIRED FOR EACH IST FILE THAT IS DESIRED.
IMPORTANT.. FOLLOW THESE PUNCHED CARDS BY A BLANK CARD.

THE FORMAT OF EACH CARD IS AS FOLLOWS..

COLUMNS	CONTENTS
1-6	FILE NAME. PUNCH THE CODE NAME OF THE DESIRED FILE.
9-12	MODE. LEAVE BLANK (STANDARD). (BUT SEE THE APPENDIX TO MERGER ON THE STARRING OF PRINTED VALUES.)
13-20	CUT. LEAVE BLANK. (BUT SEE THE NOTE IN COLS. 9-12 JUST ABOVE.)
21-24	NCOLS, REQUIRED ONLY WHEN THE DESIRED IST FILE IS A SYMMETRIC MATRIX, E.G., CORRM1. '0' OR BLANK. THE FILE IS NOT A SYMMETRIC MATRIX (STANDARD). PUNCH AN INTEGER FROM 1 TO 90 DENOTING THE NUMBER OF COLUMNS OF THE MATRIX THAT ARE TO BE STORED. RESTRICTION.. THE COLUMNS SELECTED MUST ALWAYS BE PHYSICALLY THE FIRST COLUMNS OF THE MATRIX.

CARD OUTPUT. THERE IS NO CARD OUTPUT BY SETCOPY.

TAPE INPUT. NEITHER THE IST OR DST TAPES IS REQUIRED TO ACTIVATE THE MTDEX STORAGE TAPE.

TAPE OUTPUT. THERE IS NO OUTPUT TO IST OR DST.

PRINTED OUTPUT. A TABLE OF THE FILES TO BE WRITTEN BY COPY IS PRINTED.

RESTRICTIONS. SETCOPY REFERS ONLY TO THE IST FILES OF BC TRY. WHEN SUCH A FILE IS SYMMETRIC, SEE THE RESTRICTION UNDER CARD B, NCOLS, COLS. 21-24.

USE OF COPY.

RECALL THAT SETCOPY ONLY ACTIVATES MTDEX, THE TAPE ON WHICH THE IST FILES OF THE DIFFERENT SAMPLES ARE TO BE STORED AND LATER MERGED BY PRANK. ACCORDING TO THE INSTRUCTIONS OF SETCOPY, THE COMPONENT COPY ACTUALLY DOES THE COPYING OF THE ACTUAL IST FILE VALUES OF EACH SAMPLE.

USE OF COPY IS SIMPLE. CALL COPY AFTER THE DESIRED IST FILES HAVE BEEN COMPUTED ON EACH SAMPLE.

CARD INPUT. ONLY THE EXECUTIVE CARD IS USED.

(1) COPY EXECUTIVE CARD. PUNCH '/COPY' IN COLS. 1-5. WHEN YOU USE PICK (OR ITS PRESENT SUBSTITUTES IN BERKELEY) CALL COPY AFTER THE COMPUTER RUN ON EACH SAMPLE. WHEN, HOWEVER, YOU USE THE TAKES ON THE GIVE-TAKE DECKS OF THE SAMPLES, YOU WILL BATCH THEM IN ONE COMPUTER RUN, INSERTING A COPY EXECUTIVE CARD AFTER THE GIVE DECK OF EACH SAMPLE.

USE OF PRANK.

CARD INPUT. THE EXECUTIVE AND THREE COMPONENT CONTROL CARDS ARE USED.

(1) PRANK EXECUTIVE CARD. PUNCH '/PRANK' IN COLS. 1-6.

IMPORTANT..

IF PRANK IS CALLED ON THE SAME RUN IN WHICH THE COPY PASSES ON THE SAMPLES ARE MADE, INSERT THE PRANK EXECUTIVE CARD AFTER THE LAST COPY EXECUTIVE CARD.

IF, HOWEVER, YOU ARE MOUNTING YOUR MTDEX TAPE ON A LATER, SEPARATE PRANK RUN, START THAT PRANK RUN WITH '/START' CARD FOLLOWED BY THE PRANK EXECUTIVE CARD.

IF PRANK IS RUN FROM BATCHED GIVE DECKS ON THE SAMPLES, INSERT THE PRANK EXECUTIVE CARD AFTER COPY OF THE LAST GIVE DECK.

PRANK REQUIRES AN EXTRA SCRATCH TAPE CALLED MTRANK, USUALLY NOT SAVED. IN BERKELEY A SCRATCH TAPE IS ALWAYS AVAILABLE FOR MTRANK ON 'LOGICAL 7' BUT ELSEWHERE A SPECIAL MOUNTING BY THE OPERATOR MAY BE NECESSARY.

(2) COMPONENT CONTROL CARDS. THERE ARE THREE. THE FIRST TWO, CARDS A AND B, ARE NECESSARY. CARD C IS OPTIONAL.

CARD A. TITLE CARD, GIVING THE IDENTIFICATION OF THE PROBLEM HANDLED BY PRANK. PUNCH IN TITLE CARD FORMAT.

CARD B. PARAMETERS AND CONTROLS.

COLUMNS CONTENTS

1-4 NSAMP, THE TOTAL NUMBER OF DIFFERENT SAMPLES BEING MERGED OR PROCESSED BY PRANK, I.E., THE NUMBER OF '/COPY' EXECUTIVE CARDS THAT HAVE BEEN USED.

PUNCH AS AN INTEGER NUMBER THE NUMBER OF SAMPLES, NOT TO EXCEED 25.

5-8 NVTOT, THE TOTAL NUMBER OF VARIABLES ACROSS ALL THE SAMPLES, COUNTING REDUNDANCIES, IF ANY.

PUNCH AS AN INTEGER NUMBER NOT GREATER THAN 2000.

9-12 MERGE, THE POSITIONING OF THE VARIABLES AS THEY ARE MERGED ON THE INTERMEDIATE TAPE, MTRANK, FROM WHICH PRANK PRINTS THE MASTER LIST OF VALUES AS SPECIFIED BELOW IN THE THREE OPTIONS IN COLS. 21 TO 32,'MERGED','RANKED','TAG-ALONG'.

'0' OR BLANK. VARIABLES ARE POSITIONED AND NUMBERED BY THEIR ORDINAL NUMBERS AS THEY WERE INPUT ON DATA CARDS IN DAP2. NORMALLY THESE ORDINAL NUMBERS ARE AVAILABLE TO PRANK ONLY WHEN PICK (OR ITS PRESENT BERKELEY EQUIVALENT) HAS DESIGNED THE SAMPLES.

'3' VARIABLES ARE POSITIONED AS THEY WERE INPUT BY COPY IN THEIR SAMPLES. EACH VARIABLE IS GIVEN THE ORDINAL NUMBER CORRESPONDING TO ITS POSITION IN THE ENTIRE LIST OF SAMPLES PHYSICALLY STACKED IN THE ORDER OF INPUT OF THE SAMPLES. IN SHORT, PUNCH '3' WHEN THE INSTRUCTIONS TO COPY WERE SUPPLIED BY SETCOPY VIA 'CLEAN' UNDER METHOD, CARD A, COLS. 1-6.

13-16 LEAVE THESE COLUMNS BLANK.

17-20 LEAVE THESE COLUMNS BLANK.

21-24 'MERGED' OR PRANK1 PRINTOUT, IN WHICH THE PRINTED ORDER OF THE VARIABLES IS BY THEIR ORDINAL POSITIONS ASSIGNED TO THEM IN MERGE, COLS. 9-12. V-NAMES ARE ALSO PRINTED. ATTACHED TO THE VALUES IN EACH IST FILE ARE THE RANK ORDERS OF THE VALUES USUALLY IGNORING SIGN, BUT SEE THE APPENDIX OF MERGER ABOUT MODE.

'0' DO NOT PRINT IN THIS FASHION (STANDARD).

'1' PRINT IN THIS FASHION ALL IST FILES THAT WERE COPIED.

25-28 'RANKED' FILES, OR PRANK2 PRINTOUT, IN WHICH FOR EACH IST FILE THE VARIABLES ARE PRINTED IN THE ORDER OF RANKS (USUALLY IGNORING SIGN, BUT SEE THE APPENDIX ABOUT MODE). IN ALL CASES, HIGH VALUES RECEIVE LOW RANKS. THEIR V-NAMES AND ORDINAL POSITIONS ASSIGNED IN MERGE, COLS. 9-12, ARE ALSO PRINTED.

'0' DO NOT PRINT IN THIS FASHION (STANDARD).

'-1' PRINT IN THIS FASHION.

29-32 'TAG-ALONG' FILES, OR PRANK3 PRINTOUT, IN WHICH ONE OF THE IST FILES IS DESIGNATED AS 'ANCHOR', AND THE VALUES OF THE VARIABLES IN THIS FILE ARE PRINTED IN RANK ORDER. FOR EACH ROW VARIABLE, DESIGNATED BY ITS V-NAME AND POSITION ASSIGNED IN MERGE, COLS. 9-12, ITS VALUES IN EACH OF THE TAG-ALONG IST FILES, WITH THEIR RANKS ATTACHED TO THESE TAG-ALONG VALUES, ARE PRINTED IN THE SAME ROW BUT FOLLOWING ITS VALUE IN THE ANCHOR FILE.

'0' DO NOT PRINT IN THIS FASHION (STANDARD).

'1' PRINT WITH ALL IST FILES, TAKEN IN TURN AS ANCHOR.

'2' PRINT A SELECTED SET OF ANCHOR AND TAG-ALONG FILES AS INDICATED IN THE FOLLOWING CARDS C, THE FILE DETAIL CARDS.

CARD C. OPTIONAL FILE DETAIL CARDS FOR TAG-ALONG, PRANK3 PRINTOUTS. WHEN ONE OR MORE SELECTED ANCHORS WITH THEIR TAG-ALONG FILES ARE REQUESTED BY THE PUNCH OF '2' IN COLS. 29-32, THEN FILE DETAIL CARDS MUST BE SUPPLIED FOLLOWING CARD B. THERE WILL BE ONE CARD FOR EACH PRINTOUT DESIRED SPECIFYING THE NAMES OF ANCHOR AND TAG-ALONGS FOR THAT PRINTOUT. THE NAMES OF THE REQUESTED IST COLUMNS (SEE BELOW) ARE PUNCHED ON FREE FIELD NAME CARDS, ANCHOR FIRST, TAG-ALONGS NEXT, THE LAST TAG-ALONG ENDING IN A PERIOD. USE AS MANY CARDS AS YOU LIKE., AS A SPECIAL CASE, IF YOU PUNCH A CARD WITH ONLY THE NAME OF THE ANCHOR, PRANK ROUTINELY PRINTS AS TAG-ALONGS ALL THE OTHER COLUMNS OF FILES REQUESTED BY SETCOPY, CARDS B. THE TOTAL NUMBER OF

PRINTED IST COLUMNS CANNOT EXCEED 150. REMEMBER THAT WHEN A REQUESTED ****
IST IS A MATRIX ALL OF ITS COLUMNS WILL BE TAG-ALONGS. ****
****
IMPORTANT.. FOLLOW THE LAST OF CARDS C BY ONE PUNCHED 'TAGEND' ****
IN FREE FIELD FORMAT. ****
****
NAMES OF IST COLUMNS WHEN THE REQUESTED FILES ARE MATRICES.-- ****
RECALL THAT IST FILES MAY BE VECTORS (OR SINGLE COLUMNS, SUCH AS ****
MEANS1), RECTANGULAR MATRICES (SUCH AS THE OBLIQUE FACTOR COEFFICIENTS, ****
RFACT1), OR SYMMETRIC MATRICES (SUCH AS CORRM1). EVERY COLUMN OF ****
VALUES OF VECTORS OR MATRICES MUST BE PUNCHED ON CARDS C. ****
WHEN AN IST FILE IS A VECTOR, ITS SINGLE COLUMN NAME IS, OF COURSE, ****
ITS SINGLE IST FILE NAME. WHEN THE REQUESTED FILE IS A RECTANGULAR ****
MATRIX, PRANK WILL ASSIGN ALL OF ITS COLUMNS UNIQUE NAMES WHICH YOU ****
MUST PUNCH IF YOU WISH SOME BUT NOT ALL OF ITS COLUMNS. ASSIGNED ****
COLUMN NAMES ARE SIMPLY THE FIRST THREE DIGITS OF THE CODE NAME OF ****
THEIR IST FILE FOLLOWED BY THREE DIGIT NUMBERS SIGNIFYING THEIR ORDINAL ****
POSITIONS IN THE FILE, E.G., FACTOR COEFFICIENTS ON THE 3RD DIMENSION ****
OF UFACT1 HAVE THE NAME 'UFA003'. WHEN YOU REQUEST SOME OF THE FIRST ****
NCOL COLUMNS OF THE IST FILE OF A SYMMETRIC MATRIX SUCH AS CORRM1, OR
COVAR1 (SEE SETCOPY, CARD B, COLS. 21-24), THEN FOR EACH COLUMN OF ****
THE MATRIX REQUESTED PUNCH ITS V-NAME. YOU DO NOT, HOWEVER, HAVE TO ****
PUNCH COLUMN NAMES OF A SELECTED TAG-ALONG MATRIX IF YOU WANT ALL OF ****
ITS COLUMNS--SIMPLY PUNCH ITS IST FILE NAME. ****

CARD OUTPUT. THERE IS NO CARD OUTPUT FROM PRANK.

TAPE INPUT. THE TAPE INPUT TO PRANK IS THE MTDEX TAPE.

TAPE OUTPUT. NO IST OR DST TAPE OUTPUT IS EXECUTED.

PRINTED OUTPUT. THE PRINTED OUTPUT IS DESCRIBED IN THE THREE OPTIONS OF 'MERGED', 'RANKED', AND 'TAG-ALONG'.

APPENDIX TO SETCOPY CONTROL CARDS B

STARRING VARIABLES IN UPPER PERCENTAGE CLASSES OF THE MERGED MASTER IST FILE, OR ABOVE SPECIFIED THRESHOLD VALUES.

IN THE PRINTOUT OF THE MERGED MASTER FILE OF IST VALUES, IT IS OFTEN DESIRABLE TO STAR PARTICULAR VARIABLES THAT ARE IN THE UPPER PART OF THE DISTRIBUTION OF ALL VARIABLES IN THE FULL SUPPLY. PRANK CAN MARK SUCH VARIABLES BY INSERTING AN ASTERISK AFTER THEIR RANKS. YOU DECIDE FOR EACH IST FILE ON THE 'MODE' OF STARRING, I.E., WHETHER THE MARKED VARIABLES BELONG TO A TOP-RANKING PERCENTAGE CLASS OR LIE ABOVE A NUMERICAL THRESHOLD VALUE. YOU ALSO SUPPLY THE 'CUTTING POINT', A PERCENTAGE IN THE FORMER CASE, THE PARTICULAR THRESHOLD VALUE IN THE LATTER. FOR EXAMPLE, IF YOU DECIDE ON A PERCENTAGE MODE, YOU MAY WISH TO STAR ALL VARIABLES WHOSE COMMUNALITIES ARE IN THE UPPER

50 PER CENT OF THE WHOLE DISTRIBUTION OF COMMUNALITIES. BUT IF YOU CHOOSE A THRESHOLD MODE, YOU MAY WISH TO STAR ALL VARIABLES WHOSE COMMUNALITIES ARE AT LEAST .20.

YOUR DECISIONS ARE MADE SEPARATELY FOR EACH FILE BY INFORMING THE SUBEXECUTIVE, SETCOPY, ON ITS CONTROL CARDS B HOW IT IS TO INSTRUCT PRANK TO HANDLE THE MODE AND CUT. THESE OPTIONS ARE PROVIDED VIA SETCOPY, 'CARDS B', COLS. 9-12 AND COLS. 13-20, AS FOLLOWS..

```
      COLUMNS           CONTENTS

      9-12       MODE, STARRING IN A TOP-RANKING PERCENTAGE        ****
                 CLASS, OR IN AN ABOVE-THRESHOLD CLASS.             ****
                                                                    ****
                 '-1' OR BLANK.  STAR THE TOP-RANKING PERCENTAGE    ****
                 CLASS., RANK BY MAGNITUDE (STANDARD).              ****
                                                                    ****
                 '+1'  STAR THE TOP-RANKING PERCENTAGE CLASS.,      ****
                 RANK BY ALGEBRAIC VALUE.                           ****
                                                                    ****
                 '0'  DO NOT STAR ANY VALUES., RANK BY MAGNITUDE.   ****
                                                                    ****
                 '-2'  STAR THE VALUES AT OR ABOVE THRESHOLD.,      ****
                 RANK BY MAGNITUDE.                                 ****
                                                                    ****
                 '+2'  STAR THE VALUES AT OR ABOVE THRESHOLD.,      ****
                 RANK BY ALGEBRAIC VALUE.                           ****
                                                                    ****
      NOTE.. IF CARD B IS LEFT BLANK, MODE IS A PERCENTAGE AND      ****
      CUT (BELOW) IS 10 PERCENT.                                    ****
                                                                    ****
      13-20      CUT, THE NUMERICAL VALUE OF THE CUT-OFF.           ****
                                                                    ****
                 IF MODE IS PLUS OR MINUS 1, OR BLANK, THEN         ****
                 PUNCH THE PERCENTAGE AS A NUMBER BETWEEN 0 AND     ****
                 100.  IF YOU WANT FRACTIONS OF A PERCENT THE       ****
                 DECIMAL CAN BE PUNCHED ANYWHERE OR IF NOT          ****
                 PUNCHED IT WILL BE IMPLIED BETWEEN COLS. 16        ****
                 AND 17.                                            ****
                                                                    ****
                 IF MODE IS PLUS OR MINUS 2, THEN PUNCH THE         ****
                 THRESHOLD VALUE YOU WISH ALL STARRED VARIABLES     ****
                 TO HAVE OR TO EXCEED, USING THE SAME FORMAT        ****
                 GOVERNING DECIMALS AS STATED ABOVE FOR             ****
                 PLUS OR MINUS 1.                                   ****
```

NOTE.. YOU WOULD NOT STAR VARIABLES IN A FILE IF THEY CANNOT BE MEANINGFULLY COMPARED. IN MEANS1, FOR EXAMPLE, STARRING WOULD NOT BE OPTED IF THE VARIABLES ARE SCALED IN DIFFERENT KINDS OF UNITS, E.G., I.Q., HEIGHT, INTROVERSION.. THEY MUST BE IN THE SAME METRIC, E.G., A DICHOTOMOUS SCALE, A RATING SCALE, STANDARDIZED SCORES ON A COMMON OUTSIDE NORM GROUP, THE SAME PHYSICAL SCALE, AND SO ON. RECALL ALSO THAT SOME IST FILES HAVE GROSSLY DIFFERENT MEANINGS IN V-ANALYSIS

O-ANALYSIS. FOR EXAMPLE, IN THE FORMER, MEANS1 REFERS TO RAW OBSERVATION DATA INPUT ON DST BY DAP, WHEREAS IN O-ANALYSIS MEANS1 REFERS TO AVERAGE EUCLIDEAN DISTANCES OF THE VARIABLES-AS-SUBJECTS PUT ON DST BY EUCO.

User's description of the factor and cluster score program: FACS

Introduction

FACS calculates and lists the actual scores of the objects (individuals) on the general dimensions derived by factoring, and it computes the raw R's between these scores. A score on any particular dimension is a composite score in which the component subscores that form the composite are the actual observations on the variables that define the dimension in the factoring procedure. Since FACS is designed to compute scores on dimensions from any kind of prior factoring, the composites are formed in different ways depending on the factoring procedure.

When the prior factoring is subset (key cluster) factoring (CC or NC), then the composite cluster score on a given oblique dimension is a simple sum of standard scores on the defining variables of the cluster, i.e., on the oblique cluster as listed in CSA (see Ghiselli, 1964, pp. 178-179).

When the prior factoring is total-set factoring as by CENT from CC5 or by PRIN, CANON, or AUG from FALS, then the composite score on each independent dimension includes all N variables as components of the score. The composite is therefore an orthogonal regression score in which each of the N scores has a multiple regression weight attached to its standard scores. Though these regression scores are in theory estimates of scores on the orthogonal dimensions, the computed estimates are in fact always correlated (see Method).

FACS also provides oblique regression scores designed to estimate composite scores on oblique dimensions, defined by the total set of N variables entering into each composite as a component. Thus, regression scores on the oblique cluster domains (or 'factors') given in CSA may be determined. Oblique scores on dimensions derived by any other oblique rotation program not in BCTRY can also be computed, provided the appropriate file-packages are input by GIST (see Use, below).

The final scores computed by FACS can be scaled in any linear fashion the analyst wishes. The standard procedure is to give them a mean of 50, and a standard deviation of 10. Analysts who look forward to an O-analysis of individuals based on their FACS scores may wish a different scaling, such as by adjusting the variance of each composite so that it is proportionate to the communality exhausted by its dimension in the factoring procedure.

The broad generality of FACS provides a wide variety of additional uses. Since it permits the analyst to input the weights of the component variables that form a composite, he can rationally adjust the weights of defining variables of a dimension in any way he wished. Indeed, he can forego factoring entirely and form any rational composites of the variables he wishes (by inputting the desired REFLX1 file-package via GIST with all the options of differential weighting of components or of scaling the composites as described above (see Ghiselli, 1964, chap. 10)).

Of special interest is the use of FACS in test construction. For example, when the N variables are N test items, FACS can be used to score and list up to 15 subscales of the N items (by inputting SD per item in the weight matrix).

Method

Simple sum cluster score. The logic of such a cluster score is simple: the Z scores of the individual on the defining variables are computed and added up (the standard case). If the analyst wishes differential weighting of the component definers he can figure out what he wants and input them. Either way, the result is the individual's cluster score, except for scaling (see below).

Orthogonal regression score. In principle, the procedure is straight multiple regression analysis (see Thomson, 1951, ch. 15). A given dimension derived by factoring is adjoined to the R-matrix as a 'criterion' column containing as elements the factor coefficients on the NV variables; the NV columns are conceptualized as 'predictor' variables. Then regression coefficients that maximize the multiple R between predictor and criterion are calculated on the NV variables. The added condition is imposed that all the K criterion dimensions are uncorrelated, and the calculations are simplified by dealing not with the raw correlation matrix but with the reproduced matrix (i.e., such as that given in FAST), a procedure that designates the method as Lederman's 'short-cut.'

Oblique regression score. The oblique regression design is the same as that above, except that the condition of independence of the different regression scores is relaxed.

Scaling constants. The linear transforming of the raw composite scores to a new set with a preset mean and standard deviation is a standard psychometric procedure. The individual's raw composite score is first converted to a Z score, then multiplied by the desired new standard deviation, to the product of which is added the desired mean. Most analysts will wish the FACS score to be expressed in standard form, with a mean of 50 and standard deviation of 10, provided routinely in FACS by blank control cards.

Use

Important note: FACS is designed for use only on problems with complete data (see FACS 3 for scoring with missing data). If the scores of an individual are missing on some variables, a program stop occurs in FACS.

```
     Card input.                                                     ****
     (1) Executive control card.  Punch '/FACS++' in cols. 1-7.      ****
     (2) Component control cards.  Three or more cards are required. ****
The first three cards determine the type of composite desired, the   ****
desired mean and standard deviations, and the identifying title of the ****
standardized composite scores.  These three cards are required for all ****
control options and additional card input is required by some control ****
options.  All cards must be input in the order in which they are     ****
described.                                                           ****
                                                                     ****
     Card A.  Type of composite and its standard deviation.  This card ****
specifies the type of composites desired, and the scaling factors to b ****
used as the standard deviations of the standardized composites.  Card ****
is an integer card.  Blank columns yield standard options.           ****
                                                                     ****
     COLUMNS AND CONTENTS OF CARD A                                  ****
                                                                     ****
     1-4     Specifies the type of composite score desired.          ****
             '1'.  Use simple sum method (standard).                 ****
             '2'.  Use orthogonal regression score method.           ****
             '3'.  Use oblique regression score method.              ****
             '4'.  Weights input by analyst  (see description of     ****
                   'Cards D' below).                                 ****
                                                                     ****
     5-8     Specifies the method of setting the scaling factors     ****
             (standard deviations) of the composite scores.          ****
             '1' C. Constant value method:  all standard deviations  ****
                   equal to 10 (standard).                           ****
             '2'.  Communality exhaustion method:  standard deviations ****
                   proportioned to the square root of the communality ****
                   exhausted by the corresponding dimension.         ****
             '3'.  Standard deviations are input by the analyst      ****
                   (see 'Cards E' below).  Note:  the input scaling  ****
                   factors are the square roots (sic) of the desired ****
                   variances.                                        ****
                                                                     ****
     Card B.  Value of mean.  This card contains the mean value scaling ****
constant of the composite scores, punched in cols. 1-8, using a decimal ****
card format.  Leaving Card B blank will yield the standard value (50.) ****
                                                                     ****
     Card C.  Title.  This card contains an identifying title which is ****
associated with the composite scores in the printed output.  Card C is ****
punched as title card.                                               ****
                                                                     ****
     Cards D.  Weighting of components (optional).  These cards are  ****
needed only when the weight matrix of component variables is input by ****
the analyst (i.e., when cols. 1-4 of Card A contain a '4').  Let M be ****
the number of composites desired.  Let W be the matrix of weights for ****
compositing, arranged so that the (G,J) element of W is the weight to ****
be assigned to the score on variable J in composite G,  where G = 1, 2 ****
..., M, and J = 1, 2, ..., NV, the number of variables (i.e., rows   ****
correspond to composites, columns to variables).  The first D card   ****
contains M punched in cols. 1-4 using an integer card format.  The   ****
second and succeeding D cards contain W punched as a matrix using a  ****
decimal card format.                                                 ****
```

Cards E. Non-standard scaling factors (optional). These cards are needed only when the scaling factors of the composites are input by the analyst (that is, when cols. 5-8 of Card A contain a '3'). Let M be the number of composites. Let B(G) be the desired value of the standard deviation for composite G, where G = 1,2,...,M. Then B(1) B(2),...,B(M) should be punched as a vector on decimal Cards E.

Tape input. FACS requires input from both IST and DST for all control options.

(1) Intermediate storage tape (IST). The required inputs for all control options are:

Data identification (IDFILE)
Means of the variables (MEANS1)
Standard deviations of the variables (STDEV1)
Correlation matrix (CORRM1)

The absence of any of these data will cause an error termination. Some control options require data in addition to that in the above four IST files:

When the simple sum method is used (i.e., cols. 1-4 of Card A contains a '1'), the cluster indicators (REFLX1) are required. For preset rational composites REFLX1 must therefore be input via GIST before calling FACS.

When the orthogonal regression method is used (i.e., cols. 1-4 of Card A contain a '2'), the diagonal values (DIAGV1) and the unrotated factor coefficients (UFACT1) are required.

When the oblique regression method is used (i.e., cols. 1-4 of Card A contain a '3'), the diagonal values (DIAGV1), the rotated factor coefficients (RFACT1), and the correlations among the obliquely rotated dimensions (BASIS1) are required.

When the scaling factors (SD's) of the composites are set by the communality exhaustion method (i.e., cols. 5-8 of Card A contain a '2'), the unrotated factor coefficients (UFACT1) are required.

(2) Data storage tape (DST). The program reads scores on the variables which are to be composited from DST. The absence of these scores will yield an error termination. Thus if FACS is not used in the same run as the correlation program, DAP must be used again (unless the analyst has saved the DST).

Printed output includes the clusters used to form simple sum scores or the weight matrices used to form regression scores, the standard deviations of the composites before standardization, the composite scores, and the intercorrelations among the composites.

Tape output. The mean of the composite scores (MEANS2), the standard deviations of the composite scores (STDEV2), the matrix of weights (WEIGH1), and the composite scores (FSCOR1), are written on the intermediate storage tape, regardless of the control options used.

In addition to the standard BCTRY restrictions the following should be noted. Missing data must not be used with FACS. The analyst should remove all incomplete observations from his raw data or substitute values for them and use DAP to write the data storage tape DST with complete data before attempting to use FACS (see FACS 3). If FACS is used on a data storage tape containing missing data, an error termination will occur. The maximum number of composites is 15.

Revision to user's description of FACS

Summary

A provision has now been added to FACS to output a deck of FSCOR1 cards.

Use of FACS

Add (1) another option to component Card A and (2) card output, ****
as follows: ****
****
****

9-12 KPUNCH, the punching of a deck of FSCOR1 cards. ****
****
'0' or blank. Do not punch the FSCOR1 deck ****
(standard). ****
****
'1'. Punch the deck. ****

Card output. The deck contains two information cards, one a title card labelled 'factor scores', and the other giving the format of the scores. The cards with the FSCOR1 data on them follow. There can be more than one card per subject.

User's description of the use of BCTRY components to score subjects that have incomplete data: FACS3

Introduction

The FACS program cannot correctly score individuals if any of them have missing observations on any variable. But many investigators do not have complete scores of all subjects on all variables. If such subjects were deleted from O-analysis the sample size may become too small. And furthermore, the typological location of such subjects could not be made in O-analysis.

FACS3 represents the procedures for utilizing components of BCTRY that will replace missing scores of individuals on a given variable by 'predicted' scores from variables that in general correlate most highly with the given variable. Certainly the investigator should not, as commonly happens, replace such scores with the mean score on the variable - a very poor estimate that might throw the individual's composite scores for O-analysis way off. Perhaps the best estimate would be by a multiple regression estimate from the N-1 other variables but that procedure could get complicated, and furthermore introduce difficult problems of 'shrinkage' when NV is large, and NS not large (Guilford, 1950, formula 16.7, p. 434).

The general principle of FACS3 is to select as the 'predictor' variables of a given variable with missing data only a few variables, usually not more than five, that seem most likely to give the best estimates of missing values. The missing standard score of an individual is then simply replaced by his mean standard score on the predictor variables. After all missing scores of individuals are replaced by this procedure, the score matrix becomes a complete matrix and the regular FACS-EUCO procedures of O-analysis become applicable.

There are two ways in which the analyst may select predictor variables. One is blindly empirical, namely, selecting those whose R's are highest. This method is the one described below. Recall, however, that when the analyst seeks to use FACS to get cluster scores, it is only the defining variables of the clusters whose missing data actually need replacement, not all the NV variables. In choosing predictor variables therefore he may wish to give priority to other variables in the cluster, i.e., to cluster-predictors, rather than blindly to any variables that shows the highest correlations with the variable whose scores are to be predicted. Using cluster-predictors would seem sounder theoretically since they would be sampling the same kind of general domain as the predicted variable. For simplicity, however, the procedure described below is the blind one, though the analyst may wish to alter it by choosing cluster-predictors. In many cases, the selection may work out to be about the same. The Use section below is written especially for the case of securing simple sum cluster scores from FACS, but the extension of the procedures to the case in which regression estimates by the FACS program are desired should be obvious.

Method

The FACS3 procedure consists of three essential steps, assuming the analyst has completed a V-analysis (based on COR3) followed by a GIVE card.

(1) In step 1, the analyst uses the complete-data statistics computed in the prior analysis to generate a Z-score (with M = 50, SD = 10) for each observed score on each variable. These Z-scores will be used to form predicted estimates of Z-scores to replace pieces of missing data.

(2) In step 2, the analyst replaces an individual's missing Z-score on a given variable by the average of his Z-scores on, say, the three variables that correlate highest with the given variable.

(3) In step 3, the data deck from step 2 is now a 'complete-data deck' with no missing data, and is thenceforth treated as such in FACS, and O-analysis.

Use

Note: interpret all output cards as soon as they are received. Interpreted non-binary output cards are required in step 2.

Step 1: generating Z-scores on all data.

(1) DAP. Set parameter card, cols. 13-16, to '0002', to permit missing data to be read as zeros. But DAP here treats these zeros as real scores, hence calculates and stores on IST erroneous M's and SD's.

(2) TAKE (from GIVE, V-analysis). These TAKE cards now restore on IST the correct M's and SD's based on complete cases, calculated in V-analysis.

(3) GIST (REFLX1). To use FACS to compute Z-scores on individual variables, GIST here defines each 'cluster' as composed of one variable only, as follows (assuming the number of 'clusters,' here individual variables, is 15): Operation identifier card: cols. 1-6 'REFLX1'., col. 9-12, 0001. Parameter card 1: number of 'clusters (here, variables), i.e., 15. Parameter card 2: 15 fields, all punched 0001. Data cards: 15 (i.e., NV) Cards. Punch col. 1-4 only; card no. 1, 0001; card no. 2, 0002 ... card no. NV, 00NV.

(4) FACS: all standard options.

(5) GIST (FSCOR1): to get a deck of cards punched with Z-scores of the variables, set operation identifier card: col. 1-6 'FSCOR1'.

At this point REFLX1 and FACS can be used to produce estimates for the pieces of missing data, providing that these pieces all occur within a limited number of variables. If missing data are randomly scattered throughout the NV variables or if only a few scores are missing, hand computations following the logic of the following steps will be more economical.

REFLX1 and FACS can provide estimates for missing data by producing 'phony cluster scores,' i.e., scores obtained by averaging the Z-scores on the three or four variables correlating highest with missing data variables. To set REFLX1 for this purpose, refer to DVP50 or 60 and note the three or four variables with which variable X, a missing data variable, correlates the highest. Set the first REFLX1 cluster for these variables. Repeat this process for the second and subsequent missing data variables. The final number of REFLX1 clusters should equal the number of missing data variables. If variables with negative correlations are used, be sure REFLX1 includes their negative signs.

(6) FACS: all standard options. Here the 'phony cluster scores' are produced.

(7) END.

Note: if more than 15 variables are involved, repeat component programs (3), (4) and (5) above, setting GIST (REFLX1) to variables 16 through NV, in lots of 15 variables. Set a generous (1000) card count on your job card, since numerous non-binary cards will be produced.

Step 2: replacing erroneous FSCOR1 data with estimates of correct scores.

(1) Pick up the interpreted output cards from GIST (FSCOR1) in step 1 (5). Compare these FSCOR1 cards with the FACS printout from step 1 (6). Every FSCOR1 card containing one or more wrong Z scores due to missing data should have one or more replacement scores in the FACS printout. Locate these replacements and write their value above their corresponding erroneous Z-scores on the FSCOR1 cards. Merge a blank card behind each annotated FSCOR1 card. On the IBM keypunch machine, duplicate into the blank card the nonannotated fields of each annotated card and keypunch in the annotation. Remove and throw away the annotated cards. You now have a full deck of 'complete' data for FACS and O-analysis.

Step 3: FACS and O-analysis.

(1) DAP. Use variable format options, setting INFORMAT equal to the FSCOR format. Input as data in DAP the FSCOR1 deck which step 2 produced (be sure to remove operation identifier and parameter cards from FSCOR1 deck). If more than one FSCOR1 deck was obtained from step 1 (i.e., NV is greater than 15) be sure to so merge the FSCOR1 cards and write the informat statement so that the factor scores for each individual are together.

(2) COR2. Note: results of COR2 will usually closely match COR3 results from the initial V-analysis.

(3) GIST (REFLX1). For cluster definers from V-analysis operation identifier card: col. 1-6, REFLX1; col. 9-12, '0001'. Parameter card 1, 2, and 3: set per standard GIST rules to establish desired final FACS variable clusters.

(4) Continue into FACS (with standard options) and into O-analysis.

User's description of the correlation scatter program: RSCAT

(A new Use section herewith replaces that in the February 1964 Manual.)

Introduction

Many analysts want the correlation scattergrams between cluster or factor scores obtained from FACS. A scattergram between any two of them, X and Y, depicts the degree of association between X and Y as a swarm of points-as-individuals. The degree is represented metrically by R, a function of a straight line drawn through the swarm. But if curvilinearity exists, signifying a higher association than that indicated by R, the scattergram will show it visually, and the index of curvilinear correlation, ETA, will be higher than R.

Another interest in the scattergram is that the individuals-as-points are described in 'Euclidean' score-space, a basic User’s frame in O-analysis (see EUCO-analysis). This frame is the orthogonal space in which the procedures of O-analysis TAKE place (see 'Typology'), except that O-analysis deals with the swarm in the orthogonal score-space of all the score dimensions from FACS taken simultaneously.

Method

RSCAT is a general program that can be used to compute all (or a selected set of) the scattergrams between any block of not more than 18 variables. Each scattergram provides not only the plot, but also R, ETA XY, ETA YX, marginal frequencies, percentile ranks, sigma scores, means, sigmas, and other statistical properties. The computed value of the ETAs, however, are subject to strong bias upwards if the number of classes, or 'slices' through the diagram, on the basis of which their values are determined, is large. It is quite necessary, therefore, to estimate the 'unbiased' or 'shrunken' ETA (also called ETA 'corrected for fine categories,' or more simply 'epsilon.')

RSCAT is an adaptation of the fine 704 'scattergram program,' called G2 SCT5, formulated and written by M. Maryuma. That program was converted to the 7090 by project CAP, and relabeled within the BCTRY system somewhat more euphoniously as RSCAT.

RSCAT is now programmed as an integral component of BCTRY. Previously, RSCAT was a program deck called outside the system. In RSCAT the variables correlated can be cluster scores or factor estimates from FACS, or they can be any set of up to 18 of the raw data variables input by DAP2. A new feature permits you to specify any particular scattergrams you wish RSCAT to calculate. The permissible number of subjects can also be greatly expanded. The actual scores that are plotted in the scattergrams are now listed for reference. A new feature is the spotter provided with each scattergram which identifies by ordinal number the specific location of each subject in the scattergram. Also, in each scatter, regression (predicted) scores are now calculated, so you can quickly plot in both the linear and the curvilinear regression lines. Standard errors of estimate are also calculated, both for the linear and the curvilinear cases, the latter being corrected for bias. Most of the statistical constants that were calculated by RSCAT are still produced by RSCAT. Another new feature of each scattergram is the calculation of both correlation ratios corrected for bias (i.e., the values of epsilon). You also now have much more flexibility in the choice of the 'slices' on the basis of which each ETA is computed.

Note: if you wish to use RSCAT independently of other BCTRY components, see data under control card A, below.

Use

Card input.

(1) Executive control card. Punch '/RSCAT+' in cols. 1-7. ****

****

(2) Component control cards. Only one component control card, ****
Card A, is now necessary, unless you wish to specify particular ****
scattergrams that are to be computed, in which case you specify ****
these on control cards B. ****

****

Card A. Parameters and controls. ****

****

****

COLUMNS CONTENTS ****

****

1-4 SELECT, the choice of scattergrams to be ****
computed. ****

```
          '1' or blank.  Compute all the scattergrams       ****
          between the input variables, i.e., (TV(TV-1))/2   ****
          scattergrams, where TV is the number of           ****
          variables not greater than 18 (standard).         ****
                                                            ****
          '2'.  Compute the scattergrams between all the    ****
          input variables and any one of these variables    ****
          taken to be a common criterion variable, i.e.,    ****
          TV-1 scattergrams.  Then designate this common    ****
          variable under COMMON, cols. 17-20.               ****
                                                            ****
          '3'.  Compute only the particular scattergrams    ****
          specified on Cards B.                             ****
                                                            ****
5-8       DATA, the source of the scores that are to be     ****
          correlated.  See tape input below for the tape    ****
          requirements.                                     ****
                                                            ****
          '1' or blank.  The variables are cluster or       ****
          factor scores stored in the FSCOR  file, derived  ****
          either from FACS or input by GIST (standard).     ****
                                                            ****
          '2'.  The variables are from the score matrix     ****
          input to DAP2.  These raw data variables can be   ****
          opted, however, only if the number of variables   ****
          is 18 or less.  Particular scattergrams           ****
          may be called for under SELECT by option '3'.     ****
                                                            ****
9-12      SLICE, the number of slices or categories used    ****
          to calculate the correlation ratios.              ****
                                                            ****
          '0' or blank.  The number of slices is 12 if NS   ****
          is not greater than 200, but it is 17 if NS is    ****
          greater than 200 (standard).                      ****
                                                            ****
          XX.  Punch the number of slices you want, where   ****
          the number is an integer from 1 to 51.            ****
                                                            ****
13-16     SPOT, an option to suppress the spotters.         ****
                                                            ****
          '0' or blank.  Compute and print the spotters     ****
          (standard).                                       ****
                                                            ****
          '1'.  Suppress the spotters.                      ****
                                                            ****
17-20     COMMON, the designation of the common variable    ****
          for which the TV-1 scattergrams are desired       ****
          under SELECT.                                     ****
                                                            ****
          XX.  Punch as XX the ordinal position of the      ****
          common variable opted by SELECT, cols. 1-4,       ****
          option '2'.  This number is its ordinal position  ****
          among the input variables.                        ****
```

21-24 NOLIST, suppressing the listing of the scores from which the scattergrams are calculated. Note: in batched runs (see below), when the same lists of scores are utilized in different passes of RSCAT, redundant printing of listing can be suppressed by this control.

'0' or blank. Print the listing of the score matrix (standard).

'1'. Suppress the listing.

Caution: when NS is very large, say above 2000, it may be wise to repress the listing to avoid excessive output.

Note to users who wish to use RSCAT independently of other BC TRY components: if you opt '1' under data, you will input your raw data in the FSCOR1 file in decimal card format. You have four columns to the left of the decimal for integers, and you are limited to 15 variables (or less if NS is very large - see the restrictions at the end of this Use section). But you have available all three options under SELECT, above. If you wish special names, remember to punch ONAMS in columns 73-78. You must also input an IDFILE by GIST. Variable format is available for the FSCOR1 file. If you opt '2' under data, you will input the raw scores via DAP2. You have more freedom of input and you can input up to 90 variables, but if you have more than 18 variables you can call only specific scattergrams under option '3' of SELECT.

Card B. Designation of the desired scattergrams (necessary if '3' is punched under SELECT, cols. 1-4).

In free field format, punch the list of desired scattergrams as a vector, with the designated scattergrams separated by commas. The two variables of any particular scattergram are designated by their pair of ordinal positions separated by a punched '+' sign. When the variables are FSCOR1 dimensions from FACS, their ordinal positions refer to their cluster or factor numbers, which can range from 1 to 15. But when the variables are input by DAP their ordinal positions refer to their ordinal numbers in DAP2, which can range from 1 to 90. For example, a vector of three scattergrams among 90 variables input by DAP2 could have this free field vector form, beginning in column 1: '1+3,101+119,19+90.', the plusses in this case not referring to blank columns but to actual key-punched plus signs.

Restrictions of Cards B: only 24 scattergrams can be specified, and these scattergrams can be among no more than 18 different variables (see Restrictions at the end of the Use section).

Batched runs. By batching runs you can escape from the restrictions imposed during one pass of RSCAT. With a given FSCOR1 file available, or from a list of variables input by DAP, you can call for as many passes of RSCAT as you please in order to get the particular scattergrams you wish.

Tape input. Under data, cols. 1-4 of Card A, if you opt '1' then you need both the FSCOR1 file from FACS (or input by GIST) and IDFILE, and if you opt '2' you need the DST written by DAP2, it being most desirable also to have the VNAM1 file also available for the clear identification of the variables being correlated. For both options, it is also desirable to have the ONAMS1 file available for the clear identification of the subjects in the score listings. RSCAT generates both files if they are missing.

Printed output. The printout includes the following items:

The options chosen by the user.

The score matrix from which the scattergrams are computed.

The means and standard deviations of the variables.

By variable, the names of the subjects that lie beyond plus or minus 2.55 standard deviations (they are plotted at the 2.55 score in the scatters).

A prediction table of Y from X for each scatter, showing for each X slice:
mean X score;
predicted Y score (linear and curvilinear);
standard error of estimate (linear and curvilinear) of Y from X;
frequency, N, in the slice; and
the A and B constant of the prediction equation of Y from X.

The comparable prediction table of X from Y for the scatter.

Each X,Y scattergram, which gives the following items:
the X and Y scale scores at .5 SD's in the positive and negative directions;
marginal percentile ranks, for every other row and column;
marginal frequencies for every row and column; and
the frequency in every cell of the scattergram.

For each scattergram a spotter listing the ordinal numbers of the subjects in each row reading from left to right, and enclosing those in a given cell by parentheses.

Restrictions. The total number of variables, TV, that RSCAT can deal with in one pass is 18. The number of subjects, NS, cannot exceed the value in the equation NS = 10,000 divided by TV. Thus, when TV is the maximum of 18, NS cannot exceed 555. If your NS exceeds 555 you must decrease TV, over which you have control in select, cols. 1-4 on Card A, especially by option '3' there. Using this option, you could, for example, in the limit decrease TV to 2 variables, thus providing for only one scattergram that could include up to 5,000 subjects.

The number of rows (and columns) in the scattergram is 51, giving a total possible number of cells of 2,601. The maximal frequency of cases in a given cell is bounded by a two digit number, namely 99. So also is any marginal frequency limited to a two digit maximal number of 99. However, if the count in either cells or margins exceeds 99, the following convention accommodates higher frequencies: if the first two digits of a three digit count is the number 10, then the letter A replaces 10 in the cell printing, e.g., A9 stands for 109. If the first two digits of a three digit count is 11, then B replaces the 11, i.e., B9 stands for 119. The maximal frequency in a cell is therefore Z9, standing for the integer 359. For frequencies greater than 359, a double asterisk is printed. You can always find out the exact frequency in such a cell by going to the spotter and counting the number of subjects that lie in the cell; they are enclosed in parentheses. Frequencies above 359 in a marginal total can be found by counting the numbers in the cells that compose the margin.

The largest integer of a DAP2 variable should not exceed 999 because its scale on a scatter takes only 3 digits before the decimal point. If the highest integer should exceed 999, rescale the variable by the INFORMAT statement in DAP2. Do not use RSCAT with missing data, because RSCAT assigns the number zero to missing data.

Substituting a conventional scale for the decimal scale on the scatter. The X (or Y) scale on a scatter is printed in units having decimal points that make difficult locating the exact position of scores lying in between. You need a conventional scale with points expressed in integers or terminating in values like 0,5 or, say, in even numbers. Call the conventional scale an X(C) Scale. You can quickly locate such a scale on the printed Z-scale which is segmented in 10ths of SD by hyphens printed through the lower part of the scatter at a Y of -2SD.

Proceed as follows: decide first on the type of conventional X(C) scale you wish. Then compute the Z-score values of the first two X(C) points just above the mean and write them on the hyphenated Z-scale. Then compute the interval, I, between these two Z-values. For the first X(C) value above the mean, set its Z-score in a desk calculator and then add 1 cumulatively to it. Mark off the resulting series on the Z-scale, writing down their appropriate X(C) values. On a paper edge, copy this scale and use it as a guide to mark off the X(C) scale below the mean. The X(C) scale applies, of course, to all scatters of the particular variable being scaled.

Typology

Introduction

Are all individuals in a multidimensional study completely unique? Stated specifically, do their profiles of scores on the specific variables or on the cluster scores derived from them by V-analysis form a random, undifferentiated collection of patterns? Or do their profiles fall into distinctive groups or types signifying, perhaps, that particular patterns of behavior are generated with high frequency by genetic, social, or other forces?

The procedures of O-analysis are designed to discover the substructural composition of the population-sample under study. If individuals are unique, O-analysis will reveal the fact; if a distinctive typology exists, that fact will be discovered; if some individuals are unique and others form a typology, this complex configuration will become evident.

The objective of O-analysis is (as in V-analysis) essentially reductional: to classify the large number NS of individuals (objects) into a smaller number M of salient subgroups, called O-clusters. These classes are distinguished from each other by their patterns of scores on the attributes under study. Individuals of the same O-type have approximately the same profile pattern, hence are collinear; they have scores that follow the same profile line.

When they are classified on the basis of the general attributes measured by the cluster composite scores from FACS the operations are called general O-analysis. When the typology is executed on their profiles of scores on the N specific attributes, the procedures are termed specific O-analysis, commonly referred to in orthodox factor analysis as 'inverse factor analysis'.

General O-analysis is preferred to specific because the analyst is more likely to grasp the meaning of profiles of different O-types than he would if he dealt with profiles on all NV variables. The analyst who prefers, however, to perform specific O-analysis can use BCTRY components to do so. In the User's descriptions below, most procedures utilize the programs previously described in V-analysis, namely, the reduction by factoring of the large number of entities (variables vs. individuals) to minimally-sufficient oblique subsets (V-clusters vs. O-types). In O-analysis, however, the analyst does not usually settle for the minimally-sufficient number of O-types only. For his final descriptive typology, he expands the number of types from K to a larger number, M, of O-clusters. He does so in order to map out rather thoroughly the substructural composition of the population-sample under study.

The subordinate objectives of O-analysis and the procedures that implement them are as follows:

EUCO-analysis: objectively describing the structure of the relationships between profile patterns of the NS individuals by dimensional analysis - A given individual's pattern of scores from FACS can be most efficiently represented as a point in a Euclidean space of these scores. Program EUCO, described below, so represents the NS individuals' scores. It computes the Euclidean distances, or D values, between them in pairs. In EUCO-analysis, this distance matrix is then converted to a matrix of correlations (COR2). Each D is replaced by the inter-columnar correlation, R(D,D). A full-cycle NC-analysis is then performed on this matrix, the results of which represent the NS individuals in a reduced set of K dimensions. In the structure analysis by CSA, the oblique object-cluster dimensions are defined by the K subsets of the most nearly independent or 'primary' O-types, and SPAN gives a complete layout of the relations among the score-profiles of the NS individuals. In the User's descriptions of EUCO-analysis and of EUFIT, procedures are given for the handling of problems with a very large NS.

OMARK in EUCO-analysis: selecting a set of M core O-types that are representative of the complete structure within a context of 'marker individuals.' - From the configuration of the relations among the NS individuals as depicted on the SPAN spheres of EUCO-analysis, the analyst chooses a set of core O-types, each as homogeneous as possible. These include, in addition to the K subsets of oblique dimension-definers, additional 'dependent' subsets that representatively sample the whole structured population-sample. Heavy concentrations of individuals at some points in the configuration signify highly 'compatible' or conjunctive score-profiles; lacunae reveal missing, or 'incompatible' disjunctive types. The procedures called OMARK below permit the user to introduce into the analysis 'marker' types of individuals, i.e., artificial model 'individuals' with score-patterns of theoretical interest, or rational types of real individuals of special interest on theoretical or social grounds. The marker types may aid the analyst in determining the final set of M core O-types representative of the whole configuration.

EUFIT - By the procedures called EUFIT all NS individuals are allocated to the core O-types that best represent them. The groupings of the NS individual resulting from this allocation process are termed the final O-clusters. Some individuals, however, have such aberrant profiles that they are set aside as unique individuals.

OSTAT: objectively describing the final set of O-clusters - The level of the profile of each O-cluster is sufficiently described by its pattern of mean Z-scores. The homogeneity of an O-cluster is an index of the degree of similarity of the scores of the individuals that compose the O-cluster. A User's description of the procedures for calculating these statistics on each O-type is given below in OSTAT.

PREDICT: predicting 'outside' attributes of the O-clusters - The degree to which the M O-clusters differ from each other in domains of other psychological and biological attributes than those measured in the V-analysis can also be objectively assessed. The M O-clusters constitute a discontinous series that can be conceptualized as a 'predictor,' categorical variable; an 'outside' attribute may be thought of as a continous 'criterion-variable.' Then for each O-cluster, its score-level and its homogeneity in the predicted criterion are, respectively, its predicted score and its error of estimate on the criterion. The over-all non-linear correlation, ETA (see OSTAT) analogous to R, between the O-type categorical variable and the criterion is calculated from the M O-cluster homogeneities. The procedures called PREDICT below compute the elements of these statistics.

OCOMP: comparative O-analyses. - The analyst may have two or more groups that have been measured on the same NV variables. On these groups he may have performed V-analyses and O-analyses. In addition to comparing the oblique cluster dimensions of the groups by COMP (q.v.), he may wish directly to compare the O-cluster structure of the different groups. The procedures for doing so are given in the User's description of OCOMP. The comparison of the O-cluster structure also reveals directly and in considerable detail the influence of univariate and multivariate selection in the different population-samples compared.

Note: for an illustration of typology by BIGNV procedures, see Tryon, Stein, and Chu (1965). For a simple algebraic statement of the theory of O-analysis, see Tryon (1964a).

User's description of the selection of core O-types by dimensional analysis: EUCO-analysis

Introduction

To attempt to cluster individuals on the basis of their profiles represented by line-graphs over K cluster scores would be a forbidding task. EUCO-analysis takes a radically different approach. It expresses each individual's score-profile as a single point in a score-space: each score-axis is set orthogonally to the other score-axes and each individual is then located in this joint-score space as a point determined by his different cluster scores taken as coordinates on these axes. The result is that all NS individuals then describe a configuration of objects in 'Euclidean space.' Two individuals together as points in this space have identical score profiles; the farther apart, the more their profiles are different; at opposite 'corners,' their profiles are mirror-images.

The main object of EUCO-analysis is to locate core O-types of individuals at places in the configuration where there are condensations of points. These are loci around which, by the later procedures of EUFIT, O-clusters are collected comprising all NS individuals. EUCO-analysis locates core O-types by the standard procedures of dimensional analysis used in V-analysis. The first component employed, however, is EUCO, which calculates the distance, D, between each pair of individuals. Since the resulting EUCO-matrix does not, however, conveniently describe the configuration, it is converted to a matrix of correlations between individuals. This R-matrix is then subjected to a full-cycle V-analysis that finally provides a description of the configuration of the individuals as a swarm of points on SPAN diagrams. This presentation permits the analyst to select core O-types. The selection process is aided by OMARK, a procedure for introducing 'markers' into the configuration. The User's description of OMARK is really a part of EUCO-analysis; the two procedures should be handled as a unit.

A simpler, non-dimensional method of O-clustering is also given in EUFIT (q.v.).

Method

The general logic and formulation of the methods will be treated generally under the four steps below.

(1) Determining the scores on which to compare individual objects. Since EUCO computes distances between objects in the space of K scores calculated in FACS, the analyst must decide before running FACS what kind of scores he wishes FACS to compute for EUCO. These can be the oblique simple sum or regression scores, or the orthogonal regression scores (recall they are not really independent though usually fairly so). However, since scores on the independent dimensions are unlikely to be meaningful (see SPAN) whereas oblique scores usually are, the oblique scores should be more generally preferred, especially the simple sum cluster scores.

(2) Calculating the Euclidean distance between individuals: The D-matrix. This computation is the function of the EUCO program proper. Since the score axes in which the individual occupies a point-locus are set orthogonal to each other, then the distance between any two objects I and J is simply the square root of the sum of the squared differences between the scores of I and J. For example, if there were only two scores (K = 2), then D(I,J) Is the length of the hypotenuse that separates them. This formulation holds, of course, however large K may be. Note that when dimension-scores are weighted differently in EUCO by the analyst (an option in FACS) this means that the individuals are stretched out more on those score-axes with larger standard deviations.

(3) Transforming the D-matrix to an R-matrix: the R(D,D) matrix. Recall that the EUCO matrix is a data matrix on which O-clustering is to be performed by a full-cycle V-analysis. But a key cluster factoring cannot be performed on a symmetric matrix unless it is in a metric of R. Hence the EUCO matrix is transformed to a R-matrix by intercorrelating the columns of individuals in the EUCO matrix, using COR2, and the V-analysis is executed on this symmetric R(D,D) matrix between individuals. The problem then arises as to whether score-profile differences between individuals, as measured by their D's, are also exactly reflected in their R(D,D) values. Methodological studies have shown that for a given individual his column of D values in the EUCO matrix is monotonically related to his corresponding R(D,D) values (the analyst can, of course, check this in OMARK (q.v.).

(4) Locating the general O-types by factoring and structure analysis: the core O-types. A full-cycle V-analysis (usually NC) is run off the R(D,D) matrix. The number of dimensions, K, tells the salient number required to account for all the relationships between the distances between the individuals. CSA gives the critical contrasting types of defining oblique core O-types, and the oblique factor coefficients tell how much each individual's score-profile matches that of each defining O-type. And SPAN gives the appropriate map of the resemblances and differences between the objects - the O-cluster configuration. The analyst will study CSA and SPAN and from the configuration, select M O-types as the final set of most distinctive types of individuals. Let us call the final set of selected clusters, the core O-types. The procedures usually followed in selecting these core O-types are as follows:

A. Final selection of dimension-definers. From the printouts of CSA and the SPAN spheres provided by step (4) above, the analyst checks to see whether the empirical NC factoring procedure has in fact selected the most-nearly independent sets of objects as the defining objects of the K dimensions. If not, he makes a revised selection of pivotal dimension-definers. In this process, if he has trouble perceiving the configuration of object-points on the SPAN spheres, he will utilize only the oblique factor coefficients under each oblique cluster given in the 'unifactor' listing of CLUR in CSA. From the list, he will select only those with high oblique coefficients as the revised dimension-definers. Some objects that had been selected as definers by the NC factoring procedure may turn out not to have the highest coefficients, being now replaced by others with higher values that were not selected during factoring.

B. Preset O-analysis. If the subsets of dimension-definers are revised, the analyst may wish to rerun the full-cycle NC-analysis in order to secure improved SPAN diagrams. He does so by presetting the dimension-definers of the K dimensions by the revised subsets of definers.

C. Final selection of core types. From the final SPAN diagrams he will make a final selection of the most contrasting collinear core O-types among the NS individuals. In studying the configuration of objects on each sphere, he will orient mainly to objects whose communalities are more than 90 percent accounted for, ignoring those with values less than 90 percent. The finally selected core types should represent all parts of the configuration, if possible. The number of them, M, can be as large as you please, but if M is kept to 15 or less, the analyst can then run a preset CSA with the M core O-types preset as M oblique clusters; the results will givE a metric description of their interrelations, and when NS is not greater than 90, the relations of all NS objects to the M core types.

D. Empirical vs. rational core O-types. In choosing the final core types the analyst will naturally orient to loci in the configuration where there are heavy concentrations of individuals. These are empirical or 'natural' clusterings, i.e., types of individuals whose scores on the different general dimensions have special compatibility due to some special bio-social homeostasis among the attributes. Lacunae signify types of profiles that refer to biosocially incompatible, disjunctive patterns of scores on the attributes.

But sharp condensations and lacunae may not appear. Even if they do, the analyst may wish to include as referent types certain 'marker individuals.' Two kinds of marker objects come to mind. The first are rational types such as males vs. females, old vs. young, the characteristic pattern of another group measured on the same NV variables, and so on. The second kind of markers are artificial model 'individuals' such as one with high-scores-on-all-attributes, or lows-on-all, or types that reveal special attainment or deficits in each general attribute. Since no actual individuals in the configuration may actually represent such rational types, the analyst will make up 'artificial individuals' with FACS scores of the desired types. These can be input with the actual NS individuals in the EUCO-analysis, and their loci in the configuration be observed in relation to real individuals. Special procedures for the use of marker 'individuals' are given in OMARK.

The selection of core types is basically arbitrary. Indeed, nothing prevents the analyst from selecting alternative sets of core types for study. For a detailed listing of steps in EUCO-analysis see Use.

Card input. Executive control card. This is the only control card required by EUCO. Punch '/EUCO++' in columns 1-7.

This program reads the following information from the intermediate storage tape:

IDFILE	Identification.
MEANS2	Factor score means.
STDEV2	Factor score standard deviations.
WEIGH1	Factor score weights.
FSCOR1	Factor scores.

The program prints all information input via IST in addition to a matrix of Euclidean distances.

EUCO writes the following information on IST:

MEANS1	Mean distances of each object.
STDEV1	Standard deviations of the distances of each object.
VNAMS1	Object names (the word object repeated NS times).

EUCO writes the Euclidean distance matrix on DST exactly as if it were a raw data matrix. Thus the matrix can be read by COR2, COR3, or FACS.

Since through EUCO subjects become variables, the number of subjects may not exceed 90. All other BCTRY restrictions apply as well.

Detailed steps of a EUCO-analysis

Note: the starred (*) steps in the following are not necessary unless NS is in excess of 90. When NS exceeds 90 you should also have SPAN2 available to you. See step (12).

Computing the composite scores of individuals on which they are to be compared.

(1) FACS - Having finally decided on defining variables of the KV clusters, run FACS (q.v.) to get simple sum cluster scores on them.

(2) GIST -
(A) Call for the output file package of FSCOR1 cards.
(B) * If NS exceeds 90, input the case numbers of the NS individual in the VNAMS1 file, thus getting for later reference a compact GIST printout of these numbers by ordinal position in the FSCOR1 output of FACS. Repeat this step in successive lots of 90 cases.
Note: this step is unnecessary when ONAMS file is available.

(3) For later purposes of sorting the cards by O-type key-punch the VNAMS on the FSCOR1 cards in cols. 67-70.

(4) GIST - * If NS exceeds 90, sort out a random 90 FSCOR1 cards, and input them in the FSCOR1 file. Input also IDFILE, MEANS2, STDEV2 and WEIGHT1 with the parameters of each file corrected for the number of subjects used (90 here).

Calculating the Euclidean distances between individuals: the D-matrix.

(5) EUCO - Call EUCO in order to get the D-matrix on this first sample of 90 cases.

(6) GIST - * Input in the VNAMS1 file the case numbers of these 90 cases. The GIST-VNAMS1 printout gives a compact list of case numbers for User's, and it also labels the rows and columns of the correlation matrix by them (next step).
Note: this step is unnecessary when ONAMS file is available.

Transforming the D-matrix to a correlation matrix: the R(D,D) matrix.

(7) COR2 - This gives the R(D,D) matrix, for factoring.

(8) DVP - Call DVP50 to put highest R(D,D) In the diagonals.

Selecting the core O-types by factoring and structure analysis.

(9) GIVE - Since the NC2 analysis (in step 10) may need to be repeated, it can restart from this GIVE deck.

(10) NC2 - This subset factoring gives K(O) dimensions, with their defining-variables (here, individuals).

(11) CSA - This component gives the statistical cluster structure of the objects.

(12) SPAN - This component gives the printout of spheres, showing in pictorial form the configuration of the individuals corresponding to their D-matrix.
Caution: this step requires SPAN2, with rigid rotation on dimension-definers only.

(13) Revising the factoring and the structure analysis -
(A) Label the defining-objects of the KO dimensions in The SPAN spheres, and also other objects with very high oblique factor coefficients on them (see these values in CLUR in CSA).
(B) If some dimensions have poor defining subsets, revise them, and preset the dimensions of a new NC2 run by the revised definers. Now rerun NC2, CSA, and SPAN, starting with a TAKE on GIVE of step 9 above. In this revised run, in order to reduce the number of SPAN spheres to be dealt with later, cut down on the number of dimensions providing the later dimensions of the first NC2 run appear to be insubstantial. Some objects may, under the reduced dimensionality, give a few high residuals but these can be handled later in OMARK (q.v.).

(14) * Replication of EUCO-analysis when NS is large - If NS is greater than 90, select a second set of objects for EUCO-analysis in order to get a replicated picture of the cluster structure on more than 90 cases. These will be run on a preset analysis, using the revised set of dimension-definers as in step 13b. The steps are as follows:

(A) To the deck of FSCOR1 cards of the revised dimension-definers, add the FSCOR1 cards of a second random selection of additional objects.

(B) On this second group of 90 objects, repeat steps 4 to 12 above, but in step 10 preset the NC2 analysis by the revised dimension-definers. The results in NC2, CSA, and SPAN will, on the two runs of 90 objects, be comparable. There should be a common set of SPAN spheres in the two runs.

Note: see caution of step (12), which assures comparability.

(15) Selection of the final set of M core O-types -

(A) Label all objects in each SPAN sphere in both runs whose per cent of communality accounted for is .90 and above (marked by 'V' in the table of coordinates). Ignore those marked 'V' whose per cent values are .80-.89; their loci in the configuration are usually poorly represented. Cues on labelling: ignore the Z coordinate of each object and locate its position by its X (horizontal) and Y (vertical) coordinates. If Z, however, is negative, the point is on the under hemisphere. Objects reflected (printed 'R') should be crossed out, and replotted by their coordinates as printed in the table of coordinates (watch for negative Z).

(B) * For each common sphere on the two runs, condense the configuration onto one sphere. To do so, first draw the spherical triangle on each sphere, and then take off the configuration of one of them on tracing paper. Then overlay the tracing on the second sphere, superimposing the triangles as closely as possible, and take off the points of the second sphere. Result: the configuration of all objects in the common space is now on one piece of tracing paper. There will be as many composite configurations as common spheres.

(C) Select the tightest set of M core O-types possible, cross-referencing the configurations. A systematic procedure is as follows:

(I) Encircle the definers of each of the dimensions, labeling them as core type 0(1), 0(2),...,0(K0).

(II) Encircle clusters of individuals in or near single planes or in extensions of planes, labeling them as core types 0(1,2),0(1,3),0(2,3)..., etc. If more than one are in or near the same plane, label them as, say, 0(1,2,A), 0(1,2,B)....

(III) Encircle other clusters in all three dimensions, labeling them as core types 0(1,2,3),0(1,2,4),0(1,3,4):

(IV) From the deck of FSCOR1 cards, sort out the M core O-types into M subdecks.

(V) Prepare DAP control cards for each subdeck (as described in OSTAT), list them in a listing machine, and on the printout sheet check each O-type to see that the cluster scores are properly consistent. Most analysts will also want to draw the profiles of the objects in each core O-type as a final check.

Steps after EUCO-analysis

Though the M core O-types will have been located by the above steps 1 to 15, most analysts will want to perform the additional procedures of O-analysis, certainly of OMARK, EUFIT and OSTAT, and possibly of PREDICT and OCOMP. A detailed User's description of each is given in later sections but in order that the whole design of the analysis is apparent, a brief summary follows:

OMARK - For interpretation and other purposes, a EUCO-analysis is performed on the core O-types, marker objects (model and rational) and objects with non-trivial residuals in the initial EUCO-analyses. Additional core O-types may emerge from this analysis.

EUFIT - All NS individuals, however large be NS, are run on EUCO with the final set of core O-types. As many runs are made, each with the common core types, as is necessary to accommodate all NS individuals to that core O-type with which it has its smallest D value. The resulting enlarged clusters are called the final O-clusters of the study.

OSTAT - The mean profile of each O-cluster and its homogeneity are computed, using DAP.

PREDICT - The degree to which scores on an 'outside' variable are predicted from the O-clusters is calculated, using OSTAT procedures.

OCOMP - If the analyst has performed O-analyses on other groups scored on the same KV clusters, their core O-types are projected into a single master EUCO-analysis, and thus directly compared.

User's description of markers in EUCO-analysis: OMARK

Introduction

The actual values of the cluster scores of the core O-types selected in the prior EUCO-analysis are not immediately apparent in the configuration of objects projected on the SPAN spheres of EUCO-analysis. OMARK is a set of procedures that projects into the configuration model marker 'individuals' in relation to which the analyst can see at a glance the following objective facts, the actual scores of the real individuals that comprise the core O-types as well as the regions of scores in which no individuals exist in the configuration, and the scores of individuals of special interest on a priori rational grounds. Furthermore, OMARK provides the means of locating additional residual core O-types not evident in the prior EUCO-analysis when the K(O) dimensions factored in that analysis may have left some objects with too high residuals.

Method

There are four main methodological steps in OMARK analysis:

A. Selecting the markers and computing cluster scores on them. These include:

(1) The M core O-types derived from the prior EUCO-analysis.

(2) Model marker O-types, i.e., hypothetical individuals whose distinctive cluster scores provide User's points in the configuration by means of which the scores of real individuals become immediately apparent.

(3) Rational markers, i.e., mean cluster scores of (A) sub-groups within the group of a priori interest to the analyst, or (B) of groups outside the study on whom scores on the NV variables may be available.

B. Running a complete EUCO-analysis of the marker individuals that have been selected in A, above, including also the few real individuals with high residuals in step (13,B) in the prior EUCO-analysis. In the factoring procedure the same set of K(O) core dimension definers are used as in the prior EUCO-analysis. The result is that the SPAN spheres of OMARK are common to those of the prior analyses, hence the markers can be projected directly onto the real configuration on the spheres in the prior analysis.

C. Factoring the residual objects insufficiently described in the prior analysis. Those objects previously with too high residuals will still be residuals in the present EUCO-analysis of markers. To discover additional core-types among them, the residuals of markers and of residual objects are set up as a correlation matrix by the component FAST and submitted to NC factoring. From CSA and SPAN additional core O-types that require additional dimensionality to produce final trivial residuals are selected.

D. Projecting the model markers as gradients in the spherical configurations. The model O's are identified in the SPAN diagrams. Those that form cluster score axes and other meaningful scalings of objects are connected by gradient lines in the configuration. The score patterns of the real individuals can thus be read off in relation to these score gradients.

Use

The detailed steps of OMARK analysis are organized under the above four methodological steps:

A. Selecting the markers and computing cluster scores (FSCOR1 cards) on them.

For each marker, an FSCOR1 card is required for input into the EUCO-analysis of OMARK, as follows:

The M core O-types. Two kinds of core O-types are differentiated in OMARK. (A) Dimension-definers of the K(O) dimensions in the prior EUCO-analysis. - The individual FSCOR1 cards of the definer-objects oF O(1), O(2)..., O(K(3)) are collected into K(O) subdecks for input into EUCO-analysis in order to be assured that the dimensions of the of the SPAN diagrams will correspond to those in the prior analysis. (B) The remaining core O-types. - For each such type, the mean cluster scores of the individuals that comprise it are calculated from its subdeck of FSCOR1 cards by the following BCTRY components (the computations of all the types are executed in one batched computer run).

DAP - This gives, for each type, the mean cluster scores (MEANS1) and the SD's of each cluster score (STDEV1, used later in OSTAT, q.v.).

GIST - Call for output of cards containing MEANS1 and STDEV1. The MEANS1 cards are the FSCOR1 cards for input into marker EUCO-analysis. Presently they must be key-punched in format (9F8.4).

Model marker O-types. The analyst can choose any kinds of score patterns he wishes as model O's. But there are several types of special utility.

(A) Special model O's that define a gradient of special attainment or deficiency in a given cluster dimension. A given cluster dimension can be expressed as an axis-gradient in a SPAN configuration by a selected set of model O's that have a mean Z-score of 50 on all other cluster dimensions but differ from each other in their Z-scores on the given dimension. A good set is one of 8 'individuals' whose scores on the given dimension are separated by .5 SD, namely 70,65,60,55,45,40,35,30. Note that 50 is missing. It is included in (B) below.

(B) General model O's that represent a gradient of equal Z-scores across all cluster dimensions. A good set consists of 9 such 'individuals' separated by .5 SD, i.e., one with all Z-scores at 70, one all at 65, and so on for 55, 50, 45, 40, 35, 30. Note that the model 'individual', all of whose Z-scores are 50, represents the missing model individuals in (A) above.

In general, the two types of markers above sample the entire joint raw cluster score matrix. In (A) they are equally spaced on the score scale of each cluster dimension projected through the origin of 50 on all other dimensions. In (B), the markers are equally-spaced individuals along the diagonal through the joint score matrix. The analyst may introduce other markers for which there is evidence of a crowding of real individuals. For example, if two cluster scores are highly correlated as shown in V-analysis by CSA and SPAN (and also in the tabled R's of FACS) one knows that individuals must be concentrated along the regression line of those two cluster score variables. Hence, one could make up markers equally spaced along this regression line, I. E; 8 individuals with Z-scores of 50 in all other cluster scores but jointly having 70,65,60,55,45,40,35,30 on the two clusters. A careful study of CSA and SPAN in V-analysis may suggest other marker-gradients for input into OMARK.

Rational markers.

(A) Within-group rational markers. Just as the core O-types are empirically-derived subgroups of the sample of NS individuals, so also may there be rationally-designated subgroups whose mean cluster-score pattern may be set into the SPAN configuration, e.g. males vs. females, or any other known subclasses of individuals of a priori interest to the analyst on theoretical or empirical grounds. The FSCOR1 cards of each such subclass are produced in precisely the same way as that described for core O-types. The subdeck of individual FSCOR1 cards of the subclass is sorted out, and put through the DAP-GIST sequence described above.

(B) 'Outside-group' rational markers. If any other 'outside' sample of individuals has scores on the same NV variables as the group under study, then its FSCOR1 cards can be computed by FACS and output by GIST. These cards may then be input into DAP, whose MEANS1 card becomes the FSCOR1 marker card for input into EUCO.

B. Running complete EUCO-analysis on markers and residual individuals.

The above preparations for OMARK EUCO-analysis have resulted in the following five classes of FSCOR1 cards:

Core O-types consisting of the deck of K(O) subsets of dimension-definers.

Single cards representing the remainder of the M core O-types.

Single cards of the model-marks.

Single cards representing the rational markers.

The deck of the residual objects insufficiently described in the prior EUCO-analysis (step (13,B)).

These cards, merged in a single deck, are the input for a complete EUCO-analysis beginning with step (3) and ending with step (12), ignoring the starred steps. The last step must be followed by a GIVE card, necessary to the following residual factoring procedure.

C. Residual factoring in OMARK analysis.

In the above EUCO-analysis the real residual individuals brought into the analysis will have too high residuals. Further dimensions are necessary to account for their correlations. How many? The answer appears to lie in the relation of the dimensionality, K(V), in V-analysis to the dimensionality, K(O), in O-analysis. It appears from a number of studies that one more dimension is needed in O-analysis than in V-analysis, i.e., K(O) = K(V)+1. Therefore, if we let K'(O) be the dimensionality in the prior EUCO-analysis, and X be that required in residual factoring, then X = (K(V) - K'(O)) + 1 = dimensionality in residual factoring.

For example, if the dimensionality in the V-analysis of a given problem were large, say, K(V) = 10, and if the analyst cut back the dimensions in EUCO-analysis, say to K'(O) = 5, in order to deal with a reduced set of SPAN diagrams, then the number of dimensions in the residual analysis is X = 6. Observe the general logic of this split up of the dimensions. In the EUCO-analysis on five dimensions he has accounted for the score-patterns of a very large (i.e., a salient) proportion of his NS objects; therefore in the residual analysis on six dimensions he deals only with a very small number of remaining residual objects.

Having now decided on X, the dimensionality, a full cycle NC analysis is performed on the residuals of the EUCO-analysis of the markers, using the following BCTRY components:

```
TAKE - from the GIVE card called at the end of the EUCO-analysis.
FAST - call residual (q.v.).
DVP -  call DVP50.
GIVE
NC2 -  preset to X, the required number of dimensions.
CSA
SPAN
```

From the output, the analyst has only to choose additional core O-types sufficient to account for the residual objects. The procedures are those already described for selecting the core O-types (see EUCO-analysis, step (15)). In the SPAN diagrams the residuals are imbedded in the configuration of model markers. Therefore, he can choose, if he wishes, a model O as one of the remaining core O-types if it seems best to describe such a type.

D. Projecting model markers as gradients in spherical configurations.

In the EUCO-analysis of OMARK, the SPAN diagrams GIVE the configuration of markers in relation to the core O-types on the spheres. In this configuration the analyst draws the gradients through the model markers. In relation to these gradients, he can see at a glance the score patterns of the core O-types. He can also see the relation of the rational markers to the core O-types. He can, if he wishes, transfer the score gradients to the composite configurations drawn in the prior EUCO-analysis and thus see the score patterns of all NS individuals from their relations to the gradients. Similarly, in the SPAN diagrams of the residual analysis, the gradients drawn through the markers there serve to identify the score-patterns of the residual individuals.

User's description of comparative analysis of O-clusters and of multivariate selection in different groups: OCOMP

Introduction

Just as we compare by COMP the dimensions of different groups derived from a V-analysis in each group, we also compare by OCOMP the O-clusters of individuals derived from an O-analysis of each group. The only condition required by OCOMP is that all individuals in all groups be observed on a same common set of NV variables. The final objective in OCOMP is to project all O-clusters into one master EUCO-analysis from which the resemblances and differences between them will be expressed in terms of Euclidean distances (D values) and correlations (R(D,D) values) and will be pictorially represented in one master configuration.

Multivariate selection. The special importance of comparing the cluster structure of the different groups is to assess the different influences of multivariate selection on them. Multivariate selection means, simply, the degree to which the O-clusters that form the different groups are not random selections from the grand 'population' of which they are samples, or, stated positively, the degree to which the O-clusters are biased selections from it, both in mean profiles (of the multi-variables), in homogeneities, and in the correlations among the NV variables that compose the oblique clusters ('factors').

When the objects are individual organisms, multivariate selection is normally expected to work out differently among the groups compared. At the genetic level, the probability is certainly zero that the gene-frequencies of the polygenic systems that determine the NV attributes in different groups are the same. And the social-environmental forces that form the different groups doubtless generate different selections in the attributes under study. It is therefore an exciting enterprise for the analyst both to describe the similarities and differences in O-cluster structure of the different groups and to attempt to explain these variations.

Multivariate selection is a gross embarrassment to the mystique of 'invariant underlying factors' across groups. When O-cluster structure changes from group to group, the belief in invariant factors cannot be supported. Thomson has proved (1951, ch. 19) that changes in multivariate selection from group to group change the patterns of correlations between the variables, hence 'new "factors" emerge, old ones change' (op. cit; p. 303, sect. 6). Since this fact unhappily negates the invariant simple structure belief, Thurstone attempts to demonstrate invariance by appealing to a special case of multiple selective effects which he calls 'successive univariate selection' on different variables (Thurstone, 1947, p. 464ff.). But as Thomson points out, 'multivariate selection is not the same thing as repeated univariate selection' on different variables. It is to be noted that most factorists besides Thurstone are vague in their handling of multivariate selection (Cattell, 1952) or silent on the matter (Guilford, 1954; Harman, 1960).

OCOMP procedures enable the analyst to describe with precision the expected changes in multivariate selection (in the correct Thomson sense) across the comparison groups. Precision is achieved by discriminating the O-cluster configuration of each group in a single master CSA SPAN structure analysis. Note: for an alternative OCOMP procedure, see the end of the Introduction to COMP.

Method and Use

The procedures take as given that there are K groups measured on a common set of NV variables, that the K(V) dimensions of each of the K groups derived from its V-analyses have been compared with each other group by COMP, and that M O-clusters have been selected in each group. The numbers K(V) and M may vary from group to group. Since the methods of OCOMP consist of the linking together of components already described, only the schematic steps are given below.

Selecting a common set of K(V) oblique cluster dimensions. From a study of COMP, select one common set of K(V) dimensions on which all individuals in all K groups will be scored.

Scoring by FACS all individuals in all groups on the common set of dimensions. Before calling DAP and FACS, call each group's GIVE cards. For FACS, designate the definers of the common set of cluster dimensions. Finally, use FACS to yield the output of FSCOR1 cards.

By OSTAT, securing a single FSCOR1 card for each O-cluster in each group. For specific details, see OSTAT Method and Use. Recall that the number of O-clusters cannot exceed 90 in SPAN (and are not likely to). If there is room for them, the analyst should also have included FSCOR1 cards of selected model O-types as dimension gradients in the SPAN configuration (see OMARK).

Performing one master EUCO-analysis on all O-clusters. Each FSCOR1 card corresponding to an O-cluster is now treated as if it were a single individual. The complete EUCO-analysis procedure is followed (q.v.), i.e.: EUCO-GIST-COR2-DVP50-GIVE-NC2-CSA-SPAN. From EUCO the D-values between all O-clusters are produced. COR2 gives the R(D,D) values between them. NC2 can be run empirically but the analyst will probably prefer to preset it by defining the O-cluster dimensions by a set of O-clusters corresponding to the pivotal O-types of one of the groups. SPAN presents the entire configuration. If the solution is not satisfactory, use a TAKE on the GIVE card following DVP50 and preset NC2 for a better CSA-SPAN solution.

User's description of the use of V-analysis programs in specific object clustering: SPEC-O

Introduction

As stated in the purpose section of 'Typology' above (q.v.), specific O-analysis is performed on the basis of the individuals' profiles of scores on all NV variables and not on their composite cluster or factor scores from FACS. In orthodox factor analysis the procedures are called 'inverse,' 'obverse,' or 'inverted' factor analysis, since they involved relating individuals rather than variables. Gradually the term 'Q-analysis' has come to cover these procedures.

Specific O-analysis ignores the redundancy among variables and hence tends to overweight domains that the investigator may have arbitrarily overrepresented in his initial pool of NV variables. General O-analysis, being based on scores on general dimensions, tends to concentrate such redundancies in single dimension scores. The most enlightening typologizing of individuals would be to integrate general and specific O-analysis and thus establish a hierarchy of O-clusters ranging from general down to specific.

This User's description below reviews the possible uses of the BCTRY programs in orthodox Q-analysis but emphasizes what appears to be preferred procedures that may achieve the same ends by employing the EUCO program (q.v.).

Methods and Use

The results of specific O-analysis will be affected by the score space of the NV variables in which score-profiles of individuals are related. There are three such score spaces, depending on how the score matrix is 'centered', as described below. For each such space the treatment below will indicate (A) how BCTRY programs may be used to perform the orthodox form of Q-analysis in traditional factor analysis, and (B) how the EUCO program may be used to achieve the same ends. The proposals here are quite tentative because at the time of this writing (Fall 1963) they have not yet been sufficiently explored.

Types of Q-analysis according to score space

Raw score space: the uncentered score matrix. In this case individuals are correlated with each other in terms of the raw scores on the NV variables. Normally one would perform this type of analysis only when all NV variables were scaled by the same metric having a common rational meaning. For DAP, each card represents a variable and each column on it an individual. Traditional Q-analysis consists simply of a full cycle V-analysis performed on these individuals-as-variables.

If the original data deck is in the usual form for V-analysis, namely, with each card an individual, then the above procedures require the making of a new deck with each Card A variable. But by running the original data deck directly into EUCO, then the making of a new deck becomes unnecessary and each individual becomes represented by a point in a raw score space of NV dimensions. The EUCO program computes the Euclidean distance D between each pair of individuals and O-analysis proceeds in exactly the same fashion as in general O-analysis. (presently the EUCO program is restricted to 15-variable problems since it now accepts only 15 scores on individuals derived from FACS; but this restriction will however in due course be removed).

The single-centered score matrix. When the NV variables are expressed in quite different metrics, then it becomes necessary in Q-analysis to transform the scores on the different variables to a common metric, such as Z-scores. Following such a transformation, a new data deck is made with variables-as-cards and orthodox Q-analysis proceeds in the same way as described above for the uncentered matrix.

But the EUCO program can be applied directly to the original raw score data deck. The O-analysis proceeds on the individuals represented as points in a Euclidean score space in which all variables have equal weight (i.e., equal variance). The way this operation is achieved is to run the raw score data deck into FACS, treat each variable as a 'dimension,' and scale it in standard form (M=50, SD=10). The current FACS program can transform only 15 variables at a time, but it can, of course, treat the NV variables in successive blocks of 15 variables with the scores on each block punched on cards. These would be assembled for input into EUCO (when the restriction of 15 variables in EUCO is removed).

The double-centered score matrix. A third traditional form of Q-analysis involves a second transformation of the scores. In this second transformation, after single-centering as described above, every individual's set of NV scores is itself reduced to a new set having the same mean and SD as all other individuals. On this 'double-centered' matrix, the individuals are then inter-correlated, and a full cycle

V-analysis performed on the resulting R-matrix. Because of the excessive work involved in these double transformations, they can be by-passed in problems where the attributes are rated. In such problems raters or subjects are forced to cast ratings per individual into a common distribution, usually a 'normal' curve. This procedure is known as 'Q-sort' and the pattern of each individual's scores 'ipsative.' But all these statistical and rating circumlocutions are avoided in performing O-analysis from EUCO analysis of the single-centered matrix, described just above. For in the distribution of individuals in SPAN the analyst has before him a complete map of the differences and similarities between patterns of single-centered scores of all individuals. In the SPAN configuration individuals with the same mean and variance of their NV scores fall on the same arc and can be assembled as one 'type,' if the analyst wishes such a typologizing.

Unique O-analysis

Unique O-analysis would typologize individuals on their specific reactions to all the very specific item-responses that form all of the NV variables. If the above procedures were applied to this formidable problem, not only would it stretch the capabilities of the computer but, because of the great specificity of individual responses at the discrete item-situation level, it is doubtful that O-analysis could provide fruitful results, for individuals tend to be quite unique at this level (Tryon, 1962). Most promising instead is to conduct separate multidimensional analyses by BCTRY components within individuals, especially those drawn from contrasting general and specific O-types. In such studies the analyst employs special experimental designs that permit him to discover by V-analysis general attributes within the individual in the behavior domains under investigation. (See the pioneering theses and research papers of D. Bailey (1960) and R. Stephenson (1962).

User's description of collinear comparative cluster and factor analysis: COMP

Introduction

COMP was designed primarily to permit a comparison between the dimensions discovered in one group with those discovered in other groups. In factor analysis, this problem is called 'matching factors.' The data on the basis of which the comparisons are made are the K1 dimensions of group 1 derived from V-analysis, the K2 dimensions of group 2 derived from its V-analysis, and so on to the last group. There is no requirement in COMP that the different groups have the same number of dimensions. Nor, indeed, is there a requirement that the different V-analyses in the separate groups be in practice computed on exactly the same variables. Comparative analysis is, nevertheless, always made on the same variables; there is a deletion control in COMP that permits the analyst to delete from each group those variables that are not common across all groups.

Surprise may be felt that it is possible to compare dimensions (which are, of course, variables) that are observed on quite different groups of individuals. We have all had drummed into us since statistics 1 that two variables can be compared by computing the raw correlation coefficient between them only when we have paired values on them for the same individuals. But it is necessary to conceptualize corelation and correspondence of variables more generally if the purpose and findings of COMP are to be understood. The matter is treated more extensively elsewhere and will be greatly briefed here. In a nutshell the logic is as follows:

The different column dimension-variables to be compared across the groups may be termed comparison-entities - things to be compared. The row NV variables on which the dimension-variables have observations (their factor coefficients) are termed their common referents. In general, there must always be a set of common referents in relation to which entities are observed if we are to compare them. When the observation patterns of two dimensions are identical in relation to common referents then the two entities are the same (i.e., one cannot estabish from the observations that they are different). If, on the other hand, their two patterns of observations relative to the same referents are completely dissimilar, then the two entities are most certainly different. But most important: to the degree to which the observation-patterns of two entities with a common set of referents are the same, then to that same degree are they similar.

This logic sets the stage of COMP analysis, indeed for all the main procedures of multidimensional analysis (e.g., even simple correlation, for the Pearson R between two variables is simply the index of similarity of the observation patterns of raw scores of the two variables in relation to a common pool of NS referent persons - just a special case of the comparative logic given above). Thus, in COMP analysis, different dimensions of several groups are compared in their relations to a common pool of NV referent variables, the observations being their oblique factor coefficients. To the degree that their patterns of observed coefficients on these common referents are the same, they are similar. To pursue this logic, see Tryon (1964a).

The general logical steps of the procedure are given below:

The factor matrix of the K dimensions on its NV variables in each group is first prepared for comparative analysis. This job is executed by the first program, COMP1, which directs the computer to punch cards containing the oblique matrix from the CSA analysis in each group (i.e., RFACT1).

The observation matrix for comparative analysis consists of setting side-by-side the factor matrices of the different groups (i.e., of 'adjoining' them). This is the first step executed by the second program, COMP2, providing a long rectangular matrix.

Taking the columns of dimensions in this rectangular matrix in pairs, COMP2 computes the similarity of each pair of dimensions. The resulting similarity matrix is symmetric, like a correlation matrix. On its principal diagonal are the submatrices that give the indices of similarity of the dimensions within groups; off the diagonal are submatrices of cross-group similarities among dimensions. The latter are what we are after in comparative analysis.

There has been considerable uncertainty about how to measure the similarity of two dimensions when the observations are not scores on individuals but are factor coefficients. What should be the index of similarity? General concurrence is that it should not be by the raw correlation between their two columns of factor coefficients. Some writers have proposed P, the proportionality index, the square of which is our index of mutual collinearity used in key cluster factoring (see Burt, 1948; Tucker, 1951; Wrigley and Neuhaus, 1955). The answer is, however, quite straightforward. Clearly within a group the formula for the index of similarity between two dimensions computed from factor coefficients should give the same value as the correlation between those dimensions ('factors') already computed in CSA. Now, we have those values listed in CSA as 'correlations between domains ("factors").' Therefore, all we need is to discover the formula by which these same values can be ascertained from the columns of oblique factor coefficients also listed in CSA and set up by COMP2 as columns adjoined to those of the other groups.

L, the collinearity index, or cos theta. This problem has now been solved (Tryon, 1964, pp. 109-113). The correlation between two dimensions, the value of which has been computed in CSA from the raw R matrix, turns out to be a simple quadradic function of the P value computed from their oblique factor coefficients. Since the correlation between two dimensions as given in CSA is the cosine of their central angle in SPAN, it is perforce an index of their collinearity, hence the index computed in COMP2 from P is called the collinearity index, or simply cos theta, symbolized as L. The value of L between any two dimensions within a group can be directly compared by the analyst with their inter-domain R ('correlation between factors') given in CSA for that group.

He will discover that they agree closely but not exactly. The reason is that in principle L should be computed between two dimensions only when the referent row variables are the specific definers of the two dimensions listed in CSA, and not all N common referents. But because the operation of deleting out all but the definers of two dimensions when their L is being calculated by COMP2 is rather complicated, we simply use all N common referents. Usually L from COMP2 is a close enough approximation to inter-domain R given in CSA to be satisfactory. However, if the analyst wants more precision, he can use the deletion feature of COMP2 to get more closely corresponding values. (Actually, he will get exact correspondence only if the definers of each dimension are themselves exactly collinear, usually only approximately the case.) In sum, L is a general index of similarity between dimensions both within and across groups; as such, there is no logical restriction that the dimensions be determined on the same group of individuals.

A full cycle V-analysis is finally executed on the similarity or L matrix. This matrix is accepted by CC5, to be followed by CSA and SPAN. SPAN gives a pictorial display showing all the relations among the dimensions within and across groups, and the analyst can thus draw conclusions on the similarities and differences among the dimensions in the various groups.

Though COMP was designed for comparing dimensions from V-analysis, its use is quite general and can equally well serve in O-analysis, especially when NS is large. The problem reduces to locating by regular O-analysis procedures a small group of distinctive O-types as the common referents of the O-clusters discovered in different samples of, say, 100 individuals. The obvious, appropriate, common referents would be, say, 20 model markers (see OMARK) that span the entire joint cluster score space. For the sets of O-clusters discovered in different groups, their oblique factor coefficients on the common model referents only would be the COMP1 input leading to a direct comparison of the O-cluster structures of the different samples. The different groups of individuals could be random samples drawn from the full supply, in which case the final COMP2 analysis would reveal an abridged configuration of objects in the full supply from which the final O-clusters could be determined. But the groups could also be, for example, different social groups whose O-cluster structures are being explored by COMP for differential multivariate selection.

User's description of COMP1

Introduction

This component of the BCTRY system is designed to facilitate the preparation of data cards for COMP2. Thus COMP1 simply prepares a deck of data cards required by COMP2 for any given group, from information available on the intermediate storage tape, put there as the result of a prior factoring procedure (NC + NCSA, CC + CSA, or FALS + GYRO). These cards may be saved and then later combined with data decks from other groups for input into COMP2. For the analyst who wishes to input directly to COMP2, the output of COMP1 described below constitutes specific directions on how to prepare the input cards to COMP2.

Method

Two points should be made clear. First, the use of COMP1 is not prerequisite to the use of COMP2. If the analyst likes, he may punch all or any part of the cards required by COMP2, or he may allow COMP1 to do the work. Second, COMP1 prepares only the data cards for COMP2. The analyst must punch his own control cards for COMP2.

Since COMP1 is designed to output the salient results of a factoring precedure for later comparison, COMP1 should be called after the factoring has been performed by CC5, NC2, or FALS, and the resulting factor coefficients have been rotated by CSA, NCSA, or GYRO.

Use

Card input. The executive control card is the only card ****
required. It should be punched with '/COMP1+' in cols. 1-7. ****

Tape input. The program reads the following four files from the intermediate storage tape (IST):

NAME OF FILE	PROGRAMS PRODUCING FILE	ACTION TAKEN BY PROGRAM IF FILE NOT PRESENT ON IST
IDFILE	DAP	Print error message, punch IST restart deck, and stop processing.
RFACT1	CSA,NCSA,GYRO	
VNAMS1	DAP	Generate a set of variable names of the form V001, V002, ... etc., and continue processing.
CLUST1	NC2, CC5	Set an indicator KLUST=1 and continue (otherwise KLUST=0).

COMP1 produces no printed or IST output - only cards. The deck of cards produced by COMP1 contains the above files punched in the proper order and in the format suitable for input to COMP2. The data cards required by COMP2 and output by COMP1 are as follows:

Card A. Deck title. This card contains the statement 'data cards prepared by COMP1 for COMP2'. It is intended to aid the analyst in identifying the output of COMP1. Although the contents of this card are ignored by COMP2, the card must be physically present in the COMP2 input deck.

Card B. Group identification (IDFILE). A title card containing the title to be associated with all COMP2 output for a given group.

Card C. Parameters. An integer card containing the number of variables (NV), number of dimensions (M), and the indicator specifying whether the key cluster variable indices are present (KLUST). The number is punched in cols. 1-4, M in cols. 5-8, and KLUST in cols. 9-12 (if indicators are not present KLUST = 1, otherwise KLUST = 0).

Cards D. Rotated factor coefficients (RFACT1). Let F be the martix of rotated factor coefficients, where F(I,J) is the factor coefficient of variable J on dimension I, i.e., the rows correspond to dimensions and the columns to variables. Then F is punched as a matrix in decimal card format.

Cards E. Variable names (VNAMS1). The six character variable names are punched as fixed field name cards.

Cards F. Key cluster indices (CLUST1). These cards are used only when KLUST = 0. Let S(I) be the number of variables defining dimension I, and let S(I) be the indices of the variables defining dimension I, that is, S(I) = (V(I,1), V(I,2), ..., V(I,S(I)). Then cards F are punched as follows: the first card contains (S(1), S(2), ..., S(M)) punched as a vector on integer cards; the second and all succeeding cards contain the sets of indices, S(1), S(2), S(3),..., S(M). Each SK is punched as a vector on integer cards in the order shown above, i.e., S1 first, then S2, S3, etc.

Labeling of card output. In addition to the punching in cols. 1-72 required by COMP2, cols. 73-80 are used by COMP1 to identify the card output as shown below. This identification is for the analyst's convenience only, since cols. 73-80 are not read by COMP2.

CARD OR CARDS AND THEIR CONTENTS

	cols. 73-74	cols. 75-80
B. Title	'24'	'IDFILE'

	cols. 73-80
C. NV, M, KLUST	'PARAMTRS'

	cols. 73-74	cols. 75-78	cols. 79-80
D. Factor coeffs.	'08'	Dimension no.	Card sequence no.
E. Variable names	'17'	Blank	Card sequence no.
F. S1, S2, ...,S(M)	'09'	Blank	Card sequence no.
F. S1, S2, ..., S(M)	'09'	Dimension no.	Card sequence no.

There is no tape output from COMP1.

User's description of COMP2

Introduction

COMP2 performs a comparative cluster or factor analysis. That is, this program is designed to calculate a matrix of indices of similarity (analogous to a correlation matrix) among a set of dimensions and to store them for the use of other programs in the BCTRY system. The dimensions can be derived from any form of factoring, and can be orthogonal or oblique.

For the purpose of providing in one overall analysis a comparative description of the dimensions discovered in different groups, a clustering or factoring program (NC, CC, FALS) is applied to this 'correlation' matrix.

COMP is quite general and can perform comparative analyses of any kind of matrix in different groups so long as, in all groups, (1) the rows refer to common variables (or objects) and (2) the entries in the matrix are relations between the row and column variables (or objects).

Method

The method used by COMP2 may be broken down into the following three steps:

(A) Input formation of adjoined matrix of factor coefficients, F. The identification, variable-names, factor coefficients, and cluster indicators (optional), are read for each of the groups. As each matrix of factor coefficients is read in, it is printed and then adjoined to the matrices of factor coefficients from prior groups. The rows of this matrix of adjoined factor coefficients correspond to variables, the columns to dimensions. The rows are assigned variable-names, while the columns are assigned special names on the ordinal position of the reference groups and the ordinal position of the dimensions within each reference group. For example, given two groups each with two dimensions, the dimensions would be named as follows:

COLUMN OF THE ADJOINED MATRIX	NAME ASSIGNED	CONTENTS OF COLUMN: FACTOR CO-EFFICIENTS OF ALL VARIABLES
1	G01D01	First dimension of first group.
2	G01D02	Second dimension of first group.
3	G02D01	First dimension of second group.
4	G02D02	Second dimension of second group.

(B) Selecton of rows and columns of the adjoined matrix, F, to be used for further analysis. Three options are available in selection of rows: (1) use all rows, (2) the names of the rows to be used are input by the analyst, and (3) use only rows corresponding to the defining variables of the dimensions. Two options are available in selection of columns (dimensions) to be compared: (1) use all columns, and (2) the names of the columns to be used are input by the analyst.

If any selection is performed (i.e., unless all rows and columns are used), the adjoined matrix is printed after selection.

(C) Calculation of the indices of similarity among the columns. Three methods are available: the cosine method, the signed P square method, and the Pearson product moment correlation method. The matrix of similarity indices is printed using the column (dimension) names as labels. The analysis identification, column names, and matrix of similarity indices are written on IST.

Use

This section assumes, even for the operation of the program, that certain definitions given in section 2 have been understood by the reader.

```
     Card input.                                                        ****
     (1)  Executive control card.  Punch '/COMP2+' in cols. 1-7.        ****
     (2)  Component control cards.  These cards contain the identifi-   ****
cation and certain control information specifying the methods to be     ****
used in the analysis.                                                   ****
                                                                        ****
     Card A.  This card contains the information to be printed as a     ****
general title of the analysis.  It should be punched using the title    ****
card format.                                                            ****
                                                                        ****
     Card B.  This card contains the number of groups to be analyzed    ****
and the specific options to be used in the analysis.  It is punched as  ****
in integer card.                                                        ****
                                                                        ****
     COLUMNS AND CONTENTS OF CARD B                                     ****
                                                                        ****
     1-4     NG, the number of groups of data to be analyzed.           ****
     5-8     This field specifies the method of selecting rows of the   ****
             matrix of adjoined factor coefficients, F.                 ****
             '0'.  Use all rows.                                        ****
             '1'.  Use rows whose names are input by analyst (see Card  ****
                   C below).                                            ****
             '2'.  Use rows corresponding to defining variables of      ****
                   key clusters (standard).                             ****
     9-12    This field specifies the method of selecting columns of    ****
             the matrix F.                                              ****
             '0'.  Use all columns (standard).                          ****
             '1'.  Use columns corresponding to names input by analyst  ****
                   (see Cards D below).                                 ****
     13-16   This field specifies the similarity index to be used in    ****
             the comparison.                                            ****
             '0'.  Use the cosine method (standard).                    ****
             '1'.  Use the signed P square method.                      ****
             '2'.  Use the correlation coefficient method.              ****
                                                                        ****
     Cards C.  These cards are required only when cols. 5-8 of Card B   ****
contain a '1'.  They contain the names of the rows of the adjoined      ****
matrix, F, to be selected for comparison.  The names are punched in a   ****
free field name card format (see COMP1).                                ****
```

Cards D. These cards are required only when cols. 9-12 of Card B ****
contain a '1'. They contain the names of the columns of the adjoined ****
matrix, F, to be selected for comparison. The names are punched in a ****
free field name card format (see Glossary). ****
****

Data cards. These cards contain the data required by COMP2, ****
i.e., the output decks from COMP1 or decks in COMP1 format. As stat- ****
ed in the description of Method, COMP2 requires for each group an ****
identification label, a set of factor coefficients, and a set of vari- ****
able names, and, if cols. 5-8 of Card B contain a '2', a set of cluster ****
indicators. The component program COMP1 prepares decks of cards con- ****
taining these data. These COMP1 decks (one deck for each group) ****
should be interpreted, stacked together, and input to COMP2. Since ****
the deck for, say, the Gth group is obtained by merely including a car ****
with '/COMP1' in cols. 1-6 (see the program description of COMP1) ****
after the factoring of the Gth group, the user will normally not be re ****
quired to manually punch these cards. However, should the analyst wis ****
to prepare these cards himself, the format of these decks is fully des ****
cribed in the program description of COMP1. ****

COMP2 does not read any information from the shared data tapes.

The printed output of the program includes the factor coefficients for each group from COMP1, the factor coefficients selected for COMP2 analysis, and the similarity indices among dimensions. The variables and factors are always identified by name in the printed output since certain of the control options may result in changes of ordinal position.

The following files are written on IST by COMP2:

FILE NAME	CONTENTS
IDFILE	Same as 'analysis identification card'.
VNAMS1	Names of dimensions compared by COMP2.
CORRM1	Matrix of similarity indices.

Restrictions. Number of dimensions within any group is at most 15, and the total number of dimensions for all groups combined is 90.

Important: to complete the COMP analyses, the above COMP2 cards must be followed by those that execute a full cycle key cluster analysis. Since COMP2 outputs CORRM1, then the cards that follow them are DVP50(or 60)-CC5-CSA2-SPAN2.

USER'S DESCRIPTION OF THE COMPARATIVE SIMILARITY ROTATION MODEL.. SIMRO (SYDA AND SYRN)

GENERAL INTRODUCTION

THE PURPOSES OF SIMRO ARE, BROADLY SPEAKING, THE SAME AS THOSE OF COMP.. THE COMPARISON OF THE DIMENSIONS DERIVED IN DIFFERENT GROUPS. THE MAIN DIFFERENCE BETWEEN THE TWO METHODS GOES TO THE LOGIC OF MAKING COMPARISONS BETWEEN ANY ENTITIES, HERE, DIMENSIONS. ONE WAY IS DIRECT COMPARISON, NAMELY, COMPARING DIRECTLY THE DIMENSIONS OF ONE GROUP WITH THOSE OF ANOTHER.. THAT IS THE PURPOSE OF COMP. ANOTHER WAY IS INDIRECT COMPARISON WITH A COMMON CRITERION, NAMELY, COMPARING THE DIMENSIONS OF EACH GROUP WITH THOSE OF A CRITERION POPULATION.. THAT IS THE PURPOSE OF SIMRO, IN WHICH THE DIMENSIONS OF THE DIFFERENT GROUPS ARE COMPARED BY MEASURING THEIR DEGREES OF IDENTITY WITH THE DIMENSIONS OF A COMMON HYPOTHETICAL POPULATION DERIVED FROM THE GROUPS.

IN PRINCIPLE, THE PROCEDURES OF SIMRO AND COMP ARE BROADLY MUCH THE SAME. BOTH REQUIRE PRELIMINARY PROGRAMS THAT PREPARE THE DIMENSIONAL DATA FROM THE K GROUPS FOR COMPARATIVE TREATMENT.. FOR SIMRO IT IS SYDA, FOR COMP IT IS COMP1. BOTH REQUIRE, OF COURSE, A SET OF DIMENSIONS IN EACH GROUP, BUT IN SIMRO THE NUMBER OF THEM MUST BE EXACTLY THE SAME IN ALL GROUPS, WHEREAS IN COMP THE NUMBER OF DIMENSIONS CAN BE DIFFERENT IN THE GROUPS, THOUGH OPTIONS ARE PROVIDED FOR THE DELETION OF ANY DIMENSION IN ANY GROUP, IF DESIRED. IT IS ALSO TO BE NOTED THAT IN SIMRO THE DIMENSIONS PROVIDED BY SYDA ARE THE ORTHOGONAL DIMENSIONS DERIVED BY FACTORING, WHEREAS IN COMP THEY ARE THE OBLIQUE DIMENSIONS DERIVED IN THE CLUSTER STRUCTURE PROGRAM, CSA. BOTH SIMRO AND COMP REQUIRE THE SAME SET OF REFERENT ROW VARIABLES IN THE FACTOR MATRICES COMPARED, IN SYDA THAT NUMBER MUST BE THE SAME CONSTANT NV VARIABLES FOR SYRN ANALYSIS, WHEREAS IN COMP, OPTIONS ARE PROVIDED IN COMP2 FOR DELETIONS OF ANY REFERENT ROW VARIABLES PROVIDED BY COMP1. BOTH SIMRO AND COMP USE AS AN INDEX OF SIMILARITY OF DIMENSIONS THE CORRELATION BETWEEN HYPOTHETICAL DOMAIN OR FACTOR SCORES ON THE DIMENSIONS, I.E., THE 'COMMON FACTOR R', REFERRED TO IN COMP AS COS THETA.

USER'S DESCRIPTION OF THE SYRN DATA PUNCHING PROGRAM.. SYDA

INTRODUCTION

THE PROGRAM IS DESIGNED TO AID THE USER IN PREPARING THE DATA CARDS REQUIRED BY THE SIMILARITY ROTATION PROGRAM, SYRN.

METHOD

SYDA PREPARES THE DATA BY READING THE IDENTIFICATION, STANDARD DEVIATIONS, AND FACTOR COEFFICIENT FILES FROM THE INTERMEDIATE STORAGE TAPE (IST), AND PUNCHING THESE DATA ONTO DECIMAL CARDS IN A FORMAT ACCEPTABLE TO SYRN. THUS THE PROGRAM CAN BE USED AS A CONVENIENT AID IN PREPARING, FOR ANY GIVEN DATA SET, THE DATA CARDS FOR SYRN. THESE CARDS MAY BE STORED AND LATER COMBINED WITH CARDS FROM OTHER DATA SETS FOR INPUT TO SYRN.

USE

```
     EXECUTIVE CONTROL CARD.  PUNCH '/SYDA++' IN COLUMNS 1-7.  THE         ****
PROGRAM REQUIRES NO OTHER CARD INPUT.                                      ****
                                                                           ****
     THE DATA SET IDENTIFICATION (IDFILE), STANDARD DEVIATIONS OF THE      ****
VARIABLES (STDEV1), AND UNROTATED FACTOR COEFFICIENTS (UFACT1) ARE READ    ****
FROM IST.  THE ABSENCE OF ANY OF THESE DATA WILL PRODUCE AN ERROR STOP.    ****
                                                                           ****
     THE PROGRAM PRODUCES NO PRINTED OR TAPE OUTPUT, ONLY PUNCHED CARDS    ****
                                                                           ****
THESE CARDS ARE INPUT IN THE FOLLOWING ORDER..                             ****
                                                                           ****
     DATA SET TITLE CARD.  THE CONTENTS OF IDFILE ARE PUNCHED USING THE    ****
TITLE CARD FORMAT.                                                         ****
                                                                           ****
     STANDARD DEVIATION DATA CARDS.  THE CONTENTS OF STDEV1 ARE PUNCHED    ****
AS A VECTOR USING THE DECIMAL CARD FORMAT.                                 ****
                                                                           ****
     FACTOR PATTERN DATA CARDS.  THE CONTENTS OF UFACT1, ARRANGED SO       ****
THAT ROWS CORRESPOND TO FACTORS, COLUMNS TO VARIABLES, ARE PUNCHED AS      ****
A MATRIX ON DECIMAL CARDS.                                                 ****
                                                                           ****
     THE ABOVE DESCRIPTION SPECIFIES THE CONTENTS OF COLUMNS 1-72.         ****
SINCE SYRN DOES NOT READ COLUMNS 73-80 OF THE DATA CARDS, SYDA USES        ****
THESE COLUMNS TO PROVIDE CODE NAMES FOR EACH TYPE OF CARD, AND TO          ****
SPECIFY THE CORRECT ORDER OF THE CARDS OF EACH TYPE.                       ****
                                                                           ****
     COLUMNS AND CONTENTS OF THE OUTPUT CARDS.                             ****
                                                                           ****
     73-74   THESE COLUMNS PROVIDE A CODE NUMBER FOR THE CONTENTS OF       ****
             THE CARD..                                                    ****
             ''24''   DATA SET TITLE CARD.                                 ****
             ''2''    STANDARD DEVIATION DATA CARDS.                       ****
             ''7''    FACTOR PATTERN DATA CARDS.                           ****
     75-78   THESE COLUMNS ARE BLANK ON THE TITLE AND STANDARD             ****
             DEVIATION CARDS.  ON THE FACTOR PATTERN CARDS, THESE COL-     ****
             UMNS CONTAIN THE NUMBER OF THE FACTOR DIMENSIONS WHOSE        ****
             COEFFICIENTS APPEAR ON THE CARD.                              ****
     79-80   THESE COLUMNS ARE BLANK ON THE TITLE CARD.  ON THE STAN-      ****
             DARD DEVIATION AND FACTOR PATTERN CARDS, THE COLUMNS CON-     ****
             TAIN THE CARD SEQUENCE NUMBER.                                ****
```

USER'S DESCRIPTION OF COMPARATIVE ANALYSIS BY SIMILARITY ROTATION.. SYRN

INTRODUCTION

FROM THE FACTOR MATRICES OF THE DIFFERENT GROUPS TO BE COMPARED, A BEST-FITTING FACTOR PATTERN MATRIX OF A POPULATION MATRIX, P, IS COMPOSED BY A LEAST SQUARES SOLUTION. THE GENERAL LOGIC OF THE ANALYSIS IS THAT OF STATING THE DEGREE TO WHICH THE FACTOR PATTERN MATRICES OF THE DIFFERENT GROUPS CAN BE CONSIDERED TO BE SELECTIONS FROM THE POPULATION MATRIX.

A GIVEN GROUP IS A RANDOM SELECTION FROM THE POPULATION IF THE GROUP'S FACTOR MATRIX IS INDISTINGUISHABLE FROM THE P MATRIX. IT IS A NON-RANDOM OR BIASSED SELECTION IF ITS MATRIX IS A SALIENT DEVIATION FROM THAT OF P. A BIASSED SELECTION CAN BE OF SEVERAL SORTS. THE FACTORED DIMENSIONS OF THE GROUP CAN BE THE SAME AS THOSE OF P, BUT ON THE SCALE OF MEASUREMENT OF SOME OF THE DIMENSIONS THE FREQUENCY OF INDIVIDUALS AT SOME POINTS ON THE SCALE IS EITHER SIGNIFICANTLY LESS THAN THAT IN THE POPULATION, OR IT IS SIGNIFICANTLY MORE. WE SHALL CALL THIS KIND OF BIASSED SELECTION SUBJECT-SELECTION BIAS. BUT THE GROUP CAN ALSO BE A BIASSED SELECTION BY HAVING DIFFERENT DIMENSIONS, LET US CALL THIS TYPE DIMENSIONAL BIAS. IF A GROUP HAS A SUBJECT-SELECTION BIAS, IT CAN BE SHOWN THAT THE FACTOR MATRIX CAN BE ROTATED SO AS TO MATCH THAT OF P EXCEPT FOR DIFFERENCES INTRODUCED BY VARIATION IN THE UNITS OF MEASUREMENT USED. IF SELECTION INTRODUCES NEW OR DIFFERENT DIMENSIONS, NAMELY A DIMENSIONAL BIAS, THEN NO SATISFACTORY ROTATION CAN BE ACHIEVED.

SYRN PROVIDES A MEASUREMENT OF GOODNESS OF FIT OF THE ROTATED FACTOR MATRIX OF EACH GROUP WITH THAT OF A POPULATION MATRIX ALSO DETERMINED BY SYRN, SO THAT ONE HAS A MEASURE OF THE EXTENT TO WHICH SUBJECT-SELECTION BIAS IS OPERATING. TO THE EXTENT THAT THE FIT IS GOOD THE HYPOTHESIS OF RANDOM SELECTION OR OF SUBJECT-SELECTION BIAS CANNOT BE REJECTED. A POOR FIT SIGNIFIES DIMENSIONAL BIAS.

THE PROCEDURE CONSISTS OF ROTATING THE FACTORED DIMENSIONS OF THE THE COMPARISON GROUPS TO POSITIONS AS COICIDENT AS POSSIBLE WITH THOSE OF P. THEN THE CORRELATIONS BETWEEN THE DIMENSIONS, OR FACTORS, WITHIN EACH GROUP ARE COMPUTED. IF THEY MATCH THOSE OF P THEN THE GROUPS ARE NOT DIMENSIONALLY BIASSED. FOR EACH GROUP THE CORRELATIONS OF EACH VARIABLE WITH THE ROTATED DIMENSIONS ARE COMPUTED (THE ROTATED FACTOR MATRIX). THE DIMENSIONS BOTH OF P AND OF THE DIFFERENT GROUPS ARE USUALLY OBLIQUE.

COMPARISON OF SIMRO AND COMP (SEE ALSO THE GENERAL INTRODUCTION TO SIMRO).--THE UNROTATED, OR SYDA, DIMENSIONS OF EACH OF THE K GROUPS MUST BE EQUAL IN NUMBER, NAMELY M, A RESTRICTION NOT IMPOSED BY COMP. THE SYDA DIMENSIONS ARE ALSO ORTHOGONAL, YIELDING RAW UNROTATED COEFFFICIENTS (UFACT1) USUALLY THOUGH NOT NECESSARILY DETERMINED BY TOTAL-SET FACTORING BY FALS. BEING ORTHOGONAL THEY USUALLY ARE DIFFICULT TO INTERPRET. SINCE THE BEST-FIT DIMENSIONS OF P ARE A FORM OF AVERAGE OF THE M DIMENSIONS OF THE K GROUPS, THEY BECOME OBLIQUE BECAUSE

OF THE EFFECTS OF SUBJECT-SELECTION OR DIMENSIONAL BIAS. AS AVERAGES OF THE DIFFICULT-TO-INTERPRET ORTHOGONAL DIMENSIONS OF THE GROUPS, THE M DIMENSIONS OF P ARE PROBABLY ALSO DIFFICULT TO INTERPRET. NEVERTHELESS, THESE DIFFICULTIES DO NOT GAINSAY THE POWER OF SYRN TO DESCRIBE THE SIMILARITY OF THE FIRST M DIMENSIONS OF EACH GROUP TO THOSE OF P.

THE USER MAY WISH TO PERFORM A COMP ANALYSIS ANALOGOUS TO THAT OF SYRN. HE WOULD DO SO BY COMBINING ALL K GROUPS INTO ONE GROUP, P, THEN PERFORMING A FULL CYCLE V-ANALYSIS ON IT, DRAWING FROM IT A COMP1 DECK ON CSA OF THE OVERALL P GROUP. WHEN THE COMP 1 CARDS BOTH OF P AND OF THE K GROUPS ARE PROJECTED INTO ONE COMP2 ANALYSIS, THEN THE DEGREE OF SIMILARITY OF ALL OBLIQUE DIMENSIONS OF ALL GROUPS INTER-SE AND WITH P WOULD BE SHOWN BY THE STATISTICAL STRUCTURE (CSA) AND BY THE GEOMETRIC STRUCTURE (SPAN) PROGRAMS. THE COMP RESULTS WOULD BE STRICTLY COMPARABLE TO THOSE OF SYRN ONLY IF THE SYDA CARDS CONTAIN THE OBLIQUE FACTOR COEFFICIENTS (RFACT1) FROM CSA, AND IF THE COLUMN DELETION OPTION OF COMP2 IS EXERCISED TO RESTRICT THE DIMENSIONALITY OF ALL GROUPS TO A CONSTANT SET OF M DIMENSIONS.

METHOD

THE GENERAL DESIGN OF SYRN REQUIRES THAT EACH OF THE K COMPARISON GROUPS BE FACTORED ON THE SAME NUMBER, M, OF DIMENSIONS, AND THAT DATA CARDS FROM SYDA ON EACH GROUP ARE AVAILABLE FOR INPUT TO SYRN. THE FACTOR PATTERN MATRIX OF P IS THEN CONSTRUCTED AFTER APPROPRIATE SCALING OF THE MATRICES OF THE DIFFERENT GROUPS. THEN THE M DIMENSIONS OF EACH GROUP ARE ROTATED TO POSITIONS AS COINCIDENT AS POSSIBLE WITH THOSE OF P, THE CORRELATIONS BETWEEN DIMENSIONS IN EACH GROUP ARE COMPUTED, AND THEIR ROTATED OBLIQUE FACTOR MATRIX COMPUTED. FINALLY, INDICES OF GOODNESS OF FIT OF EACH GROUP TO P ARE CALCULATED. SPECIFICALLY, HERE ARE THE STEPS..

A. RESCALING OF THE FACTOR MATRIX OF EACH GROUP SO THAT THE SD OF THE HYPOTHETICAL SCORES OF THE DIFFERENT DIMENSIONS IN A COMPOSITE POPULATION COMPOSED OF THE K SUBGROUPS WOULD BE EQUAL (I.E., UNITY).

B. CALCULATING BY LEAST SQUARES THE BEST FITTING FACTOR PATTERN MATRIX P FROM THE RESCALED FACTOR MATRICES OF THE SUBGROUPS.

C. CALCULATING THE TRANSFORMATION MATRIX GIVING THE ROTATIONAL CONSTANTS THAT SERVE TO PROJECT THE DIMENSIONS OF EACH GROUP TO POSITIONS AS COINCIDENT AS POSSIBLE WITH THOSE OF P.

D. WITHIN EACH GROUP, CALCULATING THE CORRELATIONS BETWEEN THE DIMENSIONS (CORRELATIONS BETWEEN FACTORS).

E. WITHIN EACH GROUP, CALCULATING THE CORRELATIONS OF EACH VARIABLE WITH THE ROTATED DIMENSIONS OF THE GROUP (ROTATED FACTOR MATRIX).

F. MEASURING THE GOODNESS OF FIT OF THE DIMENSIONS OF EACH GROUP WITH THOSE OF P BY..

(1) THE TRACE CRITERION.. CALCULATING AN INDEX OF FIT OF EACH GROUP FROM THE SQUARED DIFFERENCES BETWEEN ITS ROTATED FACTOR COEFFICIENTS AND THE CORRESPONDING FACTOR COEFFICIENTS OF P.

(2) DIFFERENCES.. TABLING THE DIFFERENCES USED IN THE TRACE CRITERION.

ALTERNATIVE METHODS. TWO CONTRASTING METHODS ARE AVAILABLE TO CALCULATE THE BEST-FITTING POPULATION MATRIX P (STEP B ABOVE) AND TO EXECUTE THE REMAINING STEPS OF SIMILARITY ROTATION (STEPS C TO F).

A. METHOD 1.. THE LOGIC OF THIS PROCEDURE IS TO ROTATE THE MATRICES OF THE KNOWN SUBGROUPS TO THE UNKNOWN BEST-FITTING P MATRIX.

B. METHOD 2.. IN THIS PROCEDURE AN UNKNOWN P MATRIX THAT MEETS CERTAIN RATIONAL CONDITIONS IS SO ROTATED AS BEST TO FIT THOSE OF THE KNOWN SUBGROUPS.

SINCE THE USER MUST DECIDE WHICH METHOD TO USE, HERE ARE THE GROUNDS THAT TEND TO RECOMMEND METHOD 1. BY THIS FIRST METHOD ONE PROCEDES FROM KNOWNS TO AN UNKNOWN, SEEMINGLY PREFERABLE TO THE REVERSE APPROACH. FURTHERMORE, METHOD 1 INCLUDES A MAXIMIZING FUNCTION THAT ASSURES THAT THE FIRST MOST SALIENT DIMENSIONS OF THE P MATRIX WILL PROVIDE A BETTER FIT TO THOSE OF THE SUBGROUPS THAN WILL THE LATER LESS SALIENT DIMENSIONS. MORE EXPERIENCE WITH AND STUDY OF THE RESULTS OF THE TWO METHODS, HOWEVER, ARE NEEDED TO CLARIFY THE MEANINGFULNESS OF DIFFERENCES IN THE FINDINGS BY THESE CONTRASTING METHODS.

USE

CARD INPUT. ****
(1) EXECUTIVE CONTROL CARD. PUNCH '/SYRN++' IN COLS. 1-7. ****
(2) COMPONENT CONTROL CARDS. THESE CARDS SPECIFY THE METHODS TO ****
BE USED, PROVIDE CERTAIN DATA-SET PARAMETERS, AND SPECIFY THE NAMES OF ****
THE VARIABLES. ****
****
CARD A. THIS CARD SPECIFIES THE METHOD OF COMPARISON AND THE FOR- ****
MAT TO BE USED TO READ THE DATA CARDS, AS FOLLOWS.. ****
****
COLUMNS AND CONTENTS OF CARD A. ****
1-4 DETERMINE THE METHOD OF COMPARISON TO BE USED AS DESCRIBED ****
IN THE PREVIOUS SECTION. ****
''1'' METHOD 1 IS USED (STANDARD). ****
''2'' METHOD 2 IS USED. ****
5-8 DETERMINE THE FORMAT TO BE USED IN READING DATA CARDS. ****
''1'' THE DECIMAL CARD FORMAT, DESCRIBED IN THE GLOSSARY ****
IS USED (STANDARD). ****
''2'' THE FORMAT IS INPUT BY THE ANALYST ON VARIABLE FOR- ****
MAT CARDS. SEE DATA CARDS BELOW. ****

```
     CARD B.  THIS CARD CONTAINS THE NUMBER OF VARIABLES, DIMENSIONS       ****
AND DATA SETS.  IT IS AN INTEGER CARD.                                     ****
                                                                           ****
     COLUMNS AND CONTENTS OF CARD B.                                       ****
     1-4     NV, NUMBER OF VARIABLES.                                      ****
     5-8     M, NUMBER OF FACTORS.                                         ****
     9-12    N, NUMBER OF DATA-SETS.                                       ****
                                                                           ****
     CARDS C.  THESE CARDS CONTAIN THE SET OF NV NAMES TO BE ASSOCIATED    ****
WITH THE VARIABLES IN THE PRINTED OUTPUT.  THE NAMES ARE PUNCHED ON        ****
FREE FIELD NAME CARDS.                                                     ****
                                                                           ****
     (3)  DATA CARDS.  UNLIKE MOST OF THE OTHER BC TRY COMPONENTS, SYRN     ****
READS ALL DATA FROM CARDS, (1) TITLE IDENTIFYING THE ENTIRE ANALYSIS,      ****
(2) STANDARD DEVIATIONS OF THE MANIFEST VARIABLES IN EACH OF THE N DATA    ****
SETS, (3) THE IDENTIFYING TITLE AND FACTOR PATTERN MATRIX FOR EACH OF      ****
THE N DATA SETS.  ALL OF THIS INFORMATION IS INPUT ON DATA CARDS IN THE    ****
ORDER SHOWN BELOW..                                                        ****
                                                                           ****
     IN THE FOLLOWING, LET S DENOTE THE MATRIX WHOSE ROW G IS THE          ****
VECTOR OF STANDARD DEVIATIONS OF THE NV VARIABLES IN DATA SET G, I.E.,     ****
WITH ROWS CORRESPONDING TO DATA-SETS AND COLUMNS TO VARIABLES.  LET        ****
F(G) DENOTE THE ORTHOGONAL FACTOR PATTERN (STRUCTURE) MATRIX FOR DATA      ****
SET G ARRANGED SO THAT ROWS CORRESPOND TO THE M FACTORS AND COLUMNS TO     ****
THE NV VARIABLES.                                                          ****
                                                                           ****
     STANDARD DEVIATIONS OF ALL DATA SETS.  THE DATA CARDS CONTAIN THE     ****
MATRIX OF STANDARD DEVIATIONS, S, PRECEDED BY A TITLE CARD, AND, IF        ****
COLS. 5-8 OF CARD A CONTAIN A '2', A CARD DESCRIBING HOW THE S.D.'S ARE    ****
PUNCHED.                                                                   ****
                                                                           ****
     ANALYSIS TITLE CARD.  THIS CARD CONTAINS THE IDENTIFICATION OF THE    ****
     ENTIRE ANALYSIS FOR PRINTED OUTPUT, PUNCHED IN A TITLE CARD FOR-      ****
     MAT.                                                                  ****
                                                                           ****
     S.D. FORMAT CARD.  THIS CARD IS NECESSARY ONLY WHEN COLS. 5-8 OF      ****
     CARD A CONTAIN A '2'.  IT CONTAINS THE FORMAT USED IN PUNCHING THE    ****
     STANDARD DEVIATIONS, PUNCHED AS A VARIABLE FORMAT CARD (SEE           ****
     GLOSSARY).                                                            ****
                                                                           ****
     S.D. DATA CARDS.  PUNCH S AS A MATRIX, USING THE DECIMAL CARD         ****
     FORMAT, UNLESS COLS. 5-8 OF CARD A CONTAIN A '2'.                     ****
                                                                           ****
     FACTOR PATTERN MATRICES FOR ALL DATA-SETS.  THE FACTOR PATTERN MA-    ****
TRICES F(G) ARE INPUT SUCCESSIVELY FOR THE G = 1, 2, ... N DATA SETS.      ****
EACH MATRIX F(G) MUST BE PRECEDED BY A 'DATA-SET TITLE' CARD.  THE         ****
CONTENTS OF THIS CARD WILL BE USED AS A LABEL TO IDENTIFY THE OUTPUT OF    ****
THE DATA-SET.  IF COLS. 5-8 OF CARD A CONTAIN A '2', EACH MATRIX MUST      ****
ALSO BE PRECEDED BY A CARD SPECIFYING THE FORMAT FOR ANY ROW OF F(G).      ****
THUS, FOR DATA SET G..                                                     ****
                                                                           ****
     DATA-SET TITLE CARD.  PUNCH A DISTINCT LABEL FOR DATA SET G           ****
     USING THE TITLE CARD FORMAT.                                          ****
     FACTOR PATTERN FORMAT CARD.  THIS CARD IS REQUIRED ONLY WHEN COLS.   ****
     5-8 OF CARD A CONTAIN A '2'.  IT SHOULD CONTAIN THE FORMAT DE-       ****
     SCRIBING ANY ROW OF F(G).  PUNCH AS A VARIABLE FORMAT CARD.  NOTE    ****
     THAT DIFFERENT FORMATS MAY BE USED WITH DIFFERENT DATA-SETS.         ****
                                                                          ****
     FACTOR PATTERN DATA CARDS.  PUNCH F(G) AS A MATRIX, USING THE        ****
     DECIMAL CARD FORMAT (UNLESS COLS. 5-8 OF CARD A CONTAIN '2').        ****
```

PRINTED OUTPUT INCLUDES A DESCRIPTION OF THE METHOD USED, THE BEST FITTING FACTOR PATTERN MATRIX, THE ASSOCIATED EIGENVALUES, AND THE RESULTS OF THE ROTATIONS OF EACH OF THE DATA-SETS. THE TRANSFORMATION MATRIX, ROTATED FACTOR CORRELATION MATRIX, ROTATED FACTOR STRUCTURE MATRIX, AND GOODNESS OF FIT CRITERIA ARE PRINTED SO THAT THE RESULTS OF ROTATION FOR EACH DATA BEGIN A NEW PAGE. ALL PAGES ARE CONSECUTIVELY NUMBERED.

THE PROGRAM DOES NOT PRODUCE TAPE OUTPUT.

IN ADDITION TO THE USUAL BC TRY RESTRICTIONS, THE MAXIMUM NUMBER OF DATA-SETS IS 10.

User's description of symbolic matrix interpretative system: SMIS (modification of FI BC SMIS by E. Wilson)

Introduction

This component performs general matrix-algebraic operations and provides the user with almost unlimited capacity for matrix and vector operations with access to the IST. Each operation defined under SMIS is initiated by the user by control cards. By combining the operations provided in SMIS the user can perform virtually any calculation desired even though the calculation is not an integral part of the other components of the BCTRY. Hence, with creative use of SMIS, novel and pioneering work can be done in multivariate analysis within the setting of BCTRY. For example, new forms of factoring can be formulated in terms of operations of SMIS and tested without resort to new programming. Another example is the calculation of the squared multiple correlation estimates of communality for a matrix to be factored. The clerical capacity of SMIS is of great importance. Often one wishes to perform very simple clerical transformations of the data available in an IST file, e.g., transposing the factor score matrix for input to another component via IST. Prior to the inclusion of SMIS into the system the user had to call for the file by GIST and manually transpose the matrix. Many such operations can be performed within the system under SMIS.

Method

SMIS is a collection of subprograms called by a main program when it encounters a control card punched in a specified manner. The sequence of subprograms called determines the mathematical operations performed. This sequence is entirely specified by the deck of cards provided by the user along with the other BCTRY control cards.

Three types of cards may be used to define a SMIS calculation: operation cards, data cards, and remark cards. An operation card contains information regarding the operation to be performed, the matrices involved in the operation, the numbers of rows and columns of the matrices, and constants involved in the calculation. Data cards contain the elements of matrices and vectors required in the calculations. These data are preceded in the SMIS deck by an operation card indicating the name of the matrix and the number of columns and rows involved. Remark cards are used to label output, e.g., to indicate the name of a matrix printed, the designation of the columns and rows, and the source of the output matrix. Results of calculations may be output as files on the IST. Files on IST generated by other BC TRY components are of three types with respect to SMIS: type 1, type 2, or non-accessible.

Use of SMIS

The number of control cards required depends on the calculations performed. The card input consists of a BCTRY executive control card and a deck of SMIS control cards, data cards and remark cards.

(1) Executive control card. Punch '/SMIS' in cols. 1-7.

(2) SMIS cards. Each type of card is described here, and examples of SMIS programs are presented.

Control cards: each control card has eleven six column fields, some of which may be blank. These fields, their names and their functions are as follows:

FIELD	NAME	FUNCTION OF THE FIELD
1 - 6	Operation	SMIS operation
7 -12	A	Matrix name
13-18	B	Matrix name
19-24	C	Matrix name
25-30	D	Matrix name
31-36	N1	Number of rows in a matrix
37-42	N2	Number of columns in a matrix
43-48	N3	Number of rows in a matrix
49-54	N4	Number of columns in a matrix
55-60	C1	Constant
61-66	C2	Constant

The operation symbol must begin in column 1. Blanks are considered legitimate symbols when referring to names A, B, C, and D. Blanks imply zero when they appear in the number fields N1, N2, N3, and N4, and hence, numbers in these fields should be packed right in the field. Unless decimal points are specified in the constant fields, C1 and C2, they are assumed to be at the extreme right of the field.

The operations and the associated definitions and contents of name, number, and constant fields are as follows:

OPERATION	CALCULATION TO BE PERFORMED AND NUMBERS AND CONSTANTS REQUIRED
ZERO	Formation of null matrix: A = name of null matrix. N1 = number of rows in matrix. N2 = number of columns in matrix.
ADD	Matrix addition (A) + (B) = (A). Matrix A is replaced by the sum of matrices A and B.
SUB	Matrix subtraction (A) - (B) = (A). Matrix A is replaced by matrix A minus matrix B.

MULT

Matrix multiplication (A) (B) = (C).
The product of A times B is generated and is defined as C.

TRANS

Matrix transpose (B) = TRANS (A).
The transpose of A is generated and is defined by B.

INVERT

Matrix inversion (A) = INVRS (A).
Matrix A is replaced by the inverse of itself.

SYMINV

Inversion of symmetric matrix (A) = INVRS (A).
Matrix A is replaced by the inverse of itself.

SCALE

Scalar multiplication (A) = S1 (A).
Each element in matrix A is replaced by S1 times the element.

STOSM

Store submatrix (B) in (A).
A = name of large matrix.
B = name of submatrix to be stored.
N1 = row number in large matrix of first element of submatrix.
N2 = column number in large matrix of first element of submatrix.

ADDSM

Add submatrix (B) to (A).
A = name of large matrix.
B = name of submatrix to be added into large matrix.
N1 = row number in large matrix of first element of submatrix.
N2 = column number in large matrix of first element of submatrix.

RMVSM

Remove submatrix (B) from (A).
A = name of large matrix A.
B = name of submatrix B to be removed.
N1 = row number in large matrix of first element of submatrix.
N2 = column number in large matrix of first element of submatrix.
N3 = number of rows in submatrix B.
N4 = number of columns in submatrix B.

DUPL

Matrix duplication (B) = (A).
Matrix B is defined to be identical with matrix A.

PRINT

Matrix print.
A = name of matrix to be printed.
N1 = number of cards, which follow the control card, to be printed as a label for the matrix.

Column 13 of the control card is used for carriage control. A one in column 13 will cause the matrix to be printed on a separate output page.

DELETE Matrix deletion.
A = name of matrix to be eliminated from core storage.

START This operation causes the elimination of all matrices which are in core storage.

WRITE Write matrix A onto tape N1.
Matrix A remains in core storage.

READ Read matrix from tape N1 and define as A.

REWIND Tape N1 is rewound to its load point.

TIME This operation causes the time that has elapsed from the start of the program to be printed.

REMARK This operation causes the number of cards designated by N1, which follow the operation, to be printed as remark cards.

EIGEN Eigenvalues and Eigenvectors.
The Eigenvalues and Eigenvectors of the system (A) (X) = (B) (X) are determined where (A) is a symmetrical matrix and (B) is a diagonal matrix of positive elements stored as a row matrix.
A = name of matrix (A).
B = name of matrix (B).
C = name of matrix (C) of Eigenvectors stored row-wise.
D = name of row matrix of Eigenvalues.
N1 = the number of Eigenvectors to be calculated. The ordering of Eigenvectors and Eigenvalues is determined by the sign of N1 as follows: if N1 is positive Eigenvalues are arranged in descending absolute values; if N1 is negative Eigenvalues are arranged in ascending absolute values. The Eigenvectors are normalized so that (C) (B) trans (C) = (I). Matrices (A) and (B) are destroyed in the process. The maximum size system that may be solved is 80.

SQREL Square root of each element.
Each element in the matrix designated by A is replaced by the square root of itself.

INVEL Inversion of each element.
Each element in the matrix designated by A is replaced by the reciprocal of the element.

LOG The log of each element.
Each element in the matrix designated by A is replaced by the natural log of the element.

STODG Store row matrix (B) on diagonal of (A).
A = name of square matrix.
B = name of row matrix to be stored.

RMVDG Remove row matrix (B) from diagonal of (A).
A = name of square matrix.
B = name of row matrix composed of the diagonal elements of (A).

MSCALE Scalar multiplication (A) = B (A).
Each element in matrix A is replaced by B times the element where B has previously been defined as a 1 by 1 matrix.

ISTIN Load matrix from the IST.
A = name of matrix to be input.
B = file name of matrix on the IST.
N1 = 1 - file is non-symmetric or type 1.
N1 = 2 - file is symmetric matrix or type 2.

ISTOUT Output matrix to the IST.
A = name of matrix to be output.
B = file name of matrix on the IST.
N1 = 1 - file is non-symmetric or type 1.
N1 = 2 - file is symmetric or type 2.

LOAD Matrix load.
A = name of matrix to be loaded.
N1 = number of rows in matrix.
N2 = number of columns in matrix.

RETURN Return to the BCTRY executive program.

Data cards. Data cards follow the matrix operation 'LOAD' and never appear elsewhere. Data are punched normally in fields of six columns starting in column one, with format (12F6.0). The elements of the matrix are punched by rows. If the decimal point is omitted, it is assumed to be at the extreme right of the field. The standard format may be superseded by punching any desired format in columns 13-30 of the control card.

Remark cards. As many cards of information as is desired may be printed when the operation 'REMARK' is used. These remarks appear exactly as they are keypunched.

Several restrictions should be noted. The total number of storage locations available for matrices is 20,000, and no more than 100 matrices may be stored at any one time. If an attempt is made to load or generate a matrix with the same name as a previously defined matrix the previous matrix is destroyed before the operation is executed.

Tape input and tape output are determined by the use of the operations WRITE, READ, REWIND, ISTIN and ISTOUT. Printed output is determined solely by remark cards and the operator print. Only two tape units may be used: N1=1 and N1=2.

Most IST files generated by other BCTRY components can be input from IST or output IST by the operations ISTIN and ISTOUT. Following is a list of files accessible according to the type of file (N1=1 for type 1 files N1=2 for type 2 files):

TYPE 1	TYPE 2	NOT ACCESSIBLE
MEANS1	CORRM1	REORD1
STDEV1	CORRM2	REFLX1
DIAGV1	COVAR1	FSCOR1
UFACT1		OCLUS1
RFACT1		VNAMS
BASIS1		2SENS1
MEANS2		VAREN1
STDEV2		IDFILE
WEIGH1		CLUST1
2MEAN1		REFLX1
2STDV1		

The fields on the control cards are organized as follows:

OPERATION (PACK LEFT)	CALLING SEQUENCE (PACK RIGHT)									
	MATRICES AND IST FILES				MATRIX SIZE OR SYMMETRY				CONSTANTS	
	A	B	C	D	N1	N2	N3	N4	C1	C2
1-6	7-12	13-18	19-24	25-30	31-36	37-42	43-48	49-54	55-60	61-66

Example 1. Multiple correlation between dependent clusters and the
K most nearly independent pivotal clusters (inter-domain).
Special symbols: NI equals 3 predictor independent clusters.
ND equals 4 criterion dependent clusters.
R's equals squared multiple R.

(The SMIS executive card is preceded by the BCTRY run that produces the IST files or by a TAKE deck from that run.)

The control cards for example 1 are:

```
/SMIS
ISTIN      RBASIS1            000001
PRINT      R
RMVSM      R    RI            000001000001000003000003
PRINT     RI
INVERT    RI
PRINT     RI
RMVSM      R    RD            000001000004000003000004
PRINT     RD
TRANS     RD   RDT
PRINT    RDT
MULT     RDT    RI    TS
PRINT     TS
MULT      TS    RD   R2M
PRINT    R2M
RMVDG    R2M    R2
PRINT    R2
TIME
RETURN
```

Example 2. Same multiple R but for correlations between raw cluster scores given in CSA2 printout.

The same control cards are used except replace ISTIN by a load operation card followed by data cards of raw matrix with unities in the diagonal. The load card contains 'R' in col. 12 for the R matrix followed by the decimal card format of type (9F8.4); then the number of rows and columns in fields N1 and N2, respectively. (Then come the data matrix followed by ISTIN, etc.)

Example 3. Communality of each variable estimated by squared multiple R with all the other variables (a general squared multiple R procedure).

(Before calling SMIS, the IST files must contain CORRM1 and also DVP71, i.e., unities in the diagonal.)

```
/SMIS
ISTIN    H2TDIAGV1                 1
PRINT    H2T
TRANS    H2T    H2
ISTIN      RCORRM1                 2
STODG      R    H2
PRINT      R                       1
Guilford correlation matrix
SYMINV     R
RMVDG      R     T
INVEL      T
SUB       H2     T
PRINT     H2                       1
     Communalities estimated by squared multiple correlations
TRANS     H2   H2T
ISTOUT   H2TDIAGV1                 1
RETURN
```

Incitation

Here are some examples of brain-teasers, or incitements to thought that illustrate how ingenuity in the use of BCTRY components and their IST files can solve some special, important problems. Others are suggested in the Introduction and Method sections of the different components in the Manual. If you can figure out how to solve each problem without looking at the answers you have a good grasp of the principles of the system. We invite users to send other examples (to either editor). We would like to circulate a compilation of incitations.

1. Test of cluster hypotheses.

Question. An investigator claims that there are exactly three kinds of variables in a battery and he specifies the kind for each variable. How can this hypothesis be submitted to a test in a sample of observations on this battery?

Answer. Using preset CC5-CSA2-SPAN2 one determines the degree to which the hypothesized clusters account for the sum of squared correlations and how consistent the clusters are with respect to a unitary domain (domain validity). In order to evaluate non-hypothesized clusters call FAST(residual) after SPAN2 and factor the residual matrix with a standard CC5-CSA2-SPAN2 cycle. (DEB)

2. Scores from 'outside' programs used on O-analysis.

Question. How do you perform an O-analysis on scores obtained from sources outside BCTRY, such as other programs to estimate missing data, to compute standard scores, to compute other composites than FSCOR1, etc.?

Answer. Introduce these scores directly into EUCO via GIST. Since STDEV2, MEANS2, and WEIGHT1 are not actually used in EUCO but are merely printed out there insert dummy values in these GIST input packages. (JC)

3. Comparative longitudinal studies with variable overlap of cases at the different time periods.

Question. How can you determine the consistency of the cluster or factor structure of a given set of variables in a longitudinal analysis, when numerous subjects have dropped out or new ones added across time?

Answer. Run full cycle cluster analyses of each set of data collected at the different points in time. Determine the best set of dimensions per point-in-time, call COMP1 for each analysis, then make the comparative analysis with the sequence COMP2-DVP42-CC5-CSA2-SPAN2.

This analysis will show the similarity of the dimensions at the different periods despite the change of subjects. (JC)

4. Domain prediction of a criterion.

Question. If you have a single criterion variable and wish to evaluate how it correlates with each of the domains in a battery of NV-1 variables without the criterion influencing the definition of the clusters in the factoring process how would you proceed?

Answer. Using DAP2 input all NV variables (one criterion and NV-1 other variables 71/2). After COR2 call SLEP1 to suppress the criterion in factoring. Call DVP71-CC5-SLEP2-CSA2-SPAN2. The oblique factor coefficients of the criterion with each of the cluster domains are the desired correlations. (DEB)

5. Traditional item-analysis.

Question. If you accept the total score on an achievement test as a 'criterion', how would you determine the item-difficulty and the item-validity of each item of a large pool of items that compose the test?

Answer. Punch the total score as a new variable on each subject's data card that contains the items as column variables. Then apply this sequence: DAP2-COR2-DVP71(unities in diagonals)-GIST (REFLX1 on one 'cluster' defined by the one criterion variable)-CSA2. In DAP2, the means are the item-difficulties. In CSA2 the 'oblique factor coefficients' are the item validities, i.e., the correlations of the items with the total-score criterion. With more than 119 items, run the sequence in lots of 119 items. (Note: if the criterion is defined by not more than 20 items, you won't need to key-punch the total score but can, instead, define it as a cluster in CSA2 and preset REFLX1 as the cluster.) (RCT)

6. Determining the items that differentiate contrasting criterion groups.

Question. Having a large pool of NV dichotomous items scored on two contrasting groups X and Y how can you graphically and accurately determine which items favor one group over the other, and which do so at a 'Z' value deemed significant?

Answer. Step 1: for each criterion group, treat the items as NV column variables, punching 1 for yes and 0 for no for each of the NS individual subjects (cards). Run the group X deck on DAP2. The means on the DAP2 printout are the vector of percent yes on the items, i.e., P(X) values of the Y group.

Step 2: on a new deck, treat groups X and Y as two column variables. Treat now the NV items as individuals (cards). Keypunch the per cent yes values, P(X) and P(Y), in the X and Y fields on the

NV cards. Call RSCAT for the scattergram between the X and Y groups. The items are 'individuals' in the scatter whose names are listed in the spotter. Items above the diagonal favor the Y group, those below favor the X. Set a level of significance denoted by plus or minus 'Z' on a unit normal distribution (i.e., the two-tailed test). Then in the formula for the significance ratio (Walker and Lev, Statistical inference, 1953, equation 3.12) solve for the two ordinate values of P(Y) for each of several P(X) points along the abscissal scale. Draw a line through the points above the diagonal, and one through those below. All items outside the two curves are significant at the stipulated level of significance. You can include as many as 5000 items if there are two groups X and Y (see the restriction in RSCAT), 3333 if three groups, and so on. (RCT)

7. Cluster analysis of variables that maximizes group differences.

Question. Given a BIGNV problem involving numerous variables and two contrasting groups (e.g., experimental vs. control) how can you best factor a set of the variables selected from the full supply in order to highlight differences between the groups?

Answer. Use DAP2 on each group in order to find their means and sigmas. Pick out the set of not more than 90 variables with the most significant differences. Combine the group and run a full cycle key cluster analysis. Score the individuals on FACS and test the dimension scores of the two groups for statistical significance. (JC)

8. Principal component regression analysis.

Question. How can one compute this analysis?

Answer. The calling sequence is as follows: DAP2(reorder the input variables by putting the dependent variable last)-COR2-SLEP(sleep the dependent variable)-DVP71-FALS for PFA(one factoring)-FAST-GYRO (call varimax, say)-SMIS, with the following general sequence (a detailed control deck can be procured from project CAP):

Extract the vector of correlations of the dependent variable.
Input UFACT1 (from PFA).
Compute the square roots and squares of principal factor contributions.
Compute the squared multiple R of the dependent with independents.
Compute standardized and original regression coefficients.
Compute the regression constant of the regression equation.
Compute the square roots and squares of the varimax factor contributions.
Compute the standard error of estimate.

The above can be run for several dependents in one pass. (JB,JW,WM)

TIME CHARTS, PAGE ESTIMATES, CARD COUNTS

INTRODUCTION

AT THE COMPUTER CENTER, BERKELEY, PROBLEMS ARE TERMINATED BY MONITOR BEFORE COMPLETION IF THEY OVERRUN ESTIMATED TIME, PAGE OUTPUT, OR CARD OUTPUT AS PUNCHED BY THE ANALYST ON THE JOB CARD. JAMES CAMERON HAS PREPARED THE TABLES BELOW BY MEANS OF WHICH THE ANALYST CAN QUITE ACCURATELY ESTIMATE FOR HIS JOB CARD THE OUTSIDE LIMITS OF TIME AND PAGES REQUIRED BY HIS PROBLEM. HE HAS PREPARED THESE TABLES EMPIRICALLY FROM ACTUAL VALUES ON A LARGE NUMBER OF REAL PROBLEMS. TIME AND PAGES ARE LARGELY A FUNCTION OF NV, THE NUMBER OF VARIABLES IN THE PROBLEM. HE HAD, OF COURSE, NO OBSERVATIONS FROM ALL VALUES OF NV, SO HE GRAPHICALLY INTERPOLATED TO GET THE TABLED VALUES. THE PAGE COUNT TURNS OUT TO BE MAINLY A FUNCTION OF TIME, HENCE HE HAS DERIVED A SIMPLE FUNCTION OF TIME THAT GIVES THE PAGE ESTIMATES.

FOR MANY PROBLEMS EXAMINED, THE TABLED OUTSIDE ESTIMATES HAVE PROVED TO BE ACCURATE. IF ANY ANALYST FINDS AN ESTIMATE TOO LOW, PLEASE LET PROJECT CAP KNOW SO THAT THE TABLES CAN BE CORRECTED. THE SYSTEM TAPE WILL SOON GO OFF TAPE STORAGE ONTO DISC STORAGE, A FEATURE WHICH SHOULD CUT REAL TIMES DOWN SOMEWHAT.

OUTSIDE TIME LIMITS. TABLE 1 GIVES OUTSIDE ELAPSED TIMES BY BC TRY COMPONENTS AND CUMULATIVELY FOR STANDARD FULL-CYCLE CC ANALYSIS. STANDARD NC ANALYSIS WILL BE LESS. ACCESS IS INCLUDED IN ALL RUNS, SO BE SURE TO INCLUDE ACCESS TIME OF 3 MINUTES IN THE TOTAL ESTIMATE. WE CANNOT YET ESTIMATE COR3, SYDA, OR SYRN AND INVITE ANALYSTS WHO USE THEM TO.

APPROXIMATE TIME CONSTANTS OF OTHER COMPONENTS. SOME COMPONENTS ARE NEGLIGIBLY AFFECTED BY NV. THESE ARE GIVEN IN TABLE 2.

PAGE ESTIMATES AND CARD COUNTS. TABLE 3 GIVES ESTIMATES OF PAGES AND OF DECIMAL CARD OUTPUT IN A GIVEN PROBLEM.

CUMULATIVE TIME FOR A STANDARD FULL CYCLE CC ANALYSIS INCLUDES THE FOLLOWING.. ACCESS, DAP, COR2, DVP50, CC5, CSA, SPAN. FOR LARGE VALUES OF NV AND LARGE VALUES OF NS THE TIMES SHOWN FOR COR2 ARE UNDERESTIMATED BY A FACTOR OF 2.5.

THESE ESTIMATES WERE MADE IN JANUARY 1964 ON IBM 7090, USING A SYSTEM TAPE.

TABLE 1. MINUTES FOR COMPONENT PROGRAMS IN V-ANALYSIS AND O-ANALYSIS, WITH CUMULATIVE TIMES FOR STANDARD ANALYSES

NO. VAR.	DAP	COR2	COR3	DVP50	DVP60	CC5	NC2	(N)CSA	SPAN	CUM.TIM
5	.16	.10		.12	.20	.23	.12	.32	.20	4.13
10	.18	.11		.13	.38	.28	.17	.33	.23	4.26
15	.22	.12		.14	.58	.33	.22	.34	.27	4.42
20	.24	.14		.15	.82	.38	.28	.36	.30	4.57
25	.27	.16		.16	1.07	.45	.34	.37	.33	4.74
30	.31	.19		.17	1.35	.52	.41	.38	.36	4.93
35	.34	.22		.18	1.66	.62	.48	.39	.39	5.14
40	.38	.25		.19	2.01	.73	.56	.41	.42	5.38
45	.42	.28		.20	2.38	.83	.63	.42	.46	5.61
50	.46	.32		.21	2.76	.96	.69	.44	.49	5.88
55	.50	.37		.22	3.19	1.10	.77	.46	.52	6.17
60	.54	.42		.23	3.62	1.23	.84	.48	.56	6.46
65	.60	.47		.24	4.03	1.36	.92	.50	.58	6.75
70	.66	.53		.25	4.50	1.54	1.00	.52	.62	7.12
75	.72	.59		.26	4.97	1.70	1.09	.54	.65	7.46
80	.78	.65		.26	5.50	1.88	1.17	.57	.68	7.82
85	.85	.73		.27	6.04	2.06	1.25	.60	.72	8.23
90	.93	.82		.28	6.62	2.31	1.34	.64	.75	8.73
95	1.02	.92		.29	7.22	2.58	1.47	.68		8.49
100	1.10	1.03		.30	7.87	2.90	1.64	.72		9.05
105	1.18	1.14		.31	8.57	3.27	1.90	.77		9.67
110	1.28	1.25		.32	9.22	3.70	2.20	.84		10.39
115	1.39	1.36		.33	10.02	4.19	2.54	.92		11.19
120	1.50	1.48		.34	10.90	4.65	2.93	1.01		11.98

NO. VAR.	FACS	EUCO	PRIN	APA	CANON	FAST	GYRO (QRT-MAX)	SLEP1	SLEP2	COMP1	COMP2
5	.37	.50	.50	.52	.60	.40	.77	.25	.25	.30	.30
10	.37	.50	.73	.77	.93	.65	.78	.27	.28	.32	.38
15	.38	.50	.95	1.01	1.25	.80	.79	.29	.30	.33	.45
20	.38	.50	1.20	1.28	1.60	.87	.80	.31	.33	.35	.52
25	.38	.50	1.42	1.52	1.92	.93	.81	.33	.35	.36	.60
30	.39	.50	1.69	1.81	2.29	.97	.82	.34	.38	.37	.68
35	.39	.50	1.93	2.07	2.60	1.02	.83	.35	.40	.38	.76
40	.40	.50	2.19	2.35	2.99	1.04	.84	.37	.43	.40	.83
45	.40	.50	2.40	2.58	3.30	1.06	.85	.38	.45	.42	.92
50	.41	.50	2.67	2.87	3.67	1.07	.86	.40	.48	.43	1.00
55	.42	.50	2.89	3.11	3.99	1.09	.87	.41	.51	.45	1.08
60	.42	.53	3.17	3.41	4.37	1.10	.88	.43	.54	.47	1.16
65	.43	.53	3.39	3.65	4.69	1.12	.89	.44	.56	.48	1.23
70	.43	.53	3.63	3.91	5.03	1.13	.90	.45	.59	.50	1.31
75	.44	.54	3.88	4.18	5.38	1.14	.91	.47	.62	.51	1.39
80	.44	.57	4.13	4.45	5.73	1.15	.92	.48	.64	.53	1.48
85	.45	.59	4.37	4.71	6.07	1.16	.93	.50	.67	.54	1.56
90	.45	.62	4.60	4.96	6.40	1.17	.94	.52	.69	.56	1.62
95	.46	.65	4.83	5.21	6.73	1.17	.95	.53	.72	.57	1.70
100	.47	.68	5.07	5.47	7.07	1.18	.96	.55	.74	.59	1.78
105	.47	.71	5.31	5.73	7.41	1.18	.97	.57	.77	.60	1.86
110	.48	.75	5.54	5.98	7.74	1.19	.98	.58	.79	.62	1.92
115	.48	.78	5.78	6.24	8.08	1.20	.99	.59	.82	.63	2.00
120	.48	.81	6.02	6.50	8.42	1.20	1.00	.60	.85	.65	2.08

TABLE 2. APPROXIMATE TIME CONSTANTS OF OTHER COMPONENTS

COMPONENT	TIME (MIN.)
ACCESS	3.00
GYRO (VARIMAX)	ADD .50 TO QRTMAX (Q.V.)
GIVE	.15
TAKE	.15
GIST	.15
REDE	.15

THESE VALUES ARE FOR TAPE ACCESS AND SIX TIMES GREATER THAN THE ACCESS TIMES FOR DISK ACCESS.

TABLE 3. PAGE ESTIMATES AND CARD COUNT

PAGE ESTIMATES.. TECHNICALLY, THE 'PAGES' ESTIMATE REFERS TO THE TOTAL NUMBER OF LINES IN THE PRINTOUT, DIVIDED BY 60 (THE TOTAL LINES POSSIBLE ON A PAGE). CAMERON'S FORMULA IS..

TOTAL PAGES = CUMULATIVE NUMBER OF MINUTES (FROM TABLES 1 AND 2) TIMES 15 PLUS 5.

CARD COUNT.. THE CARD COUNT REFERS ONLY TO DECIMAL CARDS OUTPUT (MAINLY BY GIST, COMP1, SYDA) AND NOT TO BINARY CARDS (AS FROM GIVE). IF THIS NUMBER EXCEEDS 100, THE JOB IS TERMINATED BY MONITOR UNLESS THE CARD COUNT NUMBER IS KEY-PUNCHED ON THE JOB CARD. THE ANALYST KNOWS THE NUMBER IN THE OUTPUT CARDS FROM GIST. FOR COMP1 AND SYDA IT IS ABOUT .5 NV. THE CARD COUNT PUNCH ON THE JOB CARD IS THE CUMULATIVE TOTAL OVER ALL COMPONENTS THAT WILL YIELD DECIMAL CARDS.

REFERENCE AND AUTHOR INDEX

Bailey, D. The role of psychological distance in human discrimination learning. Ph.D. thesis, 1960. Univ. of Calif. Library, Berkeley

Broverman, D.M. Effects of score transformations in Q and R factor analysis techniques. Psych. Rev., 1961, 68, 68-80.

Burt, C. Correlation between persons. Brit. J. Psychol., 1937, 28, 59-96.

The factors of mind. London: Univ. of London Press, 1940.

The factorial study of temperamental traits. Brit. J. Psychol. Stat. Sec., 1948, 1, 178-203.

Cattell, R.B. Psychological measurement: ipsative, normative, and interactive. Psychol. Rev., 1944, 51, 292-303.

Factor analysis. New York: Harper, 1952.

Cooley, W., + Lohnes, P. Multivariate procedures for the behavioral sciences. John Wiley + Sons, 1962.

Ezekiel, M. + Fox, K. Methods of correlation and regression analysis. New York: John Wiley and Sons, 1959 (3rd ed.).

Fisher, R.A. Mathematical foundations of statistics. Phil. Trans. Roy. Soc., A222, 1921.

Ghiselli, E. E. Theory of psychological measurement. New York: McGraw-Hill, 1964.

Guilford, J.P. Fundamental statistics in psychology and education. New York: McGraw-Hill, 1950(2nd. ed.).

Psychometric methods. New York: McGraw-Hill, 1954 (rev.).

Harman, H. On the rectilinear prediction of oblique factors. Psychomet., 1941, 6, 29-35.

Modern factor analysis. Chicago: Univ. of Chic. Press, 1960.

Holzinger, K.J. + Swineford, F. A study in factor analysis: the stability of a bi-factor solution. Suppl. Educ. Monogr., No. 48. Chicago: Dept. Educ., Univ. of Chicago, 1939.

Hotelling, H. Analyses of a complex of statistical variables. J. Educ. Psychol., 1933, 24, 417-441, 498-520.

Relations between two sets of variates. Biom., 1936, 28, 321-377.

Kaiser, H. The varimax criterion for analytic rotation in factor analysis. Psychomet., 1958, 23, 187-200.

+ Caffrey, J. Alpha factor analysis. Psychometr., 1956, 30, 1-14.

Kelley, T.L. Essential traits of mental life. Harvard Stud. in Educ., 26. Cambridge, Mass. Harvard Univ. Press 1935.

Fundamentals of statistics. Harvard Univ. Press, 1947.

Lawley, D. The estimation of factor loadings by the methods of maximum likelihood. Proc. Roy. Soc. Edinburg, A, 1940, 60, 64-82.

Lederman, W. On a shortened method of estimation of mental factors by regression. Psychometr., 1939, 4, 109-116. Also Nature, 1938, 141, 650.

McQuitty, L.L. Hierarchical syndrome analysis. Educ. + Psychol. Measmt., 1960, 20, 293-304.

Meehl, P.E. Configural scoring. J. Consult. Psychol., 1950, 14, 165-171.

Meredith, W. Rotation to achieve factorial invariance. Psychometr. (in press).

Neuhaus, J., + Wrigley, C. The quartimax method: an analytic approach to orthogonal simple structure. Brit. J. Statis. Psychol., 1954, 7, 81-91.

Pearson, K. On lines and planes of closest fit to systems of points in space. Phil. Mag., 1901, 6th ser., 559-572.

Peters, C. + Van Voorhis, W.R. Statistical procedures and their mathematical bases. N.Y. McGraw-Hill, 1940.

Rao, C.R. Estimation and tests of significance in factor analysis. Psychometr., 1955, 20, 93-111.

Spearman, C. Correlations of sums and differences. Brit. J. Psychol., 1913, 5, 417-426.

Stephenson, R. W. Originality and affect. Ph.D. thesis, 1962. Univ. of Calif. Library, Berkeley.

Stevenson, W. Correlating persons instead of tests. Char. and Person., 1935, 4, 17-24.

Thomson, G. The factorial analysis of human ability. Boston: Houghton Mifflin Co., 1939 (5th. ed. 1951).

On estimating oblique factors. Brit. J. Psychol. Stat. Sec., 1949, 2 (1), 1-2.

Thurstone, L.L. The vectors of mind. Chicago: Univ. of Chicago Press, 1935.

Multiple factor analysis. Chicago: Univ. of Chicago Press, 1947.

Tryon, R.C. Cluster analysis. Ann Arbor, Mich. Edwards Bros., 1939.

Identification of social areas by cluster analysis. Univ. of Calif. Publ. in Psychol., 1955, 8, no. 1, 1-100. Berkeley: Univ. of Calif. Press.

Reliability and behavior domain validity: reformulation and historical critique. Psychol. Bull., 1957 (A), 54, 229-249.

Communality of variable: formulation from cluster analysis. Psychometr., 1957 (B), 22, 241-259.

Cumulative communality cluster analysis. Educ. and Psychol. Measmt., 1958 (A), 18, 3-35.

General dimensions of individual differences: cluster analysis vs. factor analysis. Educ. and Psychol. Measmt., 1958 (B), 18, 477-495.

Domain sampling formulation of cluster and factor analysis. Psychometr., 1959, 24, 113-135.

Basic unpredictability of individual responses to discrete stimulus presentations. Paper given at Psychonomics Society, 1962 (ditto).

Theory of the BCTRY system: statistical theory, 1964a. (Library version in ditto)

The component programs of the BCTRY system, 1964b. (Library version in ditto)

Tryon, R. + Bailey, D. The BCTRY computer system of cluster and factor analysis: multivariate behavioral research, (in press)

Tryon, R. + Stein, K. + Chu, C. Item-clusters and profile types derived from responses of adults to the 556 MMPI items (symposium report at WPA, Hawaii, 1965).

Tucker, L.R. A method for synthesis of factor analysis studies. Person. Res. Sect. Rept., no. 984. Washington, D. C., Dept. of the Army, 1951.

Wolfe, John H. Object cluster analysis of social areas. M.A. thesis, 1963. Univ. of Calif. Library, Berkeley.

Wrigley, C., + Neuhaus, J. A refactorization of the Burt-Pearson matrix with the ORDVAC electronic computer. Brit. J. Psychol. Stat. Soc., 1952, 5, 105-108.

DST AND IST INPUT-OUTPUT (I/O) TABLES

I/O for DST

Programs which write DST:

data processor program, DAP
Euclidean distance program, EUCO

Programs which read DST:

correlation program, COR2
correlation/covariance program, COR3
cluster and factor scoring program, FACS

I/O for IST (see the tables on the pages following)

SYMBOL IN TABLE	MEANING
I	Input for all control options.
(I)	Input for some control options.
O	Output for all control options.
(O)	Output for some control options.

All files may be input and/or output by GIST and GIVE.

Note: you can always find out what files are on IST by 'dumping' them. Call '/BTD' in cols. 1-4. All files are printed out, but not erased.

	IST FILES						
PROGRAM	IDFILE	VNAMS1	VSUMS1	MEANS1	VAREN1	STDEV1	CORRM1
DAP	O	O	O	O	(O)	O	
COR2	I	I	I	I		O	O
COR3	I	I		I	(I)(O)		(O)
REDE	I	I					I
RLIST	I	I			I		I
SLEP1	I	I,O					I,O
DVP	I						I
FALS	I						I
NC2	I						I
CC5	I						I
SLEP2	I	I,O					I,O
GYRO	I						
NCSA	I						(I)
CSA	I						(I)
FAST	I						(I)(O)
SPAN	I						
FACS	I			I		I	I
EUCO	I	O		O		O	
COMP1	I	(I)					
SYDA	I					I	
COMP2	O	O					O
SYRN							

	IST FILES						
PROGRAM	2SENS1	2MEAN1	2STDEV1	COVAR1	REORD1	CORRM2	DIAGV1
DAP							
COR2							
COR3	(O)	(O)	(O)	(O)			
REDE							
RLIST	I	I	I				
SLEP1					O	O	
DVP							O
FALS							I,O
NC2							(I),O
CC5							I,O
SLEP2					I	I	O
GYRO							
NCSA							
CSA							(I)
FAST							
SPAN							
FACS							(I)
EUCO							
COMP1							
SYDA							
COMP2							
SYRN							

	IST FILES						
						MEANS2	STDEV2
PROGRAM	UFACT1	CLUST1	REFLX1	RFACT1	BASIS1	FSCOR1	WEIGH1
DAP							
COR2							
COR3							
REDE							
RLIST							
SLEP1							
DVP							
FALS	O						
NC2	O	O	O				
CC5	O	O	(I),O				
SLEP2	I,O	(I)(O)	(I)(O)				
GYRO	I			O	O		
NCSA	(I)		(I)	O	O		
CSA	(I)		I	O	O		
FAST	I						
SPAN	(I)			(I)			
FACS	(I)		(I)	(I)	(I)	O	O
EUCO						I	I
COMP1		(I)		I			
SYDA	I						
COMP2							
SYRN							

ONAMS1 - a new IST file

Previously we had not provided for printing out the names of subjects, an awkward lack in O-analysis. The problem has now been solved by generating the ONAMS1 file. The components that output (write) ONAMS1 or input (read) the file are as follows:

DAP2 reads ONAMS from the data cards and outputs them to the ONAMS1 file. If no ONAMS are punched on the data cards, DAP2 generates a standard set of names.

RSCAT generates the ONAMS1 file as a standard set of names if it does find an ONAMS1 file present.

EUCO now reads the ONAMS1 file and outputs these names on the matrix of Euclidean distances (replacing the word 'object' previously uniformly printed). If it finds no ONAMS1 file it generates a standard set of ONAMS. EUCO also now outputs ONAMS1 into the VNAMS1 file so that when the user goes from EUCO to COR2 in O-analysis the subjects' names are represented as variables.