MANAGEMENT APPLICATIONS in APL

Edited by EPHRAIM R. McLEAN and JAMES W. SCHENCK

Graduate School of Management
UCLA
MANAGEMENT APPLICATIONS in APL

Edited by
EPHRAIM R. McLEAN
and
JAMES W. SCHENCK

Graduate School of Management
UCLA
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Program Overview.</td>
<td>1</td>
</tr>
<tr>
<td>2.0 Program Participants.</td>
<td>7</td>
</tr>
<tr>
<td>3.0 Participants' APL Applications.</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Armitage, Howard</td>
<td>11</td>
</tr>
<tr>
<td>&quot;The Accounting Cycle&quot;</td>
<td></td>
</tr>
<tr>
<td>3.2 Baker, Dennis</td>
<td>21</td>
</tr>
<tr>
<td>&quot;Simulation of the Birth and Death of Firms in a Competitive Industry&quot;</td>
<td></td>
</tr>
<tr>
<td>3.3 Brown, Lawrence D.</td>
<td>41</td>
</tr>
<tr>
<td>&quot;Single Asset Depreciation Methods&quot;</td>
<td></td>
</tr>
<tr>
<td>3.4 Case, Fred</td>
<td>47</td>
</tr>
<tr>
<td>&quot;Real Estate Valuation Using the Direct Capitalization Method&quot;</td>
<td></td>
</tr>
<tr>
<td>3.5 Chai, John</td>
<td>55</td>
</tr>
<tr>
<td>&quot;Simulation of Random Samples and Sampling Distributions of Various Statistics&quot;</td>
<td></td>
</tr>
<tr>
<td>3.6 Dauner, Jack</td>
<td>67</td>
</tr>
<tr>
<td>&quot;Setting the Size of the Sales Force&quot;</td>
<td></td>
</tr>
<tr>
<td>3.7 Harpell, John L. Jr.</td>
<td>71</td>
</tr>
<tr>
<td>&quot;Simulation of a 3-Stage Order-Shipment System&quot;</td>
<td></td>
</tr>
<tr>
<td>3.8 Hickman, Edgar</td>
<td>81</td>
</tr>
<tr>
<td>&quot;Multivariate T-Square Test&quot;</td>
<td></td>
</tr>
<tr>
<td>3.9 Mason, R. Harold</td>
<td>87</td>
</tr>
<tr>
<td>&quot;Circulation of Disability Settlement Awards&quot;</td>
<td></td>
</tr>
<tr>
<td>3.10 Miller, Ronald D.</td>
<td>95</td>
</tr>
<tr>
<td>&quot;Linear Programming with Mixed Constraints&quot;</td>
<td></td>
</tr>
<tr>
<td>3.11 Park, John</td>
<td>107</td>
</tr>
<tr>
<td>&quot;Interactive Effects of the Marginal Propensity to Consume and the Investment Accelerator on the Production of Income&quot;</td>
<td></td>
</tr>
<tr>
<td>3.12 Sanford, Donald G.</td>
<td>115</td>
</tr>
<tr>
<td>&quot;Network Analysis&quot;</td>
<td></td>
</tr>
</tbody>
</table>
3.13 Schriber, Thomas
"Interactive Optimization of a Production System".................. 127

3.14 Sprague, Ralph H. Jr.
"Interactive Inventory Simulator"................................. 155

3.15 Stricklin, Wilma
"Financial Ratio Calculations"................................. 173

3.16 Thielges, Bernard
"Computation of a Price Index by the Method of the Weighted Geometric Mean"................................. 177

4.0 UCLA APL Library Functions............................................. 185

5.0 Outline of the Course.............................................................. 201

5.1 Class Schedule and Reading Assignments............................... 203

5.2 Class Handouts.............................................................. 213
  5.2.1 A Guide to Writing APL Programs......................................... 215
  5.2.2 Signing-On to the UCLA APL System................................. 221
  5.2.3 An APL Bibliography.................................................. 227
  5.2.4 "Things Go Better with APL"
      Claire Stegmann, Think, May/June 1975.............................. 255

5.3 APL Programming Assignment Instructions..................................... 261

6.0 Program Administration and Promotional Materials............................ 265

6.1 Program Announcement Brochure......................................... 267

6.2 Invitation Letter to Business School Deans.......................... 273

6.3 Publicity.............................................................. 277

6.4 Application Questionnaire and Cover Letter............................... 281

6.5 Letter of Acceptance.......................................................... 291

6.6 Program Evaluation Questionnaire..................................... 297

6.7 Post Conference Follow-Up Letter.................................. 303
1.0
PROGRAM OVERVIEW

In recognition of the growing importance of interactive computing in support of management education, the IBM Corporation agreed to sponsor a two-week program on the subject in the summer of 1975. Modelled after the highly successful program at the University of Michigan, which introduced business school educators to the use of FORTRAN, this program was aimed at accomplishing the same goal, this time with the interactive computer language APL. Approximately two dozen management school professors were to be invited to come to the Graduate School of Management (GSM) at UCLA and to be involved in an intensive learning experience.

Basically, the class had two objectives. The first was to gain some familiarity and confidence in the use of the computer and APL. The second was to write a substantial program in APL that would be useful in teaching each faculty member's own discipline (e.g., accounting, finance, production, etc.). The programs that comprise a major part of this book are the results of this second objective.

Naturally, some of these programs are more comprehensive than others; and, in fact, there were a few of the participants that set such ambitious goals for themselves that they were unable to complete their programs in the two-week time period. Then, when they returned home, the press of other activities crowded out their ability to finish their programs. This explains why a few of the participants' programs are missing from this book. However, given the fact that most of the faculty members had never written a computer program before beginning the course, the programs that do appear in the following pages are an impressive testimonial to their hard work and commitment.

Because a great deal of planning and arrangements went into the conduct of the course, it might be of interest to the reader to have a chronology of the events leading up to its offering.

Upon receiving notification of IBM's decision to support the program on Interactive Computing for Management Educators in December of 1974, plans were begun immediately for the conduct of the program. In January of 1975, announcement brochures were prepared (See Section 6.1), describing the program in detail and calling for interested parties to apply for
admission. The contents of this brochure were patterned very closely after the description of the program included in the original grant request proposal sent to IBM. This brochure was then sent to the deans of all of the member schools of the American Assembly of Collegiate Schools of Business (AACSB). It was accompanied by an individually typed cover letter (Section 6.2) which was personally signed by GSM Dean Harold Williams. This more personalized approach was taken to assure that the letter, and the brochure, received a careful reading. In all, slightly over 500 letters were mailed out, each one containing at least two brochures (so that the dean receiving them could circulate them to two groups within the school at the same time).

The response to this mailing -- and to some publicity in the Computing Newsletter for Schools of Business (Section 6.3) -- was nothing short of phenomenal. By the time of the March 15th deadline stated in the brochure, 197 requests for applications had been received for the 24 openings in the program. Several more requests were received after the deadline, particularly from abroad, which are not included in this total.

To provide a basis for selection -- and, quite frankly, to weed out the idle curious -- a detailed six-page questionnaire (Section 6.4) was prepared and mailed to each of those expressing interest in attending. From this mailing, 105 completed questionnaires were received, representing a wide cross-section of schools, geographical distribution, and academic ranks. With only one opening for every four applicants, the selection process was quite difficult, and a second or third offering could have easily been filled with qualified individuals. Of particular interest is the fact that, of those applying, over 60 came from schools that either already had APL installed and running on their campus computer or were planning its installation within the coming year -- an impressive indication of the spread of APL.

After much careful study, 24 individuals were selected and notified (See Program Participants, Section 2.0). Those who could not be accommodated were also notified at the same time, thanking them for their interest and promising to contact them personally if another offering were to be held.
These 24 faculty members were from schools large and small, public and private. Two were from Canada and two were women (one a nun). All academic ranks were represented, and most areas of specialization within the field of business (e.g., accounting, economics, finance, marketing, production, etc.). Also of interest is the fact that while many people consider APL to be a uniquely IBM product, nearly half of the schools (9 out of 21) were running APL on other than IBM equipment.

Upon arrival, the participants were housed in the Holiday Inn of Westwood which allowed them the option of either taking their evening meals at the motel or in one of the many restaurants in nearby Westwood Village. This arrangement proved to be particularly well received, as was the opening reception and dinner which served as an important get-acquainted function. Breakfasts were at the Holiday Inn and lunches at the UCLA Faculty Center. Because the first Friday of the two-week program fell on July 4th, an Independence Day barbecue was held; and, with the exception of one or two individuals who had made previous commitments, it was attended by all of the participants and by the program faculty and staff.

In terms of the conduct of the course itself, the program was split into two groups, A and B. Although it was hoped that all of the participants would be of approximately the same skill level and background, the vagaries of selection were such that some of the attendees were able to move through the course material faster than others. Thus Group B became somewhat the "fast track" and Group A, the "slow track." The reading assignments, class schedule, class handouts, and programming assignments are shown in Sections 5.1, 5.2, and 5.3.

As might be expected, a certain amount of anxiety was caused by the requirement that each person was expected to produce a working, documented program of reasonable complexity by the end of the two weeks. These programs were of their own choosing, and the finished results are published in this monograph. Most of the participants were able to accomplish this task, although a few individuals undertook projects which were really too ambitious. However, even for those who failed to complete their programs, it is felt that the exercise still contributed markedly to their understanding of APL.
and interactive computing.

At the end of the program, a three-page questionnaire (Section 6.6) was passed out to each of the participants, asking them their comments and opinions on various parts of the program. So that frank reactions could be obtained, they were not required to sign the questionnaire or otherwise identify themselves; however, most chose to do so. Fourteen questionnaires were returned (and others gave their views orally) and the general reactions were highly favorable. On the key question "Your overall evaluation of the program as a whole -- (Poor) 1 2 3 4 5 6 7 (Outstanding)," the median response was 6. The following were typical comments:

"Great! I feel that I was really pressed into learning a great deal in a short time. I am enthused about APL and will begin to use it at my school."

"Outstanding."

"The program was extremely worth while to me. While the work was ample, I feel that's what we were here for so there should be no complaints in that regard. The program was well structured and very well presented."

Of particular interest are the comments (quoted in full) of Dr. Thomas Schriber of the University of Michigan, who for several years conducted the previously mentioned IBM-sponsored program on FORTRAN:

"Excellent, on a variety of counts:

1. Personal-professional (total immersion is the way to get on top of a highly detailed subject like this)
2. Personal-personal (a pleasure to work hard with intelligent, dedicated people, and to get to know these people)
3. Home curriculum (will be upgraded as a result of this course -- APL will get into our curriculum faster than otherwise)
4. Schools of Business overall (will get into the APL generation much sooner than otherwise; why not lead instead of lag?)"

Naturally, there were some specific suggestions about particular aspects of the course, the accommodations, or other administrative details, but they were mostly minor. By and large, both the written and oral comments were extremely positive.
As a final wrap-up to the program, a follow-up letter (Section 6.7) was sent to all participants, thanking them for their diligence and offering our assistance to those who had not yet completed their programs.

In the preparation of this monograph, every effort has been made to ensure that the APL programs, as submitted by the participants, are correct and "bug" free and will work as the sample output indicates. However, it is possible that some errors do still exist. If, in using the programs, such errors are encountered, it would be appreciated if the author of the program would be notified directly, with a copy of the correspondence sent to me here at UCLA.

In closing, I would like to acknowledge and thank the several individuals who made this program a success. On the instructional staff, Dr. Kenneth Siler, a fellow member of the GSM faculty, shared equally in the teaching load. James Schenck, a member of the staff of the GSM computing facility, assisted the participants in a number of ways and helped prepare the final versions of the programs for inclusion here. Dr. William Bergquist of IBM also was in attendance during much of the program, first as an observer and later as an assistant in the computer lab. Finally, Bill DeLone and Janice Durbin, graduate students in GSM helped in the preparation of the final manuscript. To all of these individuals, I extend my thanks and deep appreciation.

Ephraim R. McLean

June 1980
Los Angeles
PARTICIPANTS IN THE CONFERENCE ON INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS
June 29 - July 11, 1975
GRADUATE SCHOOL OF MANAGEMENT
UNIVERSITY OF CALIFORNIA, LOS ANGELES

Howard M. Armitage
Assistant Professor of Business
Department of Business
Wilfrid Laurier University
Waterloo, Ontario, Canada N2L 3C5

Fred E. Case
Professor of Human Land Economics
Graduate School of Management
University of California
Los Angeles, California 90024

Dennis M. Baker
Associate Professor of Accounting
and Information Systems
Faculty of Business Administration and Commerce
University of Alberta
486 Central Academic
Edmonton, Alberta, Canada T6K 1M6

John J. Chai
Associate Professor of Quantitative Methods
Department of Quantitative Methods
School of Management
Syracuse University
Syracuse, New York 13210

Sister Mary Seraphine Bennett
Professor of Quantitative Studies
School of Business
University of Baltimore
1420 North Charles Street
Baltimore, Maryland 21201

Jack R. Dauner
Associate Professor of Marketing
Department of Marketing
University of Akron
Akron, Ohio 44325

Lawrence D. Brown
Lecturer in Accounting and Economics
School of Management
State University of New York at Binghamton
Binghamton, New York 13901

Richard A. Elnicki
Associate Professor of Management
College of Business Administration
University of Florida
Gainesville, Florida 32605

Elwood S. Buffa
Professor of Operations Research and Operations Management
Graduate School of Management
University of California
Los Angeles, California 90024

John L. Harpell, Jr.
Assistant Professor of Management
College of Business and Economics
West Virginia University
Morgantown, West Virginia 26506

Kenneth J. Carey
Assistant Professor of Finance
Department of Finance
701 Business Administration Building
The Pennsylvania State University
University Park, Pennsylvania 16802

Edgar P. Hickman
Professor and Program Director of Management Science
College of Business Administration
University of South Carolina
Columbia, South Carolina 29208
R. Harold Mason  
Professor, International Business and  
Business Policy  
Graduate School of Management  
University of California  
Los Angeles, California 90403

Ronald D. Miller  
Assistant Dean  
College of Business  
Governors State University  
Park Forest South, Illinois 60466

William H. Murphy  
Chairman, Business Administration  
South Campus  
Salem State College  
Salem, Massachusetts 01970

John Park  
Chairman, Department of Economics  
and Business Administration  
Frostburg State College  
Frostburg, Maryland 21532

Avinoam Perry  
Assistant Professor of Decision Sciences  
Decision Sciences Department  
Graduate School of Management  
Northwestern University  
Evanston, Illinois 60201

Donald L. Rogoff  
Professor of Accounting and Finance  
School of Business and Management  
Pepperdine University  
Malibu, California 90265

Donald G. Sanford  
Assistant Professor and Assistant Dean  
College of Business and Public Administration  
Florida Atlantic University  
Boca Raton, Florida 33432

Thomas J. Schriber  
Professor of Management Science  
Graduate School of Business  
Administration  
University of Michigan  
Ann Arbor, Michigan 48104

John P. Shelton  
Professor of Finance  
Graduate School of Management  
University of California  
Los Angeles, California 90024

Barry W. Sheridan  
Assistant Professor of Business  
Division of Business  
Ithaca College  
1402 West Tower  
Ithaca, New York 14850

Ralph H. Sprague, Jr.  
Associate Professor of Quantitative  
Business Analysis  
College of Business  
University of Hawaii  
2404 Maile Way  
Honolulu, Hawaii 96822

Wilma D. Stricklin  
Professor of Management  
College of Business Administration  
Northern Arizona University  
Flagstaff, Arizona 86001

Bernard A. Thielges  
Assistant Professor of Management  
College of Business Administration  
Marquette University  
606 North 13th St.  
Milwaukee, Wisconsin 53233
3.0

PARTICIPANTS' APL APPLICATIONS
3.1

THE ACCOUNTING CYCLE:
An Interactive Approach

Howard M. Armitage
Assistant Professor in Accounting
School of Business and Economics
Wilfrid Laurier University
Waterloo, Ontario
Canada

DESCRIPTION

Introduction

In many respects, accounting is a highly repetitive process. In fact, the system that a student learns to follow at the outset of his accounting education consists mainly of the following steps:

- Journalizing
- Posting
- Trial Balance Preparation
- Adjustments
- Adjusted Trial Balance Preparation
- Statement Preparation
- Closing Entries

This sequence of steps is known as the accounting cycle and the user learns very early that the effect of any set of transactions on the final statements can be determined by following the above format.

Once this system has been illustrated and used and the student has grasped the repetitive nature of the cycle, further work with hand written accounting systems seems time consuming and only marginally useful. It is clearly advantageous to introduce a student to a system which, once the original entries have been made, can take over the repetitive tasks of posting,
ensuring equality, classifying items into revenue and capital accounts, and preparing final reports. This program has been written in response to such a need.

Instructions for Use

The user should specify his account names by using the function \texttt{CNAMES}. He must also establish a numbered set of accounts to correspond to the account titles. This should be done by using the function, \texttt{INIT}. \texttt{INIT}, when executed, requests that you enter the account numbers to be used. You should respond by typing the desired accounts separated by at least one blank. For example, 101 102 103 104 201 202 203 301 etc. If you are unable to fit all the account numbers desired on a single line, just enter what will fit on the line and then strike the return key. The \texttt{INIT} function will then ask if you have any more accounts to enter. If you answer \texttt{YES}, \texttt{INIT} will request that you enter the remaining group. \texttt{INIT} will continue requesting more accounts until you respond \texttt{NO} to the question, \texttt{ANY MORE}? At this time, the two needed global variables will be created.

For each entry, the user will be asked to enter an account number, debit or credit, and dollar amount. After each transaction, the equality of the input can be checked by typing \texttt{NEXT}. If the debits do not equal the credits, the system will respond by printing out the last transaction, indicating where the error lies. When the transactions have been completed, a final equality check can be made by typing \texttt{OUT}.

The program will then take over and print out an UNADJUSTED TRIAL BALANCE. The user will be asked if he has any adjustments to make, following which the system will print out an ADJUSTED TRIAL BALANCE, INCOME STATEMENT, and BALANCE SHEET for the period
covered by the transactions. The revenue and expense accounts will automatically be set back to zero.

The program can be used for illustrating the accounts of a service, merchandising, or manufacturing enterprise whose inventory accounts are kept on a perpetual basis. With slight modifications the system can accommodate periodic inventory systems.

At the end of each iteration, an updated balance sheet will be printed out. Should the user wish to start anew, he can employ the function \texttt{ZERO} which will reduce the balances in all his accounts to zero. The user can also begin the cycle with any opening balance he wishes by debiting the asset accounts by the desired amount and similarly crediting the liability and equity accounts. When the command \texttt{OUT} is typed, the system will return the desired opening position.

In the following section is an example of program output which has resulted from a small number of transaction data. The amount of detail produced in these reports depends solely on the number of accounts set up and the extent to which these accounts are utilized during the accounting period.

In addition to the functions listed in the Program Listing Section, the program also utilizes a number of public functions. Specifically these are as follows: \texttt{AKI}, \texttt{AYN}, \texttt{CENTER}, \texttt{COLNAMES}, \texttt{\textbackslash FMT}, and \texttt{IPI}. 

13
ACT
DO YOU NEED INSTRUCTIONS? Y
ENTER THE ACCOUNT NUMBER, THE DEBIT OR CREDIT, AND THE AMOUNT FOR EACH
PART OF A TRANSACTION.
WHEN YOU HAVE COMPLETED THE TRANSACTION ENTER NEXT
WHEN YOU HAVE COMPLETED ALL THE TRANSACTIONS ENTER OUT
THE SYSTEM WILL RESPOND BY PRINTING OUT A TRIAL BALANCE.
YOU WILL BE ASKED IF YOU HAVE ANY ADJUSTMENTS TO MAKE.
FOLLOWING THESE ADJUSTMENTS, AN ADJUSTED TRIAL BALANCE, INCOME STATEMENT, AND
BALANCE WILL BE PRINTED.
IF YOU WISH TO STOP THE PROGRAM ENTER STOP UNDER ACCOUNT NUMBER.
ACCOUNT NUMBER?
•: 102
DEBIT OR CREDIT?
D
AMOUNT?
•: 3000
ACCOUNT NUMBER?
•: 201
DEBIT OR CREDIT?
C
AMOUNT?
•: 2000
ACCOUNT NUMBER?
•: OUT
DEBITS DO NOT EQUAL CREDITS
ACCT DB CR
102 3000 0
201 0 2000
ACCOUNT NUMBER?
•: 201
DEBIT OR CREDIT?
C
AMOUNT?
•: 1000
ACCOUNT NUMBER?
•: OUT
UNADJUSTED TRIAL BALANCE
**UNADJUSTED TRIAL BALANCE**

**MATRIX**

<table>
<thead>
<tr>
<th>ACCT</th>
<th>DEBITS</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>16900.00</td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>12000.00</td>
<td>0.00</td>
</tr>
<tr>
<td>103</td>
<td>5000.00</td>
<td>0.00</td>
</tr>
<tr>
<td>104</td>
<td>10000.00</td>
<td>0.00</td>
</tr>
<tr>
<td>105</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>201</td>
<td>0.00</td>
<td>11000.00</td>
</tr>
<tr>
<td>202</td>
<td>0.00</td>
<td>400.00</td>
</tr>
<tr>
<td>203</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>204</td>
<td>0.00</td>
<td>800.00</td>
</tr>
<tr>
<td>205</td>
<td>0.00</td>
<td>20000.00</td>
</tr>
<tr>
<td>206</td>
<td>0.00</td>
<td>3400.00</td>
</tr>
<tr>
<td>301</td>
<td>0.00</td>
<td>11000.00</td>
</tr>
<tr>
<td>401</td>
<td>1000.00</td>
<td>0.00</td>
</tr>
<tr>
<td>402</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>403</td>
<td>6000.00</td>
<td>0.00</td>
</tr>
<tr>
<td>404</td>
<td>500.00</td>
<td>0.00</td>
</tr>
<tr>
<td>405</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

---

**DO YOU HAVE ANY ADJUSTMENTS TO MAKE? N**

**DO YOU WISH A PRINT OUT OF THE ADJUSTED TRIAL BALANCE? N**

> **PLEASE ANSWER YES OR NO: N**

**DO YOU WISH A PRINT OUT OF THE ADJUSTED TRIAL BALANCE? Y**
## INCOME STATEMENT

**REVENUES**

<table>
<thead>
<tr>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Revenue</td>
<td>11000.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>11000.00</td>
</tr>
</tbody>
</table>

**EXPENSES**

<table>
<thead>
<tr>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depreciation Expense</td>
<td>1000.00</td>
</tr>
<tr>
<td>Interest Expense</td>
<td>100.00</td>
</tr>
<tr>
<td>Supplies Expense</td>
<td>6000.00</td>
</tr>
<tr>
<td>Salaries Expense</td>
<td>500.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>7600.00</td>
</tr>
</tbody>
</table>

**NET INCOME**

<table>
<thead>
<tr>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3400.00</td>
</tr>
</tbody>
</table>

## BALANCE SHEET

**ASSETS**

<table>
<thead>
<tr>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cash</td>
<td>16900.00</td>
</tr>
<tr>
<td>Accounts Receivable</td>
<td>12000.00</td>
</tr>
<tr>
<td>Supplies</td>
<td>5000.00</td>
</tr>
<tr>
<td>Office and Equipment</td>
<td>10000.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>43900.00</td>
</tr>
</tbody>
</table>

**LIABILITIES AND OWNERS EQUITY**

<table>
<thead>
<tr>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts Payable</td>
<td>11000.00</td>
</tr>
<tr>
<td>Salaries Payable</td>
<td>400.00</td>
</tr>
<tr>
<td>Interest Payable</td>
<td>100.00</td>
</tr>
<tr>
<td>Long Term Liabilities</td>
<td>8000.00</td>
</tr>
<tr>
<td>Contributed Capital</td>
<td>20000.00</td>
</tr>
<tr>
<td>Retained Earnings</td>
<td>3400.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>42900.00</td>
</tr>
</tbody>
</table>
PROGRAM DESCRIPTION

\[ \text{ACT} \]

\[ \text{1. TO INITIALIZE VARIABLES} \]

\[ \text{2. THIS FUNCTION REQUIRE THE FOLLOWING FUNCTIONS:} \]

\[ \text{3. AYN, CENTER, COLNAMES, 6FMT, AND INC} \]

\[ \text{4. NEXT+5*7} \]

\[ \text{5. STOP+10*10} \]

\[ \text{6. OUT+7*5} \]

\[ \text{7. A01} \times \text{AYN 'DO YOU NEED INSTRUCTIONS'} \]

\[ \text{8. 'ENTER THE ACCOUNT NUMBER, THE DEBIT OR CREDIT, AND THE AMOUNT FOR EACH'} \]

\[ \text{9. 'PART OF A TRANSACTION.'} \]

\[ \text{10. 'WHEN YOU HAVE COMPLETED THE TRANSACTION ENTER } \text{NEXT'} \]

\[ \text{11. 'WHEN YOU HAVE COMPLETED ALL THE TRANSACTIONS ENTER } \text{OUT'} \]

\[ \text{12. 'THE SYSTEM WILL RESPOND BY PRINTING OUT A TRIAL BALANCE.'} \]

\[ \text{13. 'YOU WILL BE ASKED IF YOU HAVE ANY ADJUSTMENTS TO MAKE.'} \]

\[ \text{14. 'FOLLOWING THESE ADJUSTMENTS, AN ADJUSTED TRIAL BALANCE, INCOME STATEMENT, AND'} \]

\[ \text{15. 'BALANCE WILL BE PRINTED.'} \]

\[ \text{16. 'IF YOU WISH TO STOP THE PROGRAM ENTER STOP UNDER ACCOUNT NUMBER.'} \]

\[ \text{17. 'YOU WILL BE ASKED IF YOU HAVE ANY ADJUSTMENTS TO MAKE.'} \]

\[ \text{18. L01:} \]

\[ \text{19. UN='UN'} \]

\[ \text{20. START:TEMFM=TEMP+10} \]

\[ \text{21. L1: 'ACCOUNT NUMBER?'} \]

\[ \text{22. ACTN=0} \]

\[ \text{23. 'TO CHECK IF (A)THE TRANSACTION HAS BEEN COMPLETED, (B)THE CYCLE HAS BEEN COMPLETED, (C)THE EXIT SYMBOL HAS BEEN CALLED, AND (D)A VALID ACCOUNT NUMBER HAS BEEN ENTERED.'} \]

\[ \text{24. 'AGAIN:ACTN=NEXT} \]

\[ \text{25. 'FINISH:ACTN=OUT} \]

\[ \text{26. 'STOP:ACTN=STOP} \]

\[ \text{27. 'L2=((pMATRX[11])>IN+MATRX[11];ACTN} \]

\[ \text{28. 'INVALID ACCOUNT NUMBER'} \]

\[ \text{29. 'L1} \]

\[ \text{30. L2: 'DEBIT OR CREDIT'} \]

\[ \text{31. CHG=0} \]

\[ \text{32. 'TO CHECK FOR DEBIT OR CREDIT RESPONSES.'} \]

\[ \text{33. 'L3: 'AMOUNT'} \]

\[ \text{34. AMT=0} \]

\[ \text{35. 'TO ENSURE THAT THE VARIABLES NEXT, STOP, AND OUT ARE NOT EXECUTED AND THAT THE INPUT IS A SINGLE ENTRY.'} \]

\[ \text{36. 'L1;A/NEXT=AMT} \]

\[ \text{37. 'L3;A/OUT=AMT} \]

\[ \text{38. 'L3;A/STOP=AMT} \]

\[ \text{39. 'ENTER ONLY ONE NUMBER'} \]

\[ \text{40. L3} \]

\[ \text{41. L4:TEMP+TEMP,ACTN,WW×AMT} \]

\[ \text{42. 'ENTER THE DEBITS AND CREDITS AGAINST THEIR APPROPRIATE ACCOUNT NUMBERS.'} \]

\[ \text{43. MATRX[IN; 2 3]+MATRX[IN; 2 3]+WW×AMT} \]

\[ \text{44. L1} \]
A CHECKS FOR EQUALITY OF DEBITS AND CREDITS ON COMPLETION OF THE TRANSACTION.

AGAIN: TEMPM - (((pTEMP) + 3) pTEMP

START: O = -/1 + #TEMPP

'DEBITS DO NOT EQUAL CREDITS'

'ACCOUNT DB CR'

TEMPM

AGAIN

PRINTS OUT UNADJUSTED AND ADJUSTED TRIAL BALANCES.

COMPLETE: UN. 'ADJUSTED TRIAL BALANCE'

UN = 'N.

FS = 'I6, X3, 2P10.2'

FS CENTER 'MATRIX'

FS COLNAMES '/ACCTS/DEBITS/CREDITS'

FS ADMT(MATRIX)

FS ADMT(1 3 0, 0/MATRIX[; 2 3])

'N.'

'A

'DO YOU HAVE ANY ADJUSTMENTS TO MAKE?'

'DO YOU WISH A PRINT OUT OF THE ADJUSTED TRIAL BALANCE?'

'ADJUSTED TRIAL BALANCE'

FS = 'I6, X3, 2P10.2'

FS CENTER 'MATRIX'

FS COLNAMES '/ACCTS/DEBITS/CREDITS'

FS ADMT(MATRIX)

FS ADMT(1 3 0, 0/MATRIX[; 2 3])

'N.

Goes on to INCOME STATEMENT FUNCTION.

V

VINCC[]] V INC; MA; NM; FS; TR; TE; NI

ISOLATING THE REVENUE ACCOUNTS, NETTING THEM OUT, AND PRINTING THEM.

THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:

ABSHEET, CENTER, AND ADMT

MA = +(ACCTIN < 300) AND ACCTIN < 400) / MATRIX

MA[; 3] = -/MA[; 3 2]

NM = (MA[; 3]) 0 / MA

FS = '30A1, X10, P12.2'

'N.'

FS CENTER 'INCOME STATEMENT'

'REVENUES'

FS ADMT(0, 0, NM[; 1]; NM[; 3])

(40p', '), 12p' -'}
ISOLATING THE EXPENSE ACCOUNTS, NETTING THEM OUT, AND PRINTING THEM.

MA++;(ACCTN>400)\&ACCTN<500)\#MATRIX
MA[;2]--;/MA[; 2 3]
NM+((MA[;2])\#0)/MA
FS A\$FMT(NAMES[\((\;ACCTN\;\&\;NM[;1];\);\;NM[;2])]\]
(40p''),12p''-
'X40,F12.2' A\$FMT(TE++;/NM[;2])
(40p''),12p''

ARRIVING AT THE NET INCOME FIGURE AND PRINTING IT.

'X10,THE INCOME',X30,F12.2' A\$FMT(NI+TR-TE)
(52p''),10p'

RETURNING THE NET INCOME FIGURE TO RETAINED EARNINGS.

MATRIX[\((ACCTN)\;206)];3\;+\;NI

GOES ON TO THE BALANCE SHEET FUNCTION.

DEL

BSHEET

VBSHEET[]V
V BSHEET;MA;NM;FS;TA;TLE

ISOLATING THE ASSET ACCOUNTS, NETTING THEM OUT, AND PRINTING THEM.

THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:

CENTER AND A\$FMT

MA++;(ACCTN>100)\&ACCTN<200)\#MATRIX
MA[;2]--;/MA[; 2 3]
NM+((MA[;2])\#0)/MA
FS+"30A1,X10,F12.2''

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'

\'
\textbf{CNames}

1. THIS ALLOWS THE USER TO ADD, CHANGE, OR DELETE ACCOUNTS.
2. THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:
3. \texttt{AKI} AND \texttt{IPI}
4. \texttt{N1:=(1+1+\texttt{CNames})>\texttt{L}+(\texttt{ACCTN})\texttt{I}=1+\texttt{IPI 'ENTER ACCOUNT NUMBER'})/GO}
5. 'NOT A VALID ACCTI, TRY AGAIN'
6. \texttt{GO:\texttt{CNames}[L]+30+\texttt{AKI 'ENTER TITLE:'}}
7. \texttt{\texttt{\to N1}}

\textbf{Zero}

1. THIS ALLOWS THE USER TO SET ALL HIS ACCOUNTS TO ZERO
2. \texttt{\texttt{MATRIX}[;2]+\texttt{MATRIX}[;3]=0}

\texttt{\texttt{\to}}
3.2

SIMULATION OF THE
BIRTH AND DEATH OF FIRMS
IN A COMPETITIVE INDUSTRY

Dennis M. Baker
Associate Professor of Accounting and Information Systems
Faculty of Business Administration and Commerce
University of Alberta
Edmonton, Alberta
Canada

DESCRIPTION

Introduction

The program described in this paper is an APL simulation of the birth and death of firms in a competitive industry. In summary, the model operates as follows:

1. Three firms begin production in period one.
2. Production is calculated for each firm.
3. An industry price is calculated.
4. Costs and profits are calculated for each firm.
5. Any firm with a net loss disappears from the industry.
6. If the total industry profit for the current period is equal to or greater than some minimum value, one new firm enters the industry in the next period; otherwise, a new firm does not enter.
7. When these determinations have been completed, time, the number of producing firms in the industry, and the total number that have entered the industry are updated.
8. The program then returns to 2 above until the maximum number of time periods has elapsed.

The program facilitates study of the effects of two variables, FACTR and AMULT, separately and in combination, on (1) the average industry production per time period, (2) the life of the firms that are in the industry, and (3) the industry price. These variables are described in the subsequent pages.

In an attempt to conform to the economic theory of competition, the model is constructed under the following assumptions:

1. Competition is imperfect; that is, buyers and sellers do not have complete knowledge (i.e., if one firm obtains a technological breakthrough, others in the industry do not necessarily know of it immediately).

2. Firms are in business to maximize profits.

3. Each firm cannot individually influence input prices, but all working together can cause the prices of inputs to change.

4. No one firm can affect the industry price.

5. In the long run, the increase in the number of firms causes the industry supply curve to shift; and, with a given demand curve, the market price of the product will fall.

6. As new firms enter the industry, there is an effect on the costs of inputs.

7. The assumption of a "spaceless economy" is relaxed and factors such as location are allowed to affect a firm's cost function and possibly cause the death of a firm.
This simulation is an extension of Problem 6-5 described in Naylor, et al. Thus, the output of each firm is determined by the following decision rule: each firm assumes that its competitors will not change their output in the following period. Each firm then produces at the optimum level for each period, based on its cost function and the industry demand function (in which output of competitors is held constant). Consequently, assumption 2 above is met. To meet assumption 1, parameters A and C (in Problem 6-5) are changed to exogenous variables (i.e., stochastic variates with given expected values, variances, and probability distributions).

Assumptions 3 and 6 are met by adding to B the product of B and BB, an exogenous variable. If the number of firms in the industry in period t is greater than the number in period t-1, then a "percentage-of-increase-in-input-costs" (BB) is calculated. Assumption 7 enters the model in the following manner: the environment (in which the industry operates) generates an "incident" at some point in time. The time at which the incident occurs (e.g., four time periods since the last incident) is a stochastic variate with given expected value, variance, and probability distribution. A test is made comparing the life of every firm in the industry at a given time period with a variable (VA in the program) generated for "number of years between the current incident and the immediately preceding one." The firm whose life at the point of the test exceeds "between-incident-years" is subjected to the consequences of the environment-generated event.

One way to effect a net loss for the selected firm would be to alter the revenue of that firm; and, if the net profit of the firm were negative, a "death" would result. However, this approach

---

would violate assumption 4 in that if the price of one firm is
affected, the price of all firms would be affected also because
the industry price is the firm price.

Thus, because no one firm should be able to affect price,
the shock is directed at the cost function of the firm selected.
The program variable FACTR increments the costs of this firm;
depending upon the "severity" of the incident (i.e., the magnitude
of FACTR), the firm may or may not experience a net loss for the
period and thus may or may not exit the industry. One note of
cautions: a new firm is allowed to enter the industry (in the
model) if the total industry profit for the previous period does
not fall below some arbitrarily selected amount (VALUE in the
program). Therefore, it is important to note that the magnitude
assigned to FACTR not only affects the costs of one firm, but
also the profit of the total industry, with possible consequent
effects on the entrance of a new firm. Thus, if FACTR is very
high, the large loss for the firm subjected to environmental shock
may be large enough to produce a total industry loss less than
the amount of VALUE, in which case another firm would not enter the
industry in the subsequent time period.

A second way in which one firm could influence the industry
price (and therefore violate assumption 4) is created automatically
with the birth of a new firm. This possibility occurs because a
higher current period total production would be subtracted from
the constant, D, in the industry price equation. To counter this
possibility, the variable TERM is incorporated into the model.
The basic model price equation for any period subtracts from D the
quantity, "E x Current total production." To restore most of the
original industry price (as determined in period one), TERM adds
back to D (and the industry price) the quantity, "E x (total pro-
duction in period t - total production in period 1)." Although
two additional terms are also subtracted from D in the price
calculation, the weight of these terms, .20 and .03, does not affect price as much as E which has a value of .50.

Another variable, AMULT, was created to facilitate control of the impact of TERM. In the modified price equation, AMULT is multiplied times TERM. When AMULT = 1, the full impact of TERM flows through to the industry price. As AMULT decreases below 1, the impact of TERM diminishes; that is, PRICE in a period will be less than what it would have been because less is "added back" in the price equation. Thus, by decreasing the value of AMULT, one can move into the long run situation described in assumption 5, where the market price of the product will fall because of a supply curve shift resulting from an increase in the number of firms in the industry.

Variables, parameters, and operating characteristics for the model are as follows:

<table>
<thead>
<tr>
<th>Exogenous Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>a stochastic variate with a given expected value, variance, and uniform distribution.</td>
</tr>
<tr>
<td>V</td>
<td>a stochastic variate with a given expected value, variance, and uniform distribution.</td>
</tr>
<tr>
<td>A</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
<tr>
<td>BB</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
<tr>
<td>C</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
<tr>
<td>VA</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
</tbody>
</table>
FACTR = multiple by which a selected firm's production cost is enlarged.

AMULT = amount of impact on PRICE from the variable TERM.

Parameters

B = a technological parameter
D,E,F,G = constants
QLES2 = amount of total industry production two periods prior to period one of the simulation.
VALUE = lowest total industry loss level at which a new firm can enter the industry.
NBEGF = number of firms in the industry in period one.
PLES1 = production for first firm one period prior to period one of the simulation.
PLES2 = production for second firm one period prior to period one.
PLES3 = production for third firm one period prior to period one.

Endogenous Variables

PROD = matrix of quantities of units produced by firms in the industry (rows) in each time period (columns).
COST = matrix of the total cost of production for firms in the industry for each time period.
PRICE = vector of industry price in each time period.
PRFT = matrix of the net profits for firms in the industry for each time period.
Operating Characteristics

\[ \text{COST}_{it} = (\text{PROD}_{it} - A_{i})^2 + B_{i}^2 + C + U_{it} \]

\[ \text{PRICE}_{t} = D + (\text{TERM x AMULT}) - E_{t} - F_{t-1} - G_{t-2} + V_{t} \]

where

- \( i \) = \( i^{th} \) firm
- \( t \) = period \( t \)
- \( S_{t} \) = quantity of industry output supplied and demanded in period \( t \)

Potential users of this simulation include undergraduate and graduate students in economics, management science, and information systems courses. For example, application of the program may include instruction in sensitivity analysis, detailed analysis of production and financial statistics for each firm, and modification and further enhancement of the simulation itself.

**Instructions for Use**

Entering the word `DATAENTRY` at the terminal activates the function by that name. This function requests the user to enter fourteen (14) values for the exogenous variables and eleven (11) values for the parameters listed above. The final entry is the number of periods to be simulated. In the present version of the simulation, the number of firms in the industry in period one (NBEGF) must equal three and the number of time periods (NN) must satisfy the relationship, \( 1 \leq NN \leq 10 \). The twenty-six variables and their definitions for the function `DATAENTRY` are listed in the Explanation section. Note that values for QLES2, PLES1, PLES2, and PLES3 may be either computer or non-computer simulated or may
be assigned by the user, subject to certain initial conditions.

Setting the number of firms in the industry in the initial period equal to three introduces bias in the following manner. According to the cost equation, any firm's cost is roughly equal to \( \text{PROD}^2 \) (i.e., the values of A, B, C and U are small relative to the value of PROD). Therefore, assuming that each firm produces 200 units in period one, the cost of production for each firm is about \( 200^2 \), or 40,000. Price is roughly equal to \( D - E_{St} - FS_{t-1} \) or 1000 (value of the constant D) - .5(600) - .2(580) where 600 = 3 x production of 200 for each firm and 580 = the input value for the first period preceding period one of the simulation. Thus price would be about 1000 - 300 - 116, or 584.

One consequence of these initial conditions is a net profit of approximately 76,000 \((200 \text{ units} \times 584) - 200^2\). This profit margin relates directly to the minimum level at which FACTR can be set to remove a firm from the industry and yet not produce a total industry loss below VALUE which would then block the entrance of a new firm into the industry. A net profit of 76,000 provides a buffer against "death" and indicates that FACTR would have to be about 2.9 to generate a net loss and the death of a firm (i.e., \((200 \text{ units} \times 584) - (2.9 \times 200^2) = 0\) and also not effect a reduction in industry profit from production by other firms. Therefore, when the effects of changes in FACTR on various simulation outputs are being investigated, one should recognize that the initial conditions of three firms produce a measurable "margin against death." Thus, because each simulation run is biased by its initial conditions, the experimenter should make use of his \textit{a priori} information about the system under study and the purposes of the simulation to select initial conditions with a bias which will support the purposes of this study.

A second consequence of initial conditions is that \textit{PRICE} is determined by the actions of the three firms, and it is \textit{this} price
generated in period one that is maintained or approached in the
computation of TERM. Finally, besides the number of firms in
period one, another condition is the setting of \( BB = 0 \) (i.e., the
percentage of increase in input costs in period one is zero).

The function \( \text{REPORT} \) provides two outputs: the production
matrix \( \text{PROD} \) and the average industry production per time period.
The \( \text{PROD} \) output is the set of time paths of unit quantities pro­
duced by all firms. Time of entry into, and exit from, the industry
are displayed. For example, if firm \( m \) (row \( m \) of the matrix \( \text{PROD} \))
enters the industry in period 5 and exits in period 8, row \( m \) will
contain values only in columns five through eight.

In addition to the two outputs described above, modification
of the function \( \text{REPORT} \) would facilitate the outputing of other
financial and statistical information. Other financial results
available include the industry price for each period (\( \text{PRICE} \)) and
the revenue, cost, and profit for each firm in all time periods
(i.e., the variables \( \text{REV}, \text{COST}, \text{and PRFT} \)). Firm statistics avail­
able include the number of producing firms in the industry for
each period (\( \text{NFRMS} \)) and the life of each firm (\( \text{ILIFE} \)).

Experiments designed to test the effects of changes in
values of \( \text{FACTR} \) and \( \text{AMULT} \) on different variables (e.g., \( \text{COST}, \text{ILIFE} \))
include the following:

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>FACTR</th>
<th>AMULT</th>
<th>Purposes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.0</td>
<td>1</td>
<td>To test effects of decreases in ( \text{FACTR} )</td>
</tr>
<tr>
<td>2</td>
<td>5.0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9.0</td>
<td>.7</td>
<td>To test effects of \text{decreases in AMULT}</td>
</tr>
<tr>
<td>5</td>
<td>9.0</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9.0</td>
<td>.3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5.0</td>
<td>.7</td>
<td>To test joint effect of \text{decreases in FACTR and AMULT}</td>
</tr>
</tbody>
</table>
The input of typical data requested by the function DATAENTRY and typical outputs of the function REPORT are shown in the Sample Use section. For purposes of illustration, experiment No. 1 (above) was used. Data values were entered to test this experiment. (See the symbol table in the following section for the definitions of all of the variables.)

**Program Explanation**

Public functions necessary for the successful execution of REPORT include CENTER, RWTD, and AFMT. Logical segments of the simulation appear as separate functions. Consequently, execution proceeds through the functions in the following order: DATAENTRY, INITIALIZE, PRODUCTION, COSTREV, UPDATE, and REPORT. Because two or more functions must modify the value of a given variable, all variables are global. APL code for the random number generators, RANDU and RANDN, was prepared by Dr. J. L. Politzer of the Pennsylvania State University Computation Center Staff and made available to Computing Services, University of Alberta, Edmonton.

Special features of the code are the creation of the variables P and RF. When first created in the function, INITIALIZE, P is a production indicator matrix where the $(i,j)^{th}$ cell value = 1 signifies that the $i^{th}$ firm, unless removed from the industry, will produce in the $j^{th}$ time period. To record the exit of a firm from the industry, P is multiplied in the function UPDATE by RF, a matrix of zeros and ones where a zero indicates removal of a firm and a one permits continued existence of a firm. Therefore, in the subsequent time period, PROD (in the function PRODUCTION) is multiplied by the modified P.

Variables first assigned in the functions DATAENTRY, INITIALIZE, PRODUCTION, COSTREV, and UPDATE and their definitions, in order of appearance, are as follows:
<table>
<thead>
<tr>
<th>Function</th>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAENTRY</td>
<td>B</td>
<td>a technological parameter.</td>
</tr>
<tr>
<td>&quot;</td>
<td>D,E,F,G</td>
<td>constants</td>
</tr>
<tr>
<td>&quot;</td>
<td>EXVA1</td>
<td>expected value of V.</td>
</tr>
<tr>
<td>&quot;</td>
<td>VAR1</td>
<td>variance of V.</td>
</tr>
<tr>
<td>&quot;</td>
<td>EXVA2</td>
<td>expected value of U.</td>
</tr>
<tr>
<td>&quot;</td>
<td>VAR2</td>
<td>variance of U.</td>
</tr>
<tr>
<td>&quot;</td>
<td>QLES2</td>
<td>amount of total industry production two periods prior to period one of the simulation.</td>
</tr>
<tr>
<td>&quot;</td>
<td>VALUE</td>
<td>lowest total industry loss level at which a new firm can enter the industry.</td>
</tr>
<tr>
<td>&quot;</td>
<td>NBEGF</td>
<td>number of firms in the industry in period one.</td>
</tr>
<tr>
<td>&quot;</td>
<td>PLES1</td>
<td>production for first firm one period prior to period one of the simulation.</td>
</tr>
<tr>
<td>&quot;</td>
<td>PLES2</td>
<td>production for second firm one period prior to period one of the simulation.</td>
</tr>
<tr>
<td>&quot;</td>
<td>PLES3</td>
<td>production for third firm one period prior to period one of the simulation.</td>
</tr>
<tr>
<td>&quot;</td>
<td>STDA</td>
<td>standard deviation of A.</td>
</tr>
<tr>
<td>&quot;</td>
<td>EVA</td>
<td>expected value of A.</td>
</tr>
<tr>
<td>&quot;</td>
<td>STDB</td>
<td>standard deviation of BB.</td>
</tr>
<tr>
<td>&quot;</td>
<td>EVB</td>
<td>expected value of BB.</td>
</tr>
<tr>
<td>&quot;</td>
<td>STDC</td>
<td>standard deviation of C</td>
</tr>
<tr>
<td>&quot;</td>
<td>EVC</td>
<td>expected value of C.</td>
</tr>
<tr>
<td>&quot;</td>
<td>FACTR</td>
<td>multiple by which a selected firm's production cost is enlarged.</td>
</tr>
<tr>
<td>&quot;</td>
<td>AMULT</td>
<td>amount of impact on PRICE from the variable TERM.</td>
</tr>
<tr>
<td>&quot;</td>
<td>STDV</td>
<td>standard deviation of VA.</td>
</tr>
<tr>
<td>&quot;</td>
<td>EVV</td>
<td>expected value of VA.</td>
</tr>
<tr>
<td>&quot;</td>
<td>NN</td>
<td>number of time periods to be simulated.</td>
</tr>
<tr>
<td>Function</td>
<td>Variable</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>T</td>
<td>current time period.</td>
</tr>
<tr>
<td>&quot;</td>
<td>TT</td>
<td>index for columns of PROD in which production in period T will be recorded.</td>
</tr>
<tr>
<td>&quot;</td>
<td>MM</td>
<td>maximum number of producing firms (also the number of rows the columns in PROD).</td>
</tr>
<tr>
<td>&quot;</td>
<td>IN</td>
<td>column numbers of PROD to be assigned to P.</td>
</tr>
<tr>
<td>&quot;</td>
<td>P</td>
<td>production indicator matrix.</td>
</tr>
<tr>
<td>&quot;</td>
<td>ILIFE</td>
<td>vector of lives of firms (value of element i = life of firm i).</td>
</tr>
<tr>
<td>&quot;</td>
<td>BB</td>
<td>percentage of increase in input costs.</td>
</tr>
<tr>
<td>&quot;</td>
<td>NADD</td>
<td>vector of number of firms added to the industry in period T (value will equal zero or one).</td>
</tr>
<tr>
<td>&quot;</td>
<td>NDROP</td>
<td>vector of number of firms dropped in period T.</td>
</tr>
<tr>
<td>&quot;</td>
<td>PRICE</td>
<td>vector of the industry price for each time period.</td>
</tr>
<tr>
<td>&quot;</td>
<td>EXP</td>
<td>sum of production in period T-1 weighted by F plus sum of production in period T-2 weighted by G, at any time period T.</td>
</tr>
<tr>
<td>&quot;</td>
<td>V</td>
<td>a stochastic variate with a given expected value, variance, and uniform distribution.</td>
</tr>
<tr>
<td>&quot;</td>
<td>TNFRM</td>
<td>total number of firms that have entered the industry up to and including a given time period.</td>
</tr>
<tr>
<td>&quot;</td>
<td>NFRMS</td>
<td>the number of firms producing in the industry in a given time period.</td>
</tr>
<tr>
<td>&quot;</td>
<td>REV</td>
<td>matrix of revenues for firms in the industry for each time period.</td>
</tr>
<tr>
<td>&quot;</td>
<td>PRFT</td>
<td>matrix of the net profits for the firms in the industry for each time period.</td>
</tr>
<tr>
<td>Function</td>
<td>Variable</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>COST</td>
<td>matrix of the total cost of production for the firms in the industry for each time period.</td>
</tr>
<tr>
<td>&quot;</td>
<td>U</td>
<td>a stochastic variate with a given expected value, variance, and uniform distribution.</td>
</tr>
<tr>
<td>&quot;</td>
<td>VA</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
<tr>
<td>&quot;</td>
<td>C</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
<tr>
<td>&quot;</td>
<td>A</td>
<td>a stochastic variate with a given expected value, standard deviation, and normal distribution.</td>
</tr>
<tr>
<td>&quot;</td>
<td>DIV</td>
<td>the divisor in the production equation.</td>
</tr>
<tr>
<td>&quot;</td>
<td>ALPH1</td>
<td>lower bound of the range over which ( V ) will be distributed.</td>
</tr>
<tr>
<td>&quot;</td>
<td>BETA1</td>
<td>upper bound of the range over which ( V ) will be distributed.</td>
</tr>
<tr>
<td>&quot;</td>
<td>ALPH2</td>
<td>lower bound of the range over which ( U ) will be distributed.</td>
</tr>
<tr>
<td>&quot;</td>
<td>BETA2</td>
<td>upper bound of the range over which ( U ) will be distributed.</td>
</tr>
<tr>
<td>PRODUCTION</td>
<td>VV</td>
<td>reshaping of ( V ) into a vector for use in calculating ( \text{PROD} ).</td>
</tr>
<tr>
<td>&quot;</td>
<td>EEXP</td>
<td>reshaping of ( \text{EXP} ) into a vector for use in calculating ( \text{PROD} ).</td>
</tr>
<tr>
<td>&quot;</td>
<td>PP</td>
<td>vector whose ( i )th element is total production in period T-1 minus production of firm ( i ) in T-1.</td>
</tr>
<tr>
<td>&quot;</td>
<td>PROD</td>
<td>quantities produced by each firm in the industry in period T.</td>
</tr>
<tr>
<td>Function</td>
<td>Variable</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>COSTREV</td>
<td>LIFE1</td>
<td>firms at time T with age ≤ VA.</td>
</tr>
<tr>
<td></td>
<td>LIFE2</td>
<td>firms at time T with age &gt; VA.</td>
</tr>
<tr>
<td></td>
<td>CST1</td>
<td>costs for firms at time T with age ≤ VA.</td>
</tr>
<tr>
<td></td>
<td>CST2</td>
<td>costs for firms at time T with age &gt; VA.</td>
</tr>
<tr>
<td></td>
<td>COST</td>
<td>matrix of the total cost of production for firms in the industry for each time period.</td>
</tr>
<tr>
<td></td>
<td>TERM</td>
<td>E x (production in period T - production in period 1).</td>
</tr>
<tr>
<td>UPDATE</td>
<td>FIRMDROP</td>
<td>vector indexing firms to be dropped from the industry.</td>
</tr>
<tr>
<td></td>
<td>XX</td>
<td>vector of column numbers of P to be adjusted for firms exiting the industry.</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>matrix of zeros and ones where a zero indicates removal of a firm and a one permits continued existence of a firm.</td>
</tr>
<tr>
<td></td>
<td>IZ</td>
<td>period T-1.</td>
</tr>
</tbody>
</table>
PROGRAM USAGE

DATAENTRY
ENTER THE VALUES FOR B, D, E, F AND G;
INSERT ONE SPACE BETWEEN EACH NUMBER.
U:
2 1000 0.5 0.2 0.03
ENTER THE VALUES FOR EXVA1, VARK, EXVA2, VAR2,
QLES2 AND VALUE; INSERT ONE SPACE BETWEEN EACH NUMBER.
U:
0 0.5 0 1 500 3000
ENTER THE VALUES FOR NEEGF, PLES1, PLES2 AND PLES3
(LEAVE ONE SPACE BETWEEN EACH NUMBER)
U:
3 200 300 80
ENTER THE VALUES FOR STDAB, EVA, STDDE, EVB, STDC, EVC
(LEAVE ONE SPACE BETWEEN EACH NUMBER)
U:
0.1 2 0.005 0.04 1.333 24
ENTER THE VALUES FOR FACTH, AMULT, STDV AND EVV
(LEAVE ONE SPACE BETWEEN EACH NUMBER)
U:
9 1 0.6667 5
ENTER NN; NN MUST BE AN INTEGER WITH VALUE 1≤NN≤10
U:
10
CALL THE SIMULATOR BY TYPING THE WORD INITIALIZE
INITIALIZE
THE PRODUCTION MATRIX FOR THE PREVIOUS 2 PERIODS AND THE SIMULATED 10 TIME PERIODS IS:

**TIME PERIODS**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500.0</td>
<td>200.0</td>
<td>227.7</td>
<td>208.2</td>
<td>177.3</td>
<td>159.3</td>
<td>175.4</td>
<td>175.4</td>
<td>118.2</td>
<td>262.6</td>
<td>142.6</td>
<td>153.4</td>
</tr>
<tr>
<td>2</td>
<td>300.0</td>
<td>244.4</td>
<td>211.0</td>
<td>177.8</td>
<td>176.8</td>
<td>159.1</td>
<td>176.2</td>
<td>175.4</td>
<td>175.2</td>
<td>174.4</td>
<td>142.6</td>
<td>129.6</td>
</tr>
<tr>
<td>3</td>
<td>80.0</td>
<td>207.7</td>
<td>204.9</td>
<td>171.0</td>
<td>158.1</td>
<td>175.4</td>
<td>175.4</td>
<td>175.2</td>
<td>175.2</td>
<td>175.2</td>
<td>175.2</td>
<td>175.2</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
<td>170.3</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
<td>142.6</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
<td>129.6</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
<td>170.5</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AVERAGE PRODUCTION FOR EACH OF THE 10 PERIODS IS:**

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>679.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>794.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>845.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>759.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1019.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>543.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1029.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>400.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>990.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>279.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PROGRAM DESCRIPTION

V INITIALIZE

[1] A THIS FUNCTION INITIALIZES VARIOUS ARRAYS AND PARAMETERS
[2] AND SETS VALUES FOR DIV, ALPH1, BETA1, ALPH2 AND BETA2
[3] T+SETLINK120
[4] T+1
[5] TT+T+2
[6] MM+D6+2
[7] NN+D6
[8] PROD+(1MM)*.a.sMM
[9] IN+2+NN
[10] P+PROD[;IN]
[17] ILIFE+(3Pl),(MM-3);D0
[18] V+EXP+PRICK+ADDUP+XADD+BB+D6;D0
[19] NeAMB+TPFM+D3[1]((NN-1);D0)
[20] A+C+VA+U+COST+PRFT+REV+(MM,NN);D0
[23] ALPH1+D2[1]-0.5*3xG2[2]
[24] BETA1=(2xD2[1])-ALPH1
[25] ALPH2=(2xD2[3])-ALPH2
[26] PRODUC(V)

V PRODUCTION

[1] A THIS FUNCTION CALCULATES THE UNIT PRODUCTION FOR
[2] EACH FIRM IN EXISTENCE IN THE CURRENT TIME PERIOD
[3] W+MMpV[T]-ALPH1+(BETA1-ALPH1)*Z+RANDU 1
[6] EACH FIRM ASSUMES THAT THE ONLY CHANGES IN OUTPUT FOR THE
[8] WILL BE ATTRIBUTED ONLY TO ITS CHANGES IN OUTPUT AND
[9] THAT OF ITS COMPETITORS. NOW, FOR EACH FIRM, CALCULATE
[10] THE TOTAL PRODUCTION OF ALL OTHER FIRMS IN THE PREVIOUS
[12] INDUSTRY PRODUCTION.
[14] DETERMINE PRODUCTION FOR THE CURRENT PERIOD; IF THE
[15] PRODUCTION INDICATOR MATRIX & CONTAINS A CELL VALUE
[16] OF ZERO, THIS INDICATES THE DEATH OF A FIRM IN THE
[17] PREVIOUS TIME PERIOD AND THEREFORE NO PRODUCTION
[18] FOR THIS FIRM IN THE CURRENT PERIOD.
[20] UPDATE THE LIFE OF THE FIRMS IN EXISTENCE BY ONE TIME PERIOD
[21] ILIFE+ILIFE+(PROD[T];T)>0
[22] COSTREV

V
\[ \text{\# R+A CENTER B} \]
\[ \text{\# A++/x/ 0 2 +HWTD A} \]
\[ \text{\# R+(1.A) A+((0.5+x(A-oB)+')} , \text{B-} , \text{B} \]

\[ \text{\# COSTREV} \]
\[ \text{\# THIS FUNCTION DETERMINES THE COST OF PRODUCTION FOR} \]
\[ \text{\# FIRMS WHOSE PRESENT AGE IS \leq \ OR \ > \ THE RANDOM VARIABLE} \]
\[ \text{\# AND REVENUE AND PROFIT FOR FIRMS IN THE CURRENT PERIOD} \]
\[ \text{\# ARE THEN DETERMINED.} \]
\[ \text{\# U[;T]+ALPH2+ (BETA2-ALPH2)*XU[;T]+RUNIT 1} \]
\[ \text{\# UL[;T]-UE[;T]-US[;T]} \]
\[ \text{\# C[;T]+D4[;T] RANDOM D4[;T]} \]
\[ \text{\# IDENTIFY FIRMS WITH PRESENT AGE\leq VA} \]
\[ \text{\# LIFE1+LIFE\leq VA[;T]} \]
\[ \text{\# IDENTIFY FIRMS WITH PRESENT AGE\geq VA} \]
\[ \text{\# LIFE2+LIFE>VA[;T]} \]
\[ \text{\# COST1+R1+ALPH2+ (BETA2-ALPH2)*XU[;T]+RUNIT 1} \]
\[ \text{\# U[;T]+ALPH2+ (BETA2-ALPH2)*XU[;T]+RUNIT 1} \]
\[ \text{\# C[;T]+D4[;T] RANDOM D4[;T]} \]
\[ \text{\# COST1[;T]+P[;T]xLIFE1 xC[;T]+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# LARGE C[;T]+LIFE2 xD5[;T]+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# COST1[;T]+P[;T]xLIFE1 xC[;T]+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# PRICE[;T]+D1[2]*D5[;T]+D5[;T]+LIFE1+LIFE2+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# PRICE[;T]+D1[2]*D5[;T]+D5[;T]+LIFE1+LIFE2+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# PRICE[;T]+D1[2]*D5[;T]+D5[;T]+LIFE1+LIFE2+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# PRICE[;T]+D1[2]*D5[;T]+D5[;T]+LIFE1+LIFE2+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# PRICE[;T]+D1[2]*D5[;T]+D5[;T]+LIFE1+LIFE2+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]
\[ \text{\# PRICE[;T]+D1[2]*D5[;T]+D5[;T]+LIFE1+LIFE2+U[;T]+(Bx(A[;T]*2))+( (PROD[;TT]-A[;T]))*2} \]

\[ \text{\# DATAENTRY} \]
\[ \text{\# 'ENTER THE VALUES FOR B, D, E, F AND G;'} \]
\[ \text{\# 'INSERT ONE SPACE BETWEEN EACH NUMBER.'} \]
\[ \text{\# D1+} \]
\[ \text{\# 'ENTER THE VALUES FOR EXVA1, VAR1, EXVA2, VAR2,'} \]
\[ \text{\# 'QLES2 AND VALUE; INSERT ONE SPACE BETWEEN EACH NUMBER.'} \]
\[ \text{\# D2+} \]
\[ \text{\# 'ENTER THE VALUES FOR NBEGF, PLES1, PLES2 AND PLES3'} \]
\[ \text{\# 'LEAVE ONE SPACE BETWEEN EACH NUMBER')'} \]
\[ \text{\# D3+} \]
\[ \text{\# 'ENTER THE VALUES FOR STD, EVA, STD, EVB, STDC, EVC'} \]
\[ \text{\# 'LEAVE ONE SPACE BETWEEN EACH NUMBER')'} \]
\[ \text{\# D4+} \]
\[ \text{\# 'ENTER THE VALUES FOR FACTR, AMULT, STDV AND EVV'} \]
\[ \text{\# 'LEAVE ONE SPACE BETWEEN EACH NUMBER')'} \]
\[ \text{\# D5+} \]
\[ \text{\# 'ENTER NN; NN MUST BE AN INTEGER WITH VALUE 15NN\leq10'} \]
\[ \text{\# D6+} \]
\[ \text{\# 'CALL THE SIMULATOR BY TYPING THE WORD INITIALIZE'} \]
\[ UPDATE \]

1. This function drops and/or adds firms to the industry, updates time periods and calculates the change, if any, in the parameter $\beta$.
2. Count the number of firms to be dropped.
3. Determine those firms to be dropped from the industry.
4. Index firms which will exit the industry.
5. Count the number of firms to be dropped.
6. Determine those firms to be dropped from the industry.
7. Index firms which will exit the industry.
8. If total industry profit + value ($D_2[i]$) is ≤ 0, do not add a firm to the industry; otherwise, add one firm.
9. Update: $\rightarrow (((\#PRT[i]+D_2[i]) \leq 0) / UP1)
10. $NADD[T] + NADD[T] + 1$
11. $\rightarrow ((T-NN) \geq 0) / REP$
12. $\rightarrow NEXTPERIOD$
13. $UP1: \rightarrow ((T-NN) \geq 0) / REP$
14. $\rightarrow NEXTPERIOD: T = T + 1$
15. If number of firms in new period is > number in period $T - 1$, adjust parameter $\beta$.
16. $T + T - 1$
17. $NFRMS[T] + (NFRMS[I] + NADD[I] - NDROP[I])$
18. $TNFRM[T] + (TNFRM[I] + NADD[I])$
19. $\rightarrow ((NFRMS[T] - NFRMS[I]) \leq 0) / UP3$
21. Production
22. $UP3: B[T] = 0$
23. Production
24. $\rightarrow 0$
25. $REP: REPORT$
V ZZ+SIG RANDN MU
[1] ZZ+MU+SIGx^-6+(+/^-1+?178956970)*178956969

V ZZ+SIG RANDOR MU
[1] X+1
[2] ZZ+MMp0
[4] +(MM<X+X+1)/0
[5] +3

V ZZ+RANDU X
[1] ZZ+X(-1+?2147483647)*2147483646

V RU+RANUNI X
[1] RU+X(-1+?MMp2147483647)*2147483646

V REPORT
[1] FORMAT='I4,5X1,',(RP='0123456789'[1+ 10 10 TMM]),'BP8.1'
[3] 'SIMULATED ',NN,' TIME PERIODS IS:';3PR
[4] FORMAT CENTER 'TIME PERIODS';4PQR
[5] ('X9,RP,T9') AFMT(1,MM)p-2+1M0
[6] FORMAT AFMT(\NN;PROD);8PQR
[7] 'AVERAGE PRODUCTION FOR EACH OF THE ',NN,' PERIODS IS:';3PR
[8] 'I4,5X1,F8.1' AFMT(\NN;(+#PROD[:(2+\NN)]4NFRMS[T]))
[9] +0

V TESTDATA
[1] D1+ 2 1000 0.5 0.2 0.03
[2] D2+ 0 0.5 0 1 500 3000
[3] D3+ 3 200 300 80
[4] D4+ 0.1 2 0.005 0.04 1.333 24
[5] D5+ 9 1 0.6667 5
[6] D6+10
[7] INITIALIZE
3.3

SINGLE ASSET DEPRECIATION METHODS

Lawrence D. Brown
Assistant Professor of Accounting and Economics
School of Management
State University of New York at Binghamton
Binghamton, New York

DESCRIPTION

The primary purpose of this program, entitled DEPRECIATE, is to aid management educators in teaching single asset depreciation methods to beginning and intermediate financial accounting students. The program can be used for drill and practice, homework assignments, or in the classroom. It enables the user to choose any of the following depreciation methods:

1. Straight Line (SL)
2. Sum-of-the-years Digits (SYD)
3. Declining Balance (DB)
4. Double Declining Balance (DDB)
5. Sinking Fund (SF)
6. Annuity (AN)

Usage Instructions

The user is initially asked to respond to the following four questions:

1. Historical cost of asset?
2. Expected net salvage value of asset?
3. Expected life of asset (in years)?
4. Depreciation method desired: SL, SYD, DB, DDB, SF, or AN?
The Historical Cost (HC) can be any positive number less than one trillion. The Net Salvage Value (NSV) can be any positive number less than the historical cost of the asset. The Expected Asset Life (N) can be any integer between one and nine hundred ninety-nine. If any of these constraints are violated, asterisks will be printed out in the depreciation table in lieu of numbers.

One of the mnemonics representing the six depreciation methods identified above must be entered in response to question 4. If the user does not input one of these available depreciation methods, he will receive an error message and will be asked question 4 again.

After these four questions are answered satisfactorily, the user obtains a depreciation table containing Annual Depreciation Expense (DE) Accumulated Depreciation (ACC), and Book Value (BV). The user is then given the option to plot depreciation expense over time. He is next asked whether or not he desires to use the same data for a different depreciation method. If he responds YES, he is reasked question 4 above. If he answers NO, he is asked if he wants to enter new data. An affirmative answer results in question 1 being asked again. A negative response yields the message, GOODBYE FOR NOW, and exits the user from the program.

In addition to the function DEPRECIATE, the following public functions are also required: AKI, AYN, NIP, PLOT, and VS. In addition, Scientific Time Sharing Corporations CENTER, COLNAMES, and AFMT are also needed in order to execute the program successfully.
PROGRAM USAGE

DEPRECIATE
HISTORICAL COST OF ASSET?: 25000
EXPECTED NET SALVAGE VALUE OF ASSET?: 5000
EXPECTED LIFE OF ASSET (IN YEARS)?: 13
DEPRECIATION METHOD DESIRED: SL,SYD,DB,DDB,SF,OR ANY? DDB

TABLE OF ASSET DEPRECIATION

<table>
<thead>
<tr>
<th>YEAR</th>
<th>DEP EXP</th>
<th>ACC DEP</th>
<th>BOOK VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25000.00</td>
</tr>
<tr>
<td>1</td>
<td>3846.15</td>
<td>3846.15</td>
<td>21153.85</td>
</tr>
<tr>
<td>2</td>
<td>3254.44</td>
<td>7100.59</td>
<td>17899.41</td>
</tr>
<tr>
<td>3</td>
<td>2753.76</td>
<td>9854.35</td>
<td>15145.65</td>
</tr>
<tr>
<td>4</td>
<td>2330.10</td>
<td>12184.45</td>
<td>12815.55</td>
</tr>
<tr>
<td>5</td>
<td>1971.62</td>
<td>14156.07</td>
<td>10843.93</td>
</tr>
<tr>
<td>6</td>
<td>1668.30</td>
<td>15824.37</td>
<td>9175.63</td>
</tr>
<tr>
<td>7</td>
<td>1411.64</td>
<td>17236.00</td>
<td>7764.00</td>
</tr>
<tr>
<td>8</td>
<td>1194.46</td>
<td>18430.46</td>
<td>6569.54</td>
</tr>
<tr>
<td>9</td>
<td>1010.70</td>
<td>19441.16</td>
<td>5558.84</td>
</tr>
<tr>
<td>10</td>
<td>558.84</td>
<td>20000.00</td>
<td>5000.00</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>20000.00</td>
<td>5000.00</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>20000.00</td>
<td>5000.00</td>
</tr>
<tr>
<td>13</td>
<td>0.00</td>
<td>20000.00</td>
<td>5000.00</td>
</tr>
</tbody>
</table>

DO YOU WANT A PLOT OF DEPRECIATION EXPENSE VERSUS TIME?: Y
PLEASE ANSWER YES OR NO: YES

DO YOU WANT TO USE THE SAME DATA TO CALCULATE DEPRECIATION EXPENSE, ACCUMULATED DEPRECIATION AND BOOK VALUE USING A DIFFERENT DEPRECIATION METHOD? YES
DEPRECIATION METHOD DESIRED: SL,SYD,DB,DDB,SF,OR ANY? SF
WHAT INTEREST RATE DO YOU WISH TO APPLY TO THIS ASSET? (0 ≤ INT RATE ≤ 100): 11
### Table of Asset Depreciation

<table>
<thead>
<tr>
<th>YEAR</th>
<th>DEP EXP</th>
<th>ACC DEP</th>
<th>BOOK VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>763.02</td>
<td>763.02</td>
<td>24236.98</td>
</tr>
<tr>
<td>2</td>
<td>846.95</td>
<td>1609.97</td>
<td>23390.03</td>
</tr>
<tr>
<td>3</td>
<td>940.12</td>
<td>2550.09</td>
<td>22449.91</td>
</tr>
<tr>
<td>4</td>
<td>1043.53</td>
<td>3593.62</td>
<td>21406.38</td>
</tr>
<tr>
<td>5</td>
<td>1158.32</td>
<td>4751.94</td>
<td>20248.06</td>
</tr>
<tr>
<td>6</td>
<td>1285.73</td>
<td>6037.67</td>
<td>18962.33</td>
</tr>
<tr>
<td>7</td>
<td>1427.16</td>
<td>7464.83</td>
<td>17535.17</td>
</tr>
<tr>
<td>8</td>
<td>1584.15</td>
<td>9048.98</td>
<td>15951.02</td>
</tr>
<tr>
<td>9</td>
<td>1758.41</td>
<td>10807.39</td>
<td>14192.61</td>
</tr>
<tr>
<td>10</td>
<td>1951.83</td>
<td>12759.22</td>
<td>12240.78</td>
</tr>
<tr>
<td>11</td>
<td>2166.53</td>
<td>14925.76</td>
<td>10074.24</td>
</tr>
<tr>
<td>12</td>
<td>2404.85</td>
<td>17330.61</td>
<td>7669.39</td>
</tr>
<tr>
<td>13</td>
<td>2669.39</td>
<td>20000.00</td>
<td>5000.00</td>
</tr>
</tbody>
</table>

Do you want a plot of depreciation expense versus time?: **Yes**

---

Do you want to use the same data to calculate depreciation expense, accumulated depreciation and book value using a different depreciation method? **No**

Do you want to enter new data? **No**

Goodbye for now.
PROGRAM DESCRIPTION

\[ \text{DEPRECIATE} \]

This function requires the following other functions:
- \text{AXI}, \text{AYN}, \text{IPF}, \text{NIP}, \text{OUTPUT}, \text{PLOT}, \text{AND VS}.

\[ L2 : L3 \times 0 + HC - NIP \quad \text{'HISTORICAL COST OF ASSET?'} \]
\[ L3 : L4 \times 0 + NIP \quad \text{'EXPECTED NET SALVAGE VALUE OF ASSET?'} \]

\[ 'HISTORICAL COST MUST BE GREATER THAN ZERO: ' \quad \text{\( L2 \times 0 + L2 \)} \]
\[ 'EXPECTED NET SALVAGE VALUE OF ASSET MUST BE POSITIVE: ' \quad \text{\( L3 \times 0 + L3 \)} \]

\[ L4 : (NIP < HC) / L5 \quad \text{'HISTORICAL COST MUST EXCEED NET SALVAGE VALUE: ' \quad \text{\( L3 \times 0 + L3 \)} \]
\[ L5 : L6 \times 0 + L6 \quad \text{'EXPECTED LIFE OF ASSET (IN YEARS)?': ' \quad \text{\( L6 \times 0 + L6 \)} \]
\[ 'EXPECTED LIFE OF ASSET MUST EXCEED ONE: ' \quad \text{\( L5 \times 0 + L5 \)} \]

\[ L6 : 'DEPRECIATION METHOD DESIRED: SL, SYD, DB, DDB, SF, OR AN?' \]
\[ + (SL, SYD, DB, DDB, SF, AN, ERR) \quad \text{\( \{L6|\{L6\} \times 0 + L6 \)} \]
\[ ERR: 'YOU MADE AN ERROR. THE ONLY PERMISSIBLE DEPRECIATION METHODS ARE SL, SYD, DB, DDB, SF, AND AN:' \]
\[ + L6 \quad \text{\( \{L6|\{L6\} \times 0 + L6 \)} \]

\[ SL : L9, CV + HC - ACC + ((1 + INT) \times N) \times DE + Np(0 + HC - NSV) + N \]
\[ SYD : L9, CV + HC - ACC + ((1 - FRAC) \times ((1 + INT) \times N)) \times FRAC + ((1 + INT) \times (1 + INT) \times N) \times N \times N \]
\[ DB : CV + HC - ACC + \{DE + HC \times ((1 - FRAC) \times ((1 + INT) \times N)) \times FRAC + 2 \times N \]
\[ DDB : DE - ACC - ((1 + ACC + HC - CV - CV \times NSV) \times N \times N) \]
\[ SF : L7 \times 1 + INT \times 0.01 \times INT + NIP \quad \text{WHAT INTEREST RATE DO YOU WISH TO APPLY TO THIS ASSET? (0 \leq \text{INT RATE} \leq 100): ' \]
\[ + L7 \quad \text{\( \{L7|\{L7\} \times 0 + L7 \)} \]
\[ AN : L8 \times 1 + INT \times 0.01 \times INT + NIP \quad \text{WHAT INTEREST RATE DO YOU WISH TO APPLY TO THIS ASSET? (0 \leq \text{INT RATE} \leq 100): ' \]
\[ + L8 \quad \text{\( \{L8|\{L8\} \times 0 + L8 \)} \]

\[ 'INTEREST RATE CANNOT EXCEED ONE HUNDRED.' \quad \text{\( + SF \)} \]
\[ L9 : CV + HC + ACC + \{DE + (HC - NSV) \times ((1 + INT) \times N) \times (1 + INT) \times (1 + INT) \times (1 + INT) \times (1 + INT) \times N \times N \}
\[ 'INTEREST RATE CANNOT EXCEED ONE HUNDRED.' \quad \text{\( + AN \)} \]
\[ L10 : \text{"DO YOU WANT TO USE THE SAME DATA TO CALCULATE DEPRECIATION EXPENSE,\text{"} \]
\[ + L10 \quad \text{\( \{L10|\{L10\} \times 0 + L10 \)} \]

\[ L1 : 30 \times 0 + \text{PLT DE VS N} \quad \text{GOODBYE FOR NOW' \( + 0 \)} \]

\[ \text{OUTPUT} \]

This function requires the following other functions:
- \text{CENTER}, \text{COLNAMES}, \text{AND AFMT}

\[ \text{FS =}'I4, 3F17.2' \]
\[ ' ', \]
\[ FS CENTER 'TABLE OF ASSET DEPRECIATION' \]
\[ ' ', \]
\[ FS COLNAMES '/YEAR/DEP EXP/ACC DEP/BOOK VALUE/' \]
\[ R + N \]
\[ FS AFMT(R; DE; ACC; CV) \]
REAL ESTATE VALUATION USING THE DIRECT CAPITALIZATION METHOD

Fred E. Case
Professor of Urban Land Economics
Graduate School of Management
University of California
Los Angeles, California

DESCRIPTION

Introduction

The value of income-producing real estate may be calculated by one of two methods: the present worth of a stream of future benefits or by the direct capitalization method. The most commonly used -- and most conservative -- is the latter, and it is this method which is used in this program. The basic formula is

\[ V = NY \div CF, \]

where \( V \) is value, \( NY \) is the net income, and \( CF \) is the capitalization factor. Depending upon the data available, the building residual method, the land residual method, or the property residual method can be used. The accompanying program, DIRCAL, is designed to perform each of these calculations.

This program can be used by anyone wanting to understand the fundamentals of the direct method of capitalization and to calculate the value for any type of income-producing property for which data are available. The program is useful in both undergraduate and graduate courses in real estate economics, valuation, finance, and investments.

Use of the Program

To use the function, the user provides the information requested and is then supplied with the answers. Basically, the user must supply data on gross scheduled income; a vacancy and
collection loss percentage; operating expenses, both fixed and reserved, expressed as a percentage of gross scheduled income; property taxes as a percent of the property market value; anticipated earnings rate; and anticipated earnings period or economic life. Then, depending upon the data which are available, estimates are required on the land value; the building value; and the expected residual value of the property at the end of the investment period.

In entering data, a few words of explanation are in order. The following are some of the input prompts, accompanied by a more detailed explanation of what they mean.

**ENTER GROSS SCHEDULED INCOME (GSI)**

Gross scheduled income refers only to scheduled rents, not to other forms of income.

**ENTER OPERATING EXPENSES (OE) AS A PERCENTAGE OF GROSS SCHEDULED INCOME**

Operating expenses are not total expenses; total expenses include operating expenses (OE), reserve expenses (RE), fixed expenses (FE), and property taxes (TE).

**ENTER RESERVE EXPENSE PERCENTAGE (RE)**

Reserve expenses are cash outlays which occur in one year which relate to more than one year of operation. To obtain the figure for reserve expenses, divide each reserve expense by the number of years over which it is to be allocated. For example, roof repairs might cost $1000 in one year but be expected to last for ten years. Dividing $1000 by 10 would yield an annual cost of $100. All these annualized expenses should then be totalled and expressed as a percentage of the gross operating income. Reserve expenses are not made for the purpose of increasing the rents but merely to maintain the scheduled rents.
ENTER FIXED EXPENSE PERCENTAGE (FE)

Fixed expenses include such things as fire insurance, fees, licenses, and similar types of costs whose amounts are determined externally and are not a function of the level of operation.

ENTER MARKET VALUE OF THE PROPERTY (MV)

The market value is either the original purchase price or the current estimated market value, whichever is most appropriate. If neither are available, use the assessed value, adjusted to approximate the market value.

ENTER LAND VALUE (LV)
ENTER BUILDING VALUE (BV)

Each of the above should be entered separately. If they are not available directly, they should be derived from the assessed values.

The remaining requests for input are fairly straightforward. When all inputs have been completed, the program will print out the resultant value estimations. These are calculated according to the building residual method, the land residual method, and the property residual method. If the user wishes to run the program again, retying DIRCAL will cause it to restart.

All data must be entered each time calculations are run. However, the user can vary the data each time according to his needs. The sample session provides an illustration of the kinds of data asked for and the output produced, given the data provided.
ENTER ALL PERCENTAGE FIGURES AS DECIMAL FRACTIONS, E.G.,
8 PERCENT WOULD BE ENTERED AS .08
IF YOU MAKE A MISTAKE ENTERING DATA, START OVER.

ENTER GROSS SCHEDULED INCOME (GSY)
\( \square \): 30000

ENTER VACANCY AND COLLECTION LOSS (VCL) AS A PERCENTAGE OF
GROSS SCHEDULED INCOME.
\( \square \): .06

ENTER OPERATING EXPENSES (OE), AS A PERCENTAGE OF GROSS SCHEDULED INCOME.
\( \square \): .10

ENTER RESERVE EXPENSE PERCENTAGE (RE)
\( \square \): .04

ENTER FIXED EXPENSE PERCENTAGE (FE)
\( \square \): .06

ENTER MARKET VALUE OF THE PROPERTY (MV)
\( \square \): 190000

ENTER PROPERTY TAX EXPENSES (TE) AS PERCENTAGE OF MARKET VALUE
\( \square \): .025

ENTER LAND VALUE (LV)
\( \square \): 70000

ENTER BUILDING VALUE (BV)
\( \square \): 120000

ENTER INTEREST RATE OR EARNING RATE (EGR).
\( \square \): .08

ENTER EARNING PERIOD OR ECONOMIC LIFE, IN YEARS (ECL)
\( \square \): 40

ENTER EXPECTED RESIDUAL VALUE (ERV) AT END OF EARNING PERIOD.
IF YOU CANNOT ESTIMATE RESIDUAL VALUE, USE LAND VALUE (LV).
\( \square \): 70000
VALUE ESTIMATE CALCULATIONS USING THE BUILDING RESIDUAL METHOD

LAND INCOME (LVY) 5600
BUILDING INCOME (RBV) 12210
VALUE OF BUILDING (FBV) 116285.7143
VALUE ESTIMATE DERIVED (FBV) 186285.7143

VALUE ESTIMATE CALCULATIONS USING THE LAND RESIDUAL METHOD

BUILDING INCOME (BLV) 12600
LAND INCOME (RLY) 5210
LAND VALUE ESTIMATE (RLV) 65175
VALUE ESTIMATE (RPV) 185125

VALUE ESTIMATE CALCULATIONS, DERIVED USING THE PROPERTY RESIDUE METHOD

PRESENT VALUE OF FUTURE NET INCOME STREAM (PVY) 169619.0476
PRESENT VALUE OF ANY RESIDUAL PROPERTY AT END OF INVESTMENT (DLV) 3222.165331
VALUE ESTIMATE DERIVED (PRV) 172841.213

ANNUAL PERCENT NET INCOME DECLINE WHILE PROTECTING EQUITY EARNING RATE 0.01904761905
PROGRAM DESCRIPTION

V DIRCAL
[1]  
[2]  'ENTER ALL PERCENTAGE FIGURES AS DECIMAL FRACTIONS, E.G.,'
[3]  '0 PERCENT WOULD BE ENTERED AS .00'
[4]  'IF YOU MAKE A MISTAKE ENTERING DATA, START OVER.'
[5]  
[6]  'ENTER GROSS SCHEDULED INCOME (GSY)'
[7]  GSY+\n
[8]  'ENTER VACANCY AND COLLECTION LOSS (VCL) AS A PERCENTAGE OF
GROSS SCHEDULED INCOME.'
[VCL+\n
[9]  'ENTER OPERATING EXPENSES (OE), AS A PERCENTAGE OF GROSS SCHEDULED INCOME
.'
[10]  OE+\n
[11]  'ENTER RESERVE EXPENSE PERCENTAGE (RE)'
[12]  RE+\n
[13]  'ENTER FIXED EXPENSE PERCENTAGE (FE)'
[14]  FE+\n
[15]  'ENTER MARKET VALUE OF THE PROPERTY (MV)'
[16]  MV+\n
[17]  'ENTER PROPERTY TAX EXPENSES (TE) AS PERCENTAGE OF MARKET VALUE'
[18]  TE+\n
[19]  'ENTER LAND VALUE (LV)'
[20]  LV+\n
[21]  'ENTER BUILDING VALUE (BV)'
[22]  BV+\n
[23]  'ENTER INTEREST RATE OR EARNING RATE (EGR)'
[24]  EGR+\n
[25]  'ENTER EARNING PERIOD OR ECONOMIC LIFE, IN YEARS (ECL)'
[26]  ECL+\n
[27]  'ENTER EXPECTED RESIDUAL VALUE (ERV) AT END OF EARNING PERIOD.'
[28]  'IF YOU CANNOT ESTIMATE RESIDUAL VALUE, USE LAND VALUE (LV).'
[29]  ERV+\n
[30]  'VALUE ESTIMATE CALCULATIONS USING THE BUILDING RESIDUAL METHOD'
[31]  GOY+GSY-(GSYxVCL)
[32]  NY+GOY-(GOYxOE+RE+FE)+TExMV
[33]  LVY+LVxEGR
[34]  RBY+NY-LVY
[35]  BPY=RBY*(EGR+(1+ECL))
[36]  BPV+LVxBBV
[37]  RLY=BY*(EGR+(1+ECL))
[38]  RLY=NY-BLY
[39]  RLY+ELYxEGR
[40]  RBV+BVxRLV
[41]  PVY=NY*(EGR*(1+ECL))
[42]  DLV=PVYx(1+(1+EGR)+ECL)
[43]  PRV=PY+DLV
[44]  DLY=(EGRx(1+ECL))+(EGRx(1+ECL))
[45]  
[46]  'VALUE ESTIMATE CALCULATIONS USING THE BUILDING RESIDUAL METHOD'
VALUE ESTIMATE CALCULATIONS USING THE LAND RESIDUAL METHOD

VALUE ESTIMATE CALCULATIONS, DERIVED USING THE PROPERTY RESIDUE METHOD

PRESENT VALUE OF ANY RESIDUAL PROPERTY AT END OF INVESTMENT (DLV)
3.5

SIMULATION OF RANDOM SAMPLES
AND SAMPLING DISTRIBUTIONS OF VARIOUS STATISTICS

John J. Chai
Associate Professor of Quantitative Methods
School of Management
Syracuse University
Syracuse, New York

DESCRIPTION

Introduction

This program is made up of three functions: (1) POPDIST, (2) SIMUSAMPLE, and (3) SAMPDIST. SIMUSAMPLE is the main function and the other two are subfunctions. The first function inputs a finite population and formulates the frequency (or probability) distribution with equal class intervals; the second function generates, from the input, simple random samples with replacement of a specified size for a specified number of replications; and the third function formulates the sampling distribution of a specified statistic such as sample mean, median, range, variance, and standard deviation. The program is designed primarily to teach students in a beginning applied statistics course (undergraduate and graduate levels) the concepts of simple random sampling from a finite population, the sampling distribution of a statistic, the central limit theorem, and the relationship between parameters and statistics. This program can also be used for a general simulation problem. The program is intended to be used in a flexible way in that the instructor may use it in a classroom as a teaching device or students may use it independently outside class as an exercise in statistical problems related to sampling distribution theory.
Instructions for Use

It is assumed that the users know what a frequency (or probability) distribution is; what simple random sampling without replacement is; what a sampling distribution is; what the central limit theorem is; and what relationship exists between the statistic involved and the parameters of the population under consideration. The data for input is a set of a finite population. Either a string of individual values (ungrouped data) of a population or a frequency distribution (grouped data) of a population can be entered. If a string of individual values is to be entered as input, the program asks a user to specify the parameters necessary for formulating the ungrouped data into a frequency distribution (i.e., the lower class limit value of the first class, the class interval, and the number of classes required). The program outputs the maximum and minimum values of the input for the user's convenience. If a user wants to input a frequency distribution, the program asks for the class midpoints and corresponding frequencies as input. The program also asks for the population size. If a user inputs a probability distribution, the program asks the user to convert the probabilities to absolute frequencies based on the population size. Then the program outputs a table and a graph showing the details of the population. The table shows lower and upper class limits, class midpoints, absolute and relative frequencies (probabilities), and absolute and relative cumulative frequencies. The graph is in the form of a histogram.

The program also outputs the values of population mean, median, range, variance, and standard deviation. The program then asks the user to input the size of the sample and the number of replications. Next, the program outputs the names of statistics for which sampling distributions can be developed and asks the user to designate one...
statistic. The program outputs the maximum and minimum values of
the statistic and asks the user to input the lower class limit
value of the first class, class interval, and the number of classes
required to formulate the sampling distribution. The program then
outputs the sampling distribution in tabular and graphic forms,
which are very much like the ones given for the population data.
It also outputs the various parameters of the sampling distribution.

The program gives the user the option of respecifying the
three parameter values necessary for formulating the frequency
distribution of the population and the sampling distributions. It
also provides the user with the option of continuing the program
with a new statistic or a new population.

The constraints of the program are as follows. First, the
population size is set to at least 100. This is an arbitrary num­
ber. In fact, one should be aware that the population size must be
considerably larger than the sample size in order that the effect
of the central limit theorem can be realized. On the other hand,
one should be aware of workspace size limitations if a string of
ungrouped data of a large size population is to be entered. In
this case, one is advised to formulate a frequency distribution
first and input it instead of inputting ungrouped data. Second,
the sample size and the number of replications are also limited
to a maximum of 99 and 50, respectively, in view of the workspace
size limitation and the high cost of replications. Third, a
user must select the lower class limit value of the first class to
be slightly less than the minimum value in the data for formulating
a frequency distribution of a population and for developing a
sampling distribution, for the lower class limit values are not
counted into the classes.

To use the program, the student merely needs to type
SIMUSAMPLE. This first program then calls the others (POPDIST
and SAMPDIST). In addition to these functions, the public functions AKI, AYN, CENTER, COLNAMES, ΔFMT, PLOT, and VS are needed.
SIMSAMPLE

A POPULATION DATA TO BE ENTERED AS INPUT MAY BE IN THE FORM OF A VECTOR OF INDIVIDUAL VALUES (UNGROUPED DATA) OR IN THE FORM OF FREQUENCY (OR PROBABILITY) DISTRIBUTIONS. PLEASE FOLLOW THE INPUT INSTRUCTIONS BELOW CAREFULLY.

HAVE YOU ALREADY ENTERED THE POPULATION DATA? TYPE YES OR NO. N

ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:

90

IS YOUR DATA UNGROUPED? TYPE 'Y' OR 'N'. Y

ENTER UNGROUPED DATA. TYPE ',,' IF MORE SPACE IS NEEDED FOR ENTERING DATA.

SYNTAX ERROR

A

MAXIMUM VALUE IS: 3267
MINIMUM VALUE IS: 75

IN ORDER TO SHAPE A FREQUENCY DISTRIBUTION FOR THE UNGROUPED DATA YOU ENTERED, YOU ARE ASKED TO SPECIFY THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, WIDTH OF CLASS, AND NUMBER OF CLASSES, E.G., 5 20 10. ENTER THESE THREE VALUES:

5 600 6

***********************************************************************************************

<table>
<thead>
<tr>
<th>LOWER LIMIT</th>
<th>UPPER LIMIT</th>
<th>MID POINT</th>
<th>FREQUENCY</th>
<th>REL FREQUENCY</th>
<th>CUMULATIVE FREQUENCY</th>
<th>REL CUMULATIVE FREQUENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.00</td>
<td>606.00</td>
<td>306.00</td>
<td>299</td>
<td>59.80</td>
<td>299</td>
<td>59.80</td>
</tr>
<tr>
<td>606.00</td>
<td>1206.00</td>
<td>906.00</td>
<td>90</td>
<td>18.00</td>
<td>389</td>
<td>77.80</td>
</tr>
<tr>
<td>1206.00</td>
<td>1806.00</td>
<td>1506.00</td>
<td>54</td>
<td>10.80</td>
<td>443</td>
<td>88.60</td>
</tr>
<tr>
<td>1806.00</td>
<td>2406.00</td>
<td>2106.00</td>
<td>42</td>
<td>8.40</td>
<td>485</td>
<td>97.00</td>
</tr>
<tr>
<td>2406.00</td>
<td>3006.00</td>
<td>2706.00</td>
<td>11</td>
<td>2.20</td>
<td>496</td>
<td>99.20</td>
</tr>
<tr>
<td>3006.00</td>
<td>3606.00</td>
<td>3306.00</td>
<td>4</td>
<td>0.80</td>
<td>500</td>
<td>100.00</td>
</tr>
</tbody>
</table>

TOTAL 500 100.00

***********************************************************************************************

PARAMETERS OF POPULATION

MEAN 771.80
MEDIAN 507.67
RANGE 3600.00
VARIANCE 25439.80
STAND DEV 159.50

***********************************************************************************************


SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPlications MUST BE NO GREATER THAN 50.

109 56

59
TRY AGAIN.
SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GREATER THAN 50.

10 20

ENTER THE VARIABLE NAME OF ONE STATISTIC OF YOUR INTEREST.
The variable names are as follows:

<table>
<thead>
<tr>
<th>STATISTICS</th>
<th>VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN</td>
<td>SMS</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>SMS</td>
</tr>
<tr>
<td>RANGE</td>
<td>SSDS</td>
</tr>
<tr>
<td>VARIANCE</td>
<td>SYS</td>
</tr>
<tr>
<td>STAND DEV</td>
<td>SSDS</td>
</tr>
</tbody>
</table>

SMS

PARAMETERS OF SAMPLING DISTRIBUTION

<table>
<thead>
<tr>
<th>STATISTIC</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN</td>
<td>744.00</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>756.00</td>
</tr>
<tr>
<td>RANGE</td>
<td>780.00</td>
</tr>
<tr>
<td>VARIANCE</td>
<td>86276.00</td>
</tr>
<tr>
<td>STAND DEV</td>
<td>257.44</td>
</tr>
</tbody>
</table>

SAMPLING DISTRIBUTION OF THE STATISTIC IS PRESENTED IN THE FORM OF PROBABILITY DISTRIBUTION AND GRAPH Below:
FIRST, ENTER THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, CLASS WIDTH, AND NO. OF CLASSES YOU NEED FOR THE SAMPLING DISTRIBUTION, E.G., 0 5 10:
(Maximum value of the statistic is: 1465; the minimum is: 366):

400 200 5

SAMPLING DISTRIBUTION

<table>
<thead>
<tr>
<th>LOWER CL LIMIT</th>
<th>UPPER CL LIMIT</th>
<th>CLASS MID POINT</th>
<th>PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>400.00</td>
<td>600.00</td>
<td>500.00</td>
<td>0.15</td>
</tr>
<tr>
<td>600.00</td>
<td>800.00</td>
<td>700.00</td>
<td>0.30</td>
</tr>
<tr>
<td>800.00</td>
<td>1000.00</td>
<td>900.00</td>
<td>0.20</td>
</tr>
<tr>
<td>1000.00</td>
<td>1200.00</td>
<td>1100.00</td>
<td>0.20</td>
</tr>
<tr>
<td>1200.00</td>
<td>1400.00</td>
<td>1300.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>0.85</td>
</tr>
</tbody>
</table>

Would you like to develop the sampling distribution for a new statistic? N
Would you like to respecify sample size and/or no. of replications? N

Do you like to continue exercise with a new population? N
A POPULATION DATA TO BE ENTERED AS INPUT MAY BE IN THE FORM
OF A VECTOR OF INDIVIDUAL VALUES (UNGROUPED DATA) OR IN THE FORM
OF FREQUENCY (OR PROBABILITY) DISTRIBUTIONS. PLEASE FOLLOW THE INPUT
INSTRUCTIONS BELOW CAREFULLY.

HAVE YOU ALREADY ENTERED THE POPULATION DATA? TYPE YES OR NO. N
ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:

0

IS YOUR DATA UNGROUPED? TYPE 'Y' OR 'N'. N
ENTER THE FREQUENCIES. IF THE DATA CONTAINS RELATIVE FREQUENCIES OR PROBABILITIES,
CONVERT THEM INTO ABSOLUTE FREQUENCIES AND ENTER THE ABSOLUTE FREQUENCIES:

0: 253 108 31 26 21 17 7 7 3 9 4 5 5 2 1 1

ENTER THE LOWER CLASS LIMIT OF THE FIRST CLASS, CLASS INTERVAL, AND NUMBER OF CLASSES IN
YOUR FREQUENCY DISTRIBUTION TABLE, I.E., 5 20 10.

0

***********************************************************************************************

<table>
<thead>
<tr>
<th>LOWER CL LMT</th>
<th>UPPER CL LMT</th>
<th>CLASS</th>
<th>FREQ</th>
<th>REL FREQ</th>
<th>CUMUL FREQ</th>
<th>REL CUM FREQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00 - 5.00</td>
<td>5.00 - 10.00</td>
<td>1.00</td>
<td>253</td>
<td>50.60</td>
<td>253</td>
<td>50.60</td>
</tr>
<tr>
<td>1.00 - 2.00</td>
<td>2.00 - 3.00</td>
<td>3.00</td>
<td>108</td>
<td>21.60</td>
<td>361</td>
<td>72.20</td>
</tr>
<tr>
<td>2.00 - 2.00</td>
<td>2.50 - 3.50</td>
<td>4.00</td>
<td>31</td>
<td>6.20</td>
<td>392</td>
<td>78.40</td>
</tr>
<tr>
<td>3.00 - 3.00</td>
<td>3.50 - 4.50</td>
<td>5.00</td>
<td>26</td>
<td>5.20</td>
<td>418</td>
<td>83.60</td>
</tr>
<tr>
<td>4.00 - 4.00</td>
<td>4.50 - 5.50</td>
<td>6.00</td>
<td>4</td>
<td>0.80</td>
<td>429</td>
<td>87.80</td>
</tr>
<tr>
<td>5.00 - 5.00</td>
<td>5.50 - 6.50</td>
<td>7.00</td>
<td>17</td>
<td>3.40</td>
<td>456</td>
<td>91.20</td>
</tr>
<tr>
<td>6.00 - 6.00</td>
<td>6.50 - 7.50</td>
<td>8.00</td>
<td>7</td>
<td>1.40</td>
<td>470</td>
<td>94.00</td>
</tr>
<tr>
<td>7.00 - 7.00</td>
<td>7.50 - 8.50</td>
<td>9.00</td>
<td>3</td>
<td>0.60</td>
<td>473</td>
<td>94.60</td>
</tr>
<tr>
<td>8.00 - 8.00</td>
<td>8.50 - 9.50</td>
<td>10.00</td>
<td>6</td>
<td>1.20</td>
<td>489</td>
<td>96.00</td>
</tr>
<tr>
<td>9.00 - 9.00</td>
<td>9.50 - 10.50</td>
<td>11.00</td>
<td>4</td>
<td>0.80</td>
<td>486</td>
<td>96.80</td>
</tr>
<tr>
<td>10.00 - 10.00</td>
<td>10.50 - 11.50</td>
<td>12.00</td>
<td>5</td>
<td>1.00</td>
<td>491</td>
<td>98.20</td>
</tr>
<tr>
<td>11.00 - 11.00</td>
<td>11.50 - 12.50</td>
<td>13.00</td>
<td>6</td>
<td>1.00</td>
<td>496</td>
<td>99.20</td>
</tr>
<tr>
<td>12.00 - 12.00</td>
<td>12.50 - 13.50</td>
<td>14.00</td>
<td>2</td>
<td>0.40</td>
<td>498</td>
<td>99.60</td>
</tr>
<tr>
<td>13.00 - 13.00</td>
<td>13.50 - 14.50</td>
<td>15.00</td>
<td>1</td>
<td>0.20</td>
<td>499</td>
<td>99.80</td>
</tr>
<tr>
<td>14.00 - 14.00</td>
<td>14.50 - 15.50</td>
<td>16.00</td>
<td>1</td>
<td>0.20</td>
<td>500</td>
<td>100.00</td>
</tr>
</tbody>
</table>

TOTAL
500
100.00

***********************************************************************************************

PARAMETERS OF POPULATION

<table>
<thead>
<tr>
<th>MEAN</th>
<th>MEDIAN</th>
<th>RANGE</th>
<th>VARIANCE</th>
<th>STAND DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.15</td>
<td>0.99</td>
<td>16.00</td>
<td>1.78</td>
<td>1.33</td>
</tr>
</tbody>
</table>

***********************************************************************************************

SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GREATER THAN 50.

0: 30 30
ENTER THE VARIABLE NAME OF ONE STATISTIC OF YOUR INTEREST.
THE VARIABLE NAMES ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>STATISTICS</th>
<th>VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN</td>
<td>SMS</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>SMDS</td>
</tr>
<tr>
<td>RANGE</td>
<td>SRS</td>
</tr>
<tr>
<td>VARIANCE</td>
<td>SVS</td>
</tr>
<tr>
<td>STAND DEV</td>
<td>SSDS</td>
</tr>
</tbody>
</table>

\[ \text{SMS} \]

PARAMETERS OF SAMPLING DISTRIBUTION

<table>
<thead>
<tr>
<th>AMN</th>
<th>2.06</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEDIAN</td>
<td>2.00</td>
</tr>
<tr>
<td>RANGE</td>
<td>1.83</td>
</tr>
<tr>
<td>VARIANCE</td>
<td>0.20</td>
</tr>
<tr>
<td>STAND DEV</td>
<td>0.44</td>
</tr>
</tbody>
</table>

SAMPLING DISTRIBUTION OF THE STATISTIC IS PRESENTED IN THE FORM OF PROBABILITY DISTRIBUTION AND GRAPH BELOW:

BUT, FIRST, ENTER THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, CLASS WIDTH, AND NO. OF CLASSES YOU NEED FOR THE SAMPLING DISTRIBUTION, E.G., 0.5, 10:

(MAXIMUM VALUE OF THE STATISTIC IS: 2.766666667 THE MINIMUM IS: 0.9333333333):

\[ \text{.5 .5 6} \]

SAMPLING DISTRIBUTION

<table>
<thead>
<tr>
<th>LOWER CLASS LIMIT</th>
<th>UPPER CLASS LIMIT</th>
<th>MIDPOINT</th>
<th>PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>1.00</td>
<td>0.75</td>
<td>0.03</td>
</tr>
<tr>
<td>1.00</td>
<td>1.50</td>
<td>1.25</td>
<td>0.03</td>
</tr>
<tr>
<td>1.50</td>
<td>2.00</td>
<td>1.75</td>
<td>0.43</td>
</tr>
<tr>
<td>2.00</td>
<td>2.50</td>
<td>2.25</td>
<td>0.30</td>
</tr>
<tr>
<td>2.50</td>
<td>3.00</td>
<td>2.75</td>
<td>0.20</td>
</tr>
<tr>
<td>3.00</td>
<td>3.50</td>
<td>3.25</td>
<td>0.00</td>
</tr>
</tbody>
</table>

TOTAL 1.00

WOULD YOU LIKE TO DEVELOP THE SAMPLING DISTRIBUTION FOR A NEW STATISTIC? N
WOULD YOU LIKE TO RESPECIFY SAMPLE SIZE AND/OR NO. OF REPLICATIONS? N

DO YOU LIKE TO CONTINUE EXERCISE WITH A NEW POPULATION? N
PROGRAM DESCRIPTION

V POPDIST;X;CW;F;FORM;K;LC;LCM;PARAMPM;PMD;PR;PV;PSD;RCF;RF;UC
[1]*THIS FUNCTION REQUIRES THE FOLLOWING PUBLIC FUNCTIONS:
[2]*AKI, CENTER, COLNAMES, ΔFMT, PLOT, AND VS.
[3]*TEST: 'ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:
[4]*EN=integer
[5]*((EN<100), (EN<100)) / TEST, GOOD
[6]*GOD: ANS=AKI 'IS YOUR DATA UNGROUPED? TYPE ''Y'' OR ''N''.'
[7]*((Y<ANS), ((N<ANS)) / L1, L2
[8]*L1: 'ENTER UNGROUPED DATA. TYPE '',0'' IF MORE SPACE IS NEEDED FOR ENTERING DA
[9]*X
[10]*'
[11]*MAXIMUM VALUE IS: ''/X; MINIMUM VALUE IS: ''/X
[12]*IN ORDER TO SHAPE A FREQUENCY DISTRIBUTION FOR THE UNGROUPED DATA
[13]*YOU ENTERED, YOU ARE ASKED TO SPECIFY THE LOWER CLASS LIMIT VALUE
[14]*OF THE FIRST CLASS, WIDTH OF CLASS, AND NUMBER OF CLASSES. E.G.:'
[15]*' 5 20 10. ENTER THESE THREE VALUES:'
[16]*
[17]*F
[18]*L2: 'ENTER THE GROUPED DATA AS FOLLOWS: IF THE DATA CONTAINS ABSOLUTE'
[19]*'FREQUENCIES, ENTER THE FREQUENCIES. IF THE DATA CONTAINS RELATIVE'
[20]*'FREQUENCIES OR PROBABILITIES, CONVERT THEM INTO ABSOLUTE FREQUENCIES'
[21]*'AND ENTER THE ABSOLUTE FREQUENCIES:'
[22]*
[23]*F
[24]*'ENTER THE LOWER CLASS LIMIT OF THE FIRST CLASS, CLASS INTERVAL, AND'
[25]*'NUMBER OF CLASSES IN YOUR FREQUENCY DISTRIBUTION TABLE, I.E.:'
[26]*' 5 20 10.'
[27]*
[28]*
[29]*
[30]*
[31]*
[32]*
[33]*
[34]*
[35]*
[36]*
[37]*
[38]*
[39]*
[40]*
[41]*
[42]*
[43]*
[44]*
[45]*
[46]*
[47]*
[48]*
[49]*
[50]*
[51]*
[52]*
[53]*
[54]*
[55]*
[56]*
[57]*

63
THIS FUNCTION REQUIRES THE FOLLOWING PUBLIC FUNCTIONS:

AAYN, CENTER, COLNAMES, AND AFMT

THE VARIABLE NAMES ARE AS FOLLOWS:

'X20, 9A1, X12, 9A1' COLNAMES 'STATISTICS/VARIABLE/'

'X20, 9A1, X12, 9A1' AAYN((5 9 pNAME);(5 9 'SMS
SMD SRS SVS
SDSD '))

XX *=

SDS+(SDV+/(XX-SDM+/(XX)+S[2])*2)+S[2]*0.5

SMD+0.5*XX(S[(XX)][|1 0.5 0.5*1+DXX]]

SDE=[XXX]

STATS=SDM, SMD, SDR, SDS, SDD

STAR

'X10, 9A1,F20.2' CENTER 'PARAMETERS OF SAMPLING DISTRIBUTION'

'X10, 9A1,F20.2' AFMT((5 9 pNAME);STATS)

STAR

'SAMPLING DISTRIBUTION OF THE STATISTIC IS PRESENTED IN THE FORM'

'OF PROBABILITY DISTRIBUTION AND GRAPH BELOW:'

'BUT, FIRST, ENTER THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, CLASS WIDTH, AND NO. OF'

'CLASSES YOU NEED FOR THE SAMPLING DISTRIBUTION, E.G., 0 5 10:'

'MAXIMUM VALUE OF THE STATISTIC IS: ;'X/XX;' THE MINIMUM IS: 'L/XX;'

R'=

SDV+(R[1]+0.5*R[2])*SCW+(((R[3]-R[1])*R[2])


STAR

FORM1='X10,4F14.2'

FORM1 CENTER 'SAMPLING DISTRIBUTION'

FORM1 COLNAMES '/ LOWER/ UPPER/ CLASS/ PROB/'

FORM1 COLNAMES '/CL LMT/CL LMT/MID PT/'

FORM1 AFMT(SLC; SUC; SDY; PROB) O ' '

'X47, TOTAL,F14.2' AFMT(+/PROB)

STAR

HS=1

PC='I'

20 40 PLOT PROB VS SDY

STAR

'WOULD YOU LIKE TO DEVELOP THE SAMPLING DISTRIBUTION FOR A
NEW STATISTIC?'
\begin{verbatim}
* SIMUSAMPLE;ANS;CF;I;J;P;FN;PNAME;S;SAMPLE;SM;SMD;SR;SV;SSD;SMS;SMDS;SRS;
  SVS;SSDS;STAR;SY;Y
[1] * THIS FUNCTION REQUIRES THE FUNCTIONS AKI, AYN, POPDIST, AND SAMPDIST.
[2] 'A POPULATION DATA TO BE ENTERED AS INPUT MAY BE IN THE FORM
  OF A VECTOR OF INDIVIDUAL VALUES (UNGROUPED DATA) OR IN THE FORM
  OF FREQUENCY (OR PROBABILITY) DISTRIBUTIONS. PLEASE FOLLOW THE INPUT
  INSTRUCTIONS BELOW CAREFULLY.'
[3] *SAM\textbackslash NAMES* AYN 'HAVE YOU ALREADY ENTERED THE POPULATION DATA? TYPE YES OR
  NO.'
  0 30.'
[6] 'SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GRE
  ATER THAN 50.'
[7] SY\textbackslash
[9] ' TRY AGAIN.'
[10] *SAM
[12] SMS=SMDS=SRS=SV=SSDS=1 0
[13] REPL: I=I+1
[14] SAMPE: I=0
[15] J=0
[16] SAMP: J=J+1
[17] AN+?spFN
[18] SY\textbackslash(((J<1)(RN[J;I]\leq CF)/(CF[I])
[19] SAMPLE=SAMPLE,SY
[20] *{(J<S[I])/SAMP
[21] SSD=(SV+\{(SAMPLE-SM+(+/SAMPLE)*S[I])+2)*S[I]-1)*
  0.5
[22] SMD=0.5*+/SAMPLE[(\{SAMPLE[I][1]-0.5*0.5*1+/SAMPLE)]]
[23] SR+1/SAMPLE-1/SAMPLE
[24] SMS=SMDS\textbackslash
[25] SMDS=SMDS,SMD
[26] SRS=SRS,SR
[27] SVS=SVS,SV
[28] SDDS=SDDS,SSD
[29] *(I<S[I])/REPL
[30] SAMPDIST
[31] *SAM\textbackslash NAMES* AYN 'WOULD YOU LIKE TO RESPECIFY SAMPLE SIZE AND/OR NO. OF REPL
  ICATIONS? DO 2 1 p'
[32] ANS=AKI 'WOULD YOU LIKE TO CONTINUE EXERCISE WITH A NEW POPULATION?'
[33] *(('Y'\textbackslash NAMES)','N'\textbackslash NAMES)/POP,0
\end{verbatim}
3.6

SETTING THE SIZE OF THE SALES FORCE

Jack R. Dauner
Associate Professor of Marketing
College of Business Administration
The University of Akron
Akron, Ohio

DESCRIPTION

This program facilitates the analysis and determination of the number of salesmen to employ in each sales district and hence in a company's entire sales force. The original mathematical model was developed by Professor Zarrel V. Lambert and published in the book Setting the Size for the Sales Force by the Center for Research of the College of Business Administration, The Pennsylvania State University (1968). The model is based on a case study which uses a company that manufactures and markets medical and industrial X-Ray equipment, accessories, and supplies.

The program SALESFORCE can be used to determine the number of salesmen and the estimated change in the company's gross profits where

\[ ECP = \text{Estimated change in the company's gross profits less the change in personal selling costs including commissions.} \]

\[ R = \text{Salesmen's commission rate based on gross profits.} \]

\[ MUI = \text{Gross markup percentage on product cost in district.} \]

\[ PMI = \text{Product mix ratio in district.} \]

\[ PI = \text{Selling price index in district.} \]

\[ SI = \text{Number of Salesmen employed by the company in district.} \]

\[ C = \text{Average out-of-pocket costs per salesmen.} \]
\( N \) = Number of salesmen by which the sales force in the district is to be increased or decreased.

\( N_{LOW} \) = Minimum number of salesmen in district.

\( N_{MAX} \) = Maximum number of salesmen in the district.

To begin the use of the program, type SALESFORCE and then enter the inputs as requested. The number of values entered (e.g., the number of sales districts) must be the same for each of the questions or an error condition will result. In the sample shown in the following section, there were seven districts in the company, so seven values had to be entered for each question. In the last question, the number of salesmen to be evaluated is entered.
PROGRAM USAGE

ENTER SALESMAN COMMISSION RATE BASED ON GROSS PROFITS
  .05 .05 .05 .05 .05 .05

ENTER GROSS MARKUP PERCENTAGE ON PRODUCT COST IN DISTRICT
  9 13 12 10 9 4 12

ENTER PRODUCT MIX RATIO IN DISTRICT
  51.29 13.40 42.03 25.72 19.91 33.48 38.79

ENTER SELLING PRICE INDEX IN DISTRICT
  109 113 112 110 109 104 112

ENTER NUMBER OF SALES MEN EMPLOYED IN DISTRICT
  6 2 5 9 4 4 1

ENTER AVERAGE OUT-OF-POCKET COSTS PER SALESMAN
  8269 8269 8269 8269 8269 8269 8269

ENTER RANGE OF SALESMEN TO TEST (ENTER MINIMUM AND MAXIMUM)
  1 10

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26779946.13</td>
</tr>
<tr>
<td>2</td>
<td>29731047.23</td>
</tr>
<tr>
<td>3</td>
<td>3260351.61</td>
</tr>
<tr>
<td>4</td>
<td>3540741.77</td>
</tr>
<tr>
<td>5</td>
<td>38151481.51</td>
</tr>
<tr>
<td>6</td>
<td>40842123.54</td>
</tr>
<tr>
<td>7</td>
<td>43484719.58</td>
</tr>
<tr>
<td>8</td>
<td>46083373.21</td>
</tr>
<tr>
<td>9</td>
<td>48642625.28</td>
</tr>
<tr>
<td>10</td>
<td>51165503.90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1581457.77</td>
</tr>
<tr>
<td>2</td>
<td>1975811.33</td>
</tr>
<tr>
<td>3</td>
<td>2348825.47</td>
</tr>
<tr>
<td>4</td>
<td>2705654.73</td>
</tr>
<tr>
<td>5</td>
<td>3049551.02</td>
</tr>
<tr>
<td>6</td>
<td>3382742.07</td>
</tr>
<tr>
<td>7</td>
<td>3706841.34</td>
</tr>
<tr>
<td>8</td>
<td>4023067.22</td>
</tr>
<tr>
<td>9</td>
<td>4332369.63</td>
</tr>
<tr>
<td>10</td>
<td>4635508.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20216653.97</td>
</tr>
<tr>
<td>2</td>
<td>22809048.04</td>
</tr>
<tr>
<td>3</td>
<td>25322386.26</td>
</tr>
<tr>
<td>4</td>
<td>27766583.72</td>
</tr>
<tr>
<td>5</td>
<td>30156637.90</td>
</tr>
<tr>
<td>6</td>
<td>32493563.78</td>
</tr>
<tr>
<td>7</td>
<td>34784971.17</td>
</tr>
<tr>
<td>8</td>
<td>37035439.74</td>
</tr>
<tr>
<td>9</td>
<td>39248772.81</td>
</tr>
<tr>
<td>10</td>
<td>41428174.89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23562427.73</td>
</tr>
<tr>
<td>2</td>
<td>25384674.28</td>
</tr>
<tr>
<td>3</td>
<td>27171391.88</td>
</tr>
<tr>
<td>4</td>
<td>28921655.15</td>
</tr>
<tr>
<td>5</td>
<td>30651932.71</td>
</tr>
<tr>
<td>6</td>
<td>32351225.83</td>
</tr>
<tr>
<td>7</td>
<td>34026168.14</td>
</tr>
<tr>
<td>8</td>
<td>35678599.28</td>
</tr>
<tr>
<td>9</td>
<td>37310120.24</td>
</tr>
<tr>
<td>10</td>
<td>38922135.84</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4940102.33</td>
</tr>
<tr>
<td>2</td>
<td>5692961.91</td>
</tr>
<tr>
<td>3</td>
<td>6418860.62</td>
</tr>
<tr>
<td>4</td>
<td>712241.86</td>
</tr>
<tr>
<td>5</td>
<td>7807069.24</td>
</tr>
<tr>
<td>6</td>
<td>8475282.77</td>
</tr>
<tr>
<td>7</td>
<td>9129062.77</td>
</tr>
<tr>
<td>8</td>
<td>9769992.88</td>
</tr>
<tr>
<td>9</td>
<td>10399365.96</td>
</tr>
<tr>
<td>10</td>
<td>11018255.68</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4564756.18</td>
</tr>
<tr>
<td>2</td>
<td>5259882.28</td>
</tr>
<tr>
<td>3</td>
<td>5930092.57</td>
</tr>
<tr>
<td>4</td>
<td>6579678.21</td>
</tr>
<tr>
<td>5</td>
<td>7211747.70</td>
</tr>
<tr>
<td>6</td>
<td>782648.36</td>
</tr>
<tr>
<td>7</td>
<td>8432210.32</td>
</tr>
<tr>
<td>8</td>
<td>9023897.08</td>
</tr>
<tr>
<td>9</td>
<td>9604903.44</td>
</tr>
<tr>
<td>10</td>
<td>10216221.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2190749.47</td>
</tr>
<tr>
<td>2</td>
<td>3006285.05</td>
</tr>
<tr>
<td>3</td>
<td>3763592.10</td>
</tr>
<tr>
<td>4</td>
<td>4480323.57</td>
</tr>
<tr>
<td>5</td>
<td>5166278.42</td>
</tr>
<tr>
<td>6</td>
<td>5827642.33</td>
</tr>
<tr>
<td>7</td>
<td>6468650.50</td>
</tr>
<tr>
<td>8</td>
<td>7092370.91</td>
</tr>
<tr>
<td>9</td>
<td>7701120.26</td>
</tr>
<tr>
<td>10</td>
<td>8475282.77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>ESTIMATED NET CHANGE IN PROFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4332369.63</td>
</tr>
<tr>
<td>2</td>
<td>4635508.22</td>
</tr>
<tr>
<td>3</td>
<td>4940102.33</td>
</tr>
<tr>
<td>4</td>
<td>5259882.28</td>
</tr>
<tr>
<td>5</td>
<td>5692961.91</td>
</tr>
<tr>
<td>6</td>
<td>6418860.62</td>
</tr>
<tr>
<td>7</td>
<td>712241.86</td>
</tr>
<tr>
<td>8</td>
<td>7807069.24</td>
</tr>
<tr>
<td>9</td>
<td>8475282.77</td>
</tr>
<tr>
<td>10</td>
<td>9129062.77</td>
</tr>
</tbody>
</table>

69
PROGRAM DESCRIPTION

V SALESFORCE;ECP;ECP1;ECP2;ECP3;ECP4;ECP5;MUI;N;NLOW;NMAX;PI;PMI;R;SI
[1] THIS PROGRAM REQUIRES THE PUBLIC FUNCTION AFMT
[2] 'ENTER SALESMAN COMMISSION RATE BASED ON GROSS PROFITS' O R+D
[3] 'ENTER GROSS MARKUP PERCENTAGE ON PRODUCT COST IN DISTRICT' O MUI*D
[4] 'ENTER PRODUCT MIX RATIO IN DISTRICT' O PMI*D
[5] 'ENTER SELLING PRICE INDEX IN DISTRICT' O PI*D
[6] 'ENTER NUMBER OF SALESMEN EMPLOYED IN DISTRICT' O SI*D
[7] 'ENTER AVERAGE OUT-OF-POCKET COSTS PER SALESMAN' O C*D
[8] 'ENTER RANGE OF SALESMEN TO TEST (ENTER MINIMUM AND MAXIMUM)' O IN*D
[10] N=IN[1]
[12] A: ECP1=409220750
[13] ECP2+PMI*1.169518
[14] ECP3+PI*-2.645349
[15] ECP4+(SI+N)*0.7847286
[16] ECP5+SI*0.7847286
[17] ECP6+((1-R)*MUI*ECP1*ECP2*ECP3*ECP4*ECP5)-(C*N)
[18] ECP=ECP,N,ECP6
[19] N=N+1
[20] '+IN+NMIX+1)*/A
[21] ' N ESTIMATED NET CHANGE IN PROFITS'
[22] ECP=(NMIX,(1+pR))*ECP
[23] ('I3' AFMT ECP[;1]);'F12.2' AFMT ECP[;1+1)+ECP]
SIMULATION OF A THREE STAGE ORDER/SHIPMENT SYSTEM

John L. Harpell
Assistant Professor of Management
West Virginia University
Morgantown, West Virginia

DESCRIPTION

The accompanying function \textit{INVENTORY} is a simple simulation exercise based upon the type of format used by Jay Forrester in his book \textit{Industrial Dynamics}. The principal use of this program (or function) is to show the concept of simulation and the effects of manipulating decision rules to undergraduate business students. However, it is likely to be of interest to anyone desiring an understanding of simulation. While the program was developed during an introductory course in APL at UCLA, a strong attempt was made to keep the program independent of the UCLA system. Only the plotting ability is not self-contained and it can be ignored.

The system which is to be simulated is shown in Figure 1, where lags 1-3 represent delays from order placement to order shipment and lags 4-6 represent delays from order shipment to order receipt.

Internal to the program are the following variables and relationships:

<table>
<thead>
<tr>
<th>VARS</th>
<th>MEANING</th>
<th>COMPUTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number (period)</td>
<td>Time period counter</td>
</tr>
<tr>
<td>BI</td>
<td>Beginning Inventory</td>
<td>= Ending inventory of preceding period</td>
</tr>
<tr>
<td>DMND</td>
<td>Demand</td>
<td>Input data</td>
</tr>
<tr>
<td>SALES</td>
<td>Amount Sold</td>
<td>Smaller of BI or DMND</td>
</tr>
<tr>
<td>BAL</td>
<td>Balance in Stock</td>
<td>BI-SALES (or SHIP)</td>
</tr>
<tr>
<td>SHIP</td>
<td>Amount Shipped</td>
<td>Smaller of BI or Order Placed</td>
</tr>
<tr>
<td>VAR</td>
<td>MEANING</td>
<td>COMPUTATION</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>ORD</td>
<td>Order Placed</td>
<td>*200-BAL</td>
</tr>
<tr>
<td>REC</td>
<td>Amount Received</td>
<td>SHIP of LAG4,5, or 6 periods prior</td>
</tr>
<tr>
<td>EI</td>
<td>Ending Inventory</td>
<td>BAL+REC</td>
</tr>
</tbody>
</table>

* A somewhat arbitrary rule.

It is suggested that the first time the program is run, the user should ask for an explanation each time it is offered. This is done by typing a "2" when asked.

![Diagram](http://example.com/diagram.png)

Figure 1
PROGRAM USAGE

INVENTORY

FOR PROGRAM DESCRIPTION, SEE DESCRIBE FUNCTION

SET LAGS? (ENTER 0 FOR NO, 1 FOR YES, AND 2 FOR EXPLANATION.

2

ENTERING A 1 ALLOWS YOU TO CHANGE THE SIX DELAYS FOR THE ORDER TO BE SHIPPED FROM A) RETAIL TO WHOLESALE, B) WHOLESALE TO FACTORY, AND C) FACTORY WAREHOUSE AND WHOLESALE AND F) FACTORY AND FACTORY WAREHOUSE.

ENTER 1 FOR YES AND SIX RESPECTIVE NUMBERS (ENTER SPACES ONLY BETWEEN THEM). IF YOU ENTER 2 FOR NO, THE DEFAULT IS 2 4 0 0 0

SET LAGS? (ENTER 0 FOR NO, 1 FOR YES, AND 2 FOR EXPLANATION.

1

ENTER 6 LAGS

1 1 4 1 1 0

ENTER NUMBER OF PERIODS FOR PROGRAM TO RUN. (SUGGEST 10 MAX FOR FIRST TRIAL.)

10

SELECT ONE OF THE FOLLOWING OPTIONS AND ENTER THE CORRESPONDING NUMBER:

1 I WILL ENTER MY OWN SALES DATA.

2 USE AN EQUALLY LIKELY RANDOM NUMBER GENERATOR TO CREATE SALES DATA.

3 USE CONSTANT INCREASE/DECREASE RANDOM NUMBER GENERATOR TO CREATE SALES DATA.

4 WHAT ARE YOU TALKING ABOUT? (EXPLANATION)

4
PROGRAM DESCRIPTION

[1] FOR PROGRAM DESCRIPTION, SEE DESCRIBE FUNCTION.

[2] THIS IS THE MAIN PROGRAM.

[3] THE FOLLOWING VARIABLES ARE ASSOCIATED WITH THE NUMBERS USED IN THE

[4] PLOTTING CALLS:

[5] RBI C FDMND + 2 C RSALES + 3 C REAL + 4 C FORD + 5 C REC + 6 C REI + 7

[6] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[7] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[8] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[9] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[10] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[11] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[12] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[13] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[14] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[15] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[16] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[17] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[18] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[19] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[20] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[21] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[22] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[23] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[24] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[25] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[26] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[27] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[28] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[29] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[30] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[31] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[32] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[33] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[34] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[35] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[36] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[37] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[38] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[39] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[40] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[41] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[42] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[43] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[44] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[45] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[46] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[47] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[48] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[49] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[50] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[51] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[52] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[53] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[54] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[55] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[56] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,

[57] XSHIP C WAREHOUSE, XSALES C SALES, XDFN C FDMND, XTALE C PTALE,
THIS FUNCTION DETERMINES AND FEEDS SALES DATA TO MAIN FUNCTION(INVNT). E

SELECT ONE OF THE FOLLOWING OPTIONS AND ENTER THE CORRESPONDING NUMBER:

1. I WILL ENTER MY OWN SALES DATA.
2. USE AN EQUALLY LIKELY RANDO NUMBER GENERATOR TO CREATE SALES DATA.
3. USE CONSTANT INCREASE/DECREASE RANDOM NUMBER GENERATOR TO CREATE SALES DATA.
4. WHAT ARE YOU TALKING ABOUT? (EXPLANATION)

TYPESLS+D

+((NOPFS<pSLSV),(NOPFS=pSLSV),(NOPFS>pSLSV))/L1.L2.L3

L1:'TOO MUCH DATA. TRY AGAIN................'
L2:'TOO LITTLE DATA. GIVE IT ANOTHER TRY**********'
L3:'ENTER SALES DATA FOR 'NOPFS' PERIODS. (E.G. 110 110 90 90 )'
SLSV=0

CHECK "MANUALLY INPUT DATA.

+((NOPFS<pSLSV),(NOPFS=pSLSV),(NOPFS>pSLSV))/L1.L2.L3

L1:'TOO MUCH DATA. TRY AGAIN.............'
L2:'TOO LITTLE DATA. GIVE IT ANOTHER TRY**********'
L3:'ENTER SALES DATA FOR 'NOPFS' PERIODS. (E.G. 110 110 90 90 )'
SLSV=0

CHECK "MANUALLY INPUT DATA.

+((NOPFS<pSLSV),(NOPFS=pSLSV),(NOPFS>pSLSV))/L1.L2.L3

L1:'TOO MUCH DATA. TRY AGAIN.............'
L2:'TOO LITTLE DATA. GIVE IT ANOTHER TRY**********'
L3:'ENTER SALES DATA FOR 'NOPFS' PERIODS. (E.G. 110 110 90 90 )'
SLSV=0

CHECK "MANUALLY INPUT DATA.

+((NOPFS<pSLSV),(NOPFS=pSLSV),(NOPFS>pSLSV))/L1.L2.L3

L1:'TOO MUCH DATA. TRY AGAIN.............'
L2:'TOO LITTLE DATA. GIVE IT ANOTHER TRY**********'
L3:'ENTER SALES DATA FOR 'NOPFS' PERIODS. (E.G. 110 110 90 90 )'
SLSV=0

CHECK "MANUALLY INPUT DATA.

+((NOPFS<pSLSV),(NOPFS=pSLSV),(NOPFS>pSLSV))/L1.L2.L3

L1:'TOO MUCH DATA. TRY AGAIN.............'
L2:'TOO LITTLE DATA. GIVE IT ANOTHER TRY**********'
L3:'ENTER SALES DATA FOR 'NOPFS' PERIODS. (E.G. 110 110 90 90 )'
SLSV=0

CHECK "MANUALLY INPUT DATA.
'UPSET THIS EQUILIBRIUM. YOU HAVE SEVERAL OPTIONS FOR DETERMINING THE
'CUSTOMER DEMAND, NAMELY:

1. IF YOU ENTER A 1, THE PROGRAM WILL REQUIRE YOU TO FEED IN
   THE DEMAND FIGURES FOR EACH OF THE PERIODS. YOU MUST ENTER
   A FEW OF FIGURES (SEPARATED BY SPACES) EQUAL IN NUMBER TO
   THE NUMBER OF PERIODS YOU HAVE ALREADY ENTERED.

2. IF YOU ENTER A 2, THE COMPUTER WILL GENERATE A RANDOM NUMBER
   FROM A CONSTANT (OR EQUALLY LIKELY) PROBABILITY DISTRIBUTION FOR
   SALES BETWEEN 90 AND 110. TO CHANGE 90 AND 110 FIND THE PROGRAM
   LINE "PAND" IN THE FUNCTION SIMSALE AND CHANGE THE 89 TO
   "(MINT-1)" AND THE 11 TO "(RANGE+1)".

3. IF YOU ENTER A 3, THE COMPUTER WILL GENERATE A RANDOM NUMBER
   FROM A CONSTANTLY INCREASING (FROM 90 TO 100) AND CONSTANTLY
   DECREASING (FROM 100 TO 110) PROBABILITY DISTRIBUTION. TO
   CHANGE TO 90, 100, AND 110, FIND THE LINE "JDCFSE" IN SIMSALE
   AND THEN CHANGE THE 100 TO "MID-RANGE" AND THE 11 TO
   "RANGE/2 + 1".

4. NOW YOU KNOW...***

NOW YOU WANT TO PLOT ANY RESULTS? ENTER O FOR NO; 1 FOR YES; 2 FOR EXPLANATION.

YES; 'HOW MANY VARIABLES DO YOU WANT TO PLOT? (MAXIMUM OF 3 )'

NUMB=0

USER COLUMN NUMBERS IN THE PRECEDING TABLE TO ENTER THE
'VARIABLES FOR THE HORIZONTAL AXIS (VERTICAL AXIS IS ONLY ONE VARIABLE).'
"(F.G. FOR RETAIL PI, USE 1; FOR FACTORY WAREHOUSE, USE 19)."

USE THE COLUMN NUMBERS IN THE PRECEDING TABLE TO ENTER THE
'VARIABLES FOR THE HORIZONTAL AXIS (VERTICAL AXIS IS ONLY ONE VARIABLE).'
"(F.G. FOR RETAIL PI, USE 1; FOR FACTORY WAREHOUSE, USE 19)."
THREE: 'WHAT ARE THE THREE VARIABLES? (ENTER WITH SPACES ONLY BETWEEN.)'

TROIS =

THREE = (3 * TROIS) / THREE


UCLA-NEXT FIVE LINES.

'0 REPRESENTS 'G'

'* REPRESENTS 'H'

'* REPRESENTS 'I'

'READ THESE VARIABLES ON THE LEFT AXIS.'

G + TROIS[1] 0
H + TROIS[2]
I + TROIS[3]

UCLA-NEXT LINES.

'0 REPRESENTS 'G'

'* REPRESENTS 'H'

'* REPRESENTS 'I'

'READ THESE VARIABLES ON THE LEFT AXIS.'

G + TROIS[1] 0
H + TROIS[2]
I + TROIS[3]

UCLA-NEXT EQUH LINES

'0 REPRESENTS 'G'

'* REPRESENTS 'H'

'* REPRESENTS 'I'

'READ THESE VARIABLES ON THE LEFT AXIS.'

G + TROIS[1] 0
H + TROIS[2]
I + TROIS[3]

UCLA-NEXT LINES.

EXPLN: 'THIS FUNCTION (SIMPLOT) UTILIZES THE PLOTFORMAT ROUTINE IN'

'LIBRARY 1 AT UCLA. TO USE, YOU WILL NEED TO HAVE ALREADY LOADED THE'

'PLOTFORMAT FUNCTION INTO YOUR WORKSPACE. YOU MAY BE ABLE TO USE'

'PLOTFORMATS FROM OTHER LIBRARIES BY CALLING PROGRAM LINES MARKED BY A'

EXPLN: 'THREE VARIABLS TO BE PLOTTED ON HORIZONTAL AXIS.'

UNE =

UNE = (1 + UNE) / ONE

UNE = (1 + UNE) / ONE

UCLA

ONE: PFFT+1

UCLA

(20,NN) PLOT MPRINT[;G] AND MPRINT[;H] VS MPRINT[;21]

UCLA

(20,NN) PLOT MPRINT[;G] AND MPRINT[;H] VS MPRINT[;21]

UCLA

(20,NN) PLOT MPRINT[;G] AND MPRINT[;21]

UCLA
MULTIVARIATE T-SQUARE TEST

Edgar P. Hickman
Professor and Program Director of Management Science
University of South Carolina
Columbia, South Carolina 29208

DESCRIPTION

The program MULTEST is designed to test a hypothesis concerning a vector of means using Hotelling's $T^2$ statistic. It is designed for instructional use in an intermediate or advanced level statistical methods course covering multivariate statistical methods.

The user is required to specify the hypothesis to be tested and to input the sample data. The program then computes and prints the sample variance-covariance matrix and its inverse. Finally the program computes and prints Hotelling's $T^2$ statistic and the associated $F$ value.

Numerical examples of the multivariate $T^2$ test concerning means may be found in the following references:


Instructions for Use

In order to use MULTEST the user must copy MULTEST into his workspace as well as the supporting functions COLNAMES, DATAMAT, INPUT, RWTD, and ΔFMT. DATAMAT and INPUT are functions written especially for use with MULTEST and should always be saved in the
same library.

*MULTEST* is a niladic function which supplies instructions to the user at each step. Once the functions have been copied into the user's workspace, the function is activated by simply typing *MULTEST* and striking the return key.
PROGRAM USAGE

MULTEST
THIS PROGRAM TESTS THE HYPOTHESIS THAT A VECTOR OF MEANS IS EQUAL TO
A VECTOR OF CONSTANTS. FOR EXAMPLE:

HO: [U(1)=100, U(2)=75, U(3)=543,...,U(P)=25].

WHAT IS THE NUMBER OF VARIABLES FOR THIS PROBLEM?

3

ENTER THE VALUES FOR YOUR NULL HYPOTHESIS BELOW - ONE VALUE ON EACH LINE.

U(1)=
10
U(2)=
20
U(3)=
30

THE HYPOTHESIS YOU WISH TO TEST IS
U(1) =10
U(2) =20
U(3) =30

ENTER YOUR DATA WITH ONE OBSERVATION VECTOR ON EACH ROW. EACH ROW
SHOULD CONTAIN 3 VARIABLES. BELOW THE LAST ROW TYPE: STOP

15 16 40
14 11 38
18 9 38
17 16 37
14 12 31
9 15 37
15 16 39
13 18 33
16 12 35
18 11 38
STOP
YOUR DATA MATRIX CONTAINS 10 ROWS AND 3 COLUMNS. PLEASE CHECK TO SEE THAT IT IS CORRECT.

1 15 16 40
2 14 11 38
3 18 9 38
4 17 16 37
5 14 12 31
6 9 15 37
7 15 16 39
8 13 18 33
9 16 12 35
10 16 11 38

IF DATA MATRIX IS CORRECT TYPE: GOOD
IF DATA MATRIX IS NOT CORRECT TYPE: CHANGE

WHAT IS THE NUMBER OF THE ROW YOU WISH TO CHANGE?
6

ENTER THE CORRECT OBSERVATION FOR ROW 6
10 15 38

DO YOU WISH TO CHANGE ANOTHER ROW?
NO

YOUR DATA MATRIX IS
15 16 40
14 11 38
18 9 38
17 16 37
14 12 31
10 15 38
15 16 39
13 18 33
16 12 35
18 11 38

MEAN( 1) MEAN( 2) MEAN( 3)
15.00 13.60 36.70

THE SUMS OF SQUARES AND CROSS PRODUCTS ARE
2304 2010 5517
2010 1928 4990
5517 4990 13541

THE SAMPLE VARIANCE COVARIANCE MATRIX IS
\[
\begin{pmatrix}
6 & \tilde{3.3333} & \tilde{1.3333} \\
\tilde{3.3333} & \tilde{8.7111} & \tilde{0.1333} \\
\tilde{1.3333} & \tilde{0.1333} & \tilde{8.0111}
\end{pmatrix}
\]

THE INVERSE OF THE SAMPLE VARIANCE COVARIANCE MATRIX IS
\[
\begin{pmatrix}
0.2212 & 0.0841 & -0.0354 \\
0.0841 & 0.1468 & 0.0115 \\
-0.0354 & 0.0115 & 0.1305
\end{pmatrix}
\]

T-SQUARE E-COMPUTED
106.4 27.6
PROGRAM DESCRIPTION

\[ R^*MULTEST;C;COUNT:D;F;INDEX;INVERSE;KXX:S;SUMS:TSQ \]

1. This program requires the use of functions: DATAMAT, COLNAMES.
2. INPUT, RWTD, AND FMT.
3. The variables N, X, MEAN, and SS are global variables since they are required in both DATAMAT and MULTEST.
4. THIS PROGRAM TESTS THE HYPOTHESIS THAT A VECTOR OF MEANS IS EQUAL TO 'A VECTOR OF CONSTANTS. FOR EXAMPLE:
HO: [U(1)=100, U(2)=75, U(3)=543, ... , U(P)=25].'
6. WHAT IS THE NUMBER OF VARIABLES FOR THIS PROBLEM?
7. P+0
8. 'ENTER THE VALUES FOR YOUR NULL HYPOTHESIS BELOW - ONE VALUE ON EACH LINE.'
9. U+0
10. COUNT+1
11. LOOP REQUESTS THE USER TO INPUT THE HYPOTHETICAL VALUES FOR MU(1) TO MU(P). FMT IS REQUIRED.
12. LOOP: U+U.INPUT 'MU(T,I2, M)=\(M\)' FMT(COUNT) COUNT=COUNT+1
13. THE HYPOTHESIS YOU WISH TO TEST IS '
14. 'U(1) = \(1+U \cdot \text{NUMBER}+2\)
15. LP1: 'U('NUMBER') = \(1+(\text{NUMBER}-1)\cdot U \cdot \text{NUMBER}+\text{NUMBER}+1\)
16. 'THE SAMPLE VARIANCE COVARIANCE MATRIX IS'
17. 'S'= 'S
18. 'THE INVERSE OF THE SAMPLE VARIANCE COVARIANCE MATRIX IS'
19. INVERSE=S
20. 'TSQ=NXD+XINVERSE+(QD)
21. THE FOLLOWING LINES FORMAT THE PRINTING OF THE VALUES FOR T-SQUARE AND F
22. FS='2F12.1'
23. 'FS COLNAMES T-SQUARE/F-COMPUTED
24. FS FMT(TSQ;F)
V R=DATAMAT;FS;H;INDEX;V
[1] 'ENTER YOUR DATA WITH ONE OBSERVATION VECTOR ON EACH ROW. EACH ROW '
[2] 'SHOULD CONTAIN ';P;' VARIABLES. BELOW THE LAST ROW TYPE: STOP'
[3] STOP+35.34758697821
[4] X+(0,F)0
[6] *(STOP=1+V)/NEXT
[8] 'ERROR'
[9] *REPEAT
[10] ADD:X=X,[1](1,P)pV
[12] NEXT:N+1+(P)x
[13] 'YOUR DATA MATRIX CONTAINS ';N;' ROWS AND ';P;' COLUMNS. PLEASE CHECK'
[14] 'TO SEE THAT IT IS CORRECT.'
[15] '
[16] ((N,1)p\n),X
[17] 'IF DATA MATRIX IS CORRECT TYPE: GOOD
   IF DATA MATRIX IS NOT CORRECT TYPE: CHANGE'
[18] CHECK=* 
[19] *('GOOD'=4+CHECK)/PRODUCT
[20] CHANGE:'WHAT IS THE NUMBER OF THE ROW YOU WISH TO CHANGE?' 
[21] INDEX=* 
[22] 'ENTER THE CORRECT OBSERVATION FOR ROW ';INDEX
[23] X[INDEX,1]=0
[24] 'DO YOU WISH TO CHANGE ANOTHER ROW?'
[25] *(YES'=3+YES)/CHANGE
[26] 'YOUR DATA MATRIX IS'
[27] '
[28] X
[29] ' 
[31] SS=(QX)+.\X
[32] H=,'\MEAN(\,I2,\,ME') AFMT(\P)
[33] ' 
[34] FS=,'I2,\,AFMT(\P)
[35] FS COLNAMES H
[36] FS AFMT((1,P)pMEANS)
[37] ' 
[38] 'THE SUMS OF SQUARES AND CROSS PRODUCTS ARE'
[39] ' 
[40] SS

V R=INPUT PROM
[1] I1:PROM
[3] 'ENTER ONE VALUE ONLY'
[4] *I1

V
DESCRIPTION

**LUMPSUM** is a program designed to discount future streams of income, medical costs, retirement annuities, etc., and reduce these to a lump sum payment in liability cases. For example, a person injured in an automobile accident may have:

1. Lost income between time of injury and settlement of a liability case.
2. Incurred uninsured medical costs during the same period.
3. Been permanently disabled to some degree and therefore lose future income, promotions, and retirement benefits and incur both uninsured maintenance medical costs and periodic uninsured major medical costs.

**LUMPSUM** will handle this problem and can be used by students, lawyers and economists who wish to calculate the net present value of medical costs and lost income due to the disability of an injured person.

**LUMPSUM** is a prompted program. The inputs required are:

1. Annual income of the injured person at the time of injury.
2. Age at the time of injury.
3. Present age.
4. Risk-free current interest rate (or some agreed upon interest rate acceptable to litigants), in percentage.

5. Long-run improvement in productivity (real percentage increase in per capita income).

6. Employer contribution to retirement annuity, in percentage of income at time of injury.

7. Age of the injured at the time of future promotions that the employer could have anticipated for the injured.

8. Each promotion as a percentage of income at that age.

9. Based upon current age, the remaining expected life, taken from mortality tables.

10. Continuing uninsured maintenance medical costs as a percentage of income at time of injury due to injury. (For example, the injured may require continuing physical therapy throughout his or her life.)

11. Major future uninsured medical costs: age when they are to be incurred and the dollar amounts. (For example, physicians may suggest the need for one or two major operations to correct the more serious physical aspects of the disability. If so, they should provide current dollar estimates which are then adjusted for inflation at a rate equal to the productivity factor.)

12. Lost income between the time of injury and the time of settlement resulting from the injury.
13. Uninsured medical costs incurred between the time of injury and the settlement.

14. Interest rates for each year of lost income or uninsured medical costs incurred between the time of injury and the settlement.

15. Degree of disability in percentage. (For example, physicians may estimate that the injured has lost 25% of his or her capacity to continue earning income and pursuing a career.)

The outputs are:

1. The discounted value of future income, including promotions.

2. The discounted value of employer contributions to retirement.

3. The discounted value of uninsured medical maintenance costs, adjusted for cost escalation.

4. The value of lost income and uninsured medical costs between the time of injury and the settlement, compounded by the past annual interest rate.

5. The discounted value of uninsured major medical costs, adjusted for cost escalation.

6. The value of the award, adjusted for the degree of continuing disability.

To illustrate the use of the program, suppose you have the following case:

1. Income at time of injury $10,000  
2. Age at time of injury 30 yrs.  
3. Present age 35 yrs.
4. Current interest rate 6%
5. Rate of productivity improvement 3%
6. Employer's contribution to retirement 12% of income
7. Anticipated promotions

<table>
<thead>
<tr>
<th>Age</th>
<th>40 yrs.</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of Income</td>
<td>10%</td>
<td>8%</td>
<td>6%</td>
<td>5%</td>
<td>4%</td>
</tr>
</tbody>
</table>

8. Remaining life expectancy 35 yrs.
9. Uninsured maintenance medical costs 10% of income
10. Future major medical costs

<table>
<thead>
<tr>
<th>Age</th>
<th>40 yrs.</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dollars</td>
<td>$3000</td>
<td>$2500</td>
</tr>
</tbody>
</table>

11. Lost income uninsured medical costs, and interest rates since injury.

<table>
<thead>
<tr>
<th>Age</th>
<th>Lost Income</th>
<th>Uninsured Medical Costs</th>
<th>Interest Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 yrs.</td>
<td>$1200</td>
<td>$500</td>
<td>5%</td>
</tr>
<tr>
<td>31</td>
<td>$1300</td>
<td>$600</td>
<td>4%</td>
</tr>
<tr>
<td>32</td>
<td>$1500</td>
<td>$400</td>
<td>6%</td>
</tr>
<tr>
<td>33</td>
<td>$1200</td>
<td>$600</td>
<td>7%</td>
</tr>
<tr>
<td>34</td>
<td>$1300</td>
<td>$500</td>
<td>6%</td>
</tr>
<tr>
<td>35</td>
<td>$1200</td>
<td>$600</td>
<td>6%</td>
</tr>
</tbody>
</table>

12. Degree of disability, in percentage 25%

Using these data, the following section shows a sample run.
PROGRAM USAGE

LUMP SUM
ENTER INCOME AT TIME OF INJURY
\[ \text{\$10000} \]
ENTER AGE AT TIME OF INJURY AND PRESENT AGE
\[ 30 \ 35 \]
ENTER CURRENT INTEREST RATE IN PERCENTAGE
\[ 6 \]
ENTER PRODUCTIVITY FACTOR IN PERCENTAGE
\[ 3 \]
ENTER RETIREMENT ANNUITY IN PERCENTAGE OF CURRENT INCOME
\[ \text{\$12} \]
ENTER AGE AT PROMOTION, PERCENTAGE PROMOTION WOULD BE AT THEN INCOME
ENTER ALL PROMOTIONS AND WHEN DONE JUST HIT CARRIAGE RETURN

<table>
<thead>
<tr>
<th>AGE</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>45</td>
<td>8</td>
</tr>
<tr>
<td>50</td>
<td>6</td>
</tr>
<tr>
<td>55</td>
<td>5</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
</tr>
</tbody>
</table>

ENTRY:
ENTER BASED ON CURRENT AGE, ENTER REMAINING YEARS OF LIFE EXPECTANCY FROM MORTALITY TABLES
\[ \text{\$35} \]
ENTER BASIC UNINSURED MAINTENANCE MEDICAL COSTS AS PERCENT OF INCOME AT TIME OF INJURY
\[ \text{\$10} \]
ENTER ESTIMATED AGE AT INCURRENCE OF FUTURE MAJOR MEDICAL COSTS AND ENTER DOLLAR ANNUAL AMOUNTS OF COSTS VALUED IN PRESENT DOLLARS
WHEN ALL MAJOR MEDICAL COST OCCURRENCES ARE ENTERED JUST HIT RETURN

<table>
<thead>
<tr>
<th>AGE</th>
<th>DOLLARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>3000</td>
</tr>
<tr>
<td>45</td>
<td>2500</td>
</tr>
</tbody>
</table>

ENTRY:
ENTER FOR AGE INDICATED SINCE INJURY ENTER ANNUAL LOST INCOME, UNINSURED MEDICAL COSTS IN DOLLAR VALUES AND INTEREST RATES IN PERCENT.

<table>
<thead>
<tr>
<th>AGE</th>
<th>UNINS MED COSTS</th>
<th>INTEREST RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1200</td>
<td>500</td>
</tr>
<tr>
<td>31</td>
<td>1300</td>
<td>600</td>
</tr>
<tr>
<td>32</td>
<td>1500</td>
<td>400</td>
</tr>
<tr>
<td>33</td>
<td>1200</td>
<td>600</td>
</tr>
<tr>
<td>34</td>
<td>1300</td>
<td>500</td>
</tr>
<tr>
<td>35</td>
<td>1200</td>
<td>600</td>
</tr>
</tbody>
</table>

ENTER PERCENTAGE DEGREE OF DISABILITY
\[ 25 \]
THE DISCOUNT RATES ARE:
0.9708737864 0.9425959091 0.9151416594 0.8884870479 0.8626087844
0.9275363273 0.9005207061 0.8742919477 0.8488271337 0.8241040133
0.8674779087 0.8422115619 0.817681128 0.7938651728 0.7707428862
0.7945802951 0.7714371797 0.7489681356 0.7271535297 0.7059743007
0.7203819395 0.6993999413 0.6790290692 0.6592515235 0.6400500228
0.6465151746 0.6276846355 0.6094025587 0.5916529696 0.5744203589
0.5576896688

THE DISCOUNTED VALUE OF FUTURE INCOME IS: 238005.5728
THE DISCOUNTED VALUE OF THE RETIREMENT ANNUITY IS: 28560.66873
THE DISCOUNTED VALUE OF EXPECTED MAINTENANCE MEDICAL COSTS IS: 22487.22007
THE VALUE OF LOST INCOME AND MEDICAL COSTS SINCE INJURY PLUS INTEREST IS:
13299.63044
THE DISCOUNTED VALUE OF FUTURE MAJOR MEDICAL EXPENSES IS: 4446.06114
THE AWARD IS: 233425.7725
PROGRAM DESCRIPTION

[1] *FUNCTION REQUIRES THE USE OF PUBLIC FUNCTIONS INP AND IPI
[2] "ENTER INCOME AT TIME OF INJURY" INP 1
[4] "ENTER CURRENT INTEREST RATE IN PERCENTAGE" RATE
[5] "ENTER PRODUCTIVITY FACTOR IN PERCENTAGE" PROD
[6] "ENTER RETIREMENT ANNUITY IN PERCENTAGE OF CURRENT INCOME" RET
[7] PROM 2 0 p0
[8] "ENTER AGE AT PROMOTION, PERCENTAGE PROMOTION WOULD BE AT THEN INCOME"
[9] "ENTER ALL PROMOTIONS AND WHEN DONE JUST HIT CARRIAGE RETURN"
[10] "BASED ON CURRENT AGE, ENTER REMAINING YEARS OF LIFE EXPECTANCY FROM"
[11] "MORTALITY TABLES"
[12] "ENTER BASIC UNINSURED MAINTENANCE MEDICAL COSTS AS PERCENT OF INCOME"
[13] "ENTER ESTIMATED AGE AT INCURRENCE OF FUTURE MAJOR MEDICAL COSTS AND"
[14] "ENTER DOLLAR ANNUAL AMOUNTS OF COSTS VALUED IN PRESENT DOLLARS"
[15] "WHEN ALL MAJOR MEDICAL COST OCCURRENCES ARE ENTERED JUST HIT RETURN"
[16] "ENTER PERCENTAGE DEGREE OF DISABILITY"
[17] "THE DISCOUNT RATES ARE:
[18] "THE DISCOUNTED VALUE OF FUTURE INCOME IS:"
[19] "THE DISCOUNTED VALUE OF THE RETIREMENT ANNUITY IS:
[20] "THE DISCOUNTED VALUE OF EXPECTED MAINTENANCE MEDICAL COSTS IS:
[21] "THE VALUE OF LOST INCOME AND MEDICAL COSTS SINCE INJURY PLUS INTEREST IS:
[22] "THE DISCOUNTED VALUE OF FUTURE MAJOR MEDICAL EXPENSES IS:
[23] "THE AWARD IS:"

LUMP8UM

1. THIS FUNCTION REQUIRES THE USE OF PUBLIC FUNCTIONS INP AND IPI
2. ENTER INCOME AT TIME OF INJURY: INP 1
3. ENTER AGE AT TIME OF INJURY AND PRESENT AGE: AGE
4. ENTER CURRENT INTEREST RATE IN PERCENTAGE: RATE
5. ENTER PRODUCTIVITY FACTOR IN PERCENTAGE: PROD
6. ENTER RETIREMENT ANNUITY IN PERCENTAGE OF CURRENT INCOME: RET
7. PROM 2 0 p0
8. ENTER AGE AT PROMOTION, PERCENTAGE PROMOTION WOULD BE AT THEN INCOME
9. ENTER ALL PROMOTIONS AND WHEN DONE JUST HIT CARRIAGE RETURN
10. BASED ON CURRENT AGE, ENTER REMAINING YEARS OF LIFE EXPECTANCY FROM
11. MORTALITY TABLES
12. ENTER BASIC UNINSURED MAINTENANCE MEDICAL COSTS AS PERCENT OF INCOME
13. ENTER ESTIMATED AGE AT INCURRENCE OF FUTURE MAJOR MEDICAL COSTS AND
14. ENTER DOLLAR ANNUAL AMOUNTS OF COSTS VALUED IN PRESENT DOLLARS
15. WHEN ALL MAJOR MEDICAL COST OCCURRENCES ARE ENTERED JUST HIT RETURN
16. ENTER PERCENTAGE DEGREE OF DISABILITY
17. THE DISCOUNT RATES ARE:
18. THE DISCOUNTED VALUE OF FUTURE INCOME IS:
19. THE DISCOUNTED VALUE OF THE RETIREMENT ANNUITY IS:
20. THE DISCOUNTED VALUE OF EXPECTED MAINTENANCE MEDICAL COSTS IS:
21. THE VALUE OF LOST INCOME AND MEDICAL COSTS SINCE INJURY PLUS INTEREST IS:
22. THE DISCOUNTED VALUE OF FUTURE MAJOR MEDICAL EXPENSES IS:
23. THE AWARD IS:
3.10

LINEAR PROGRAMMING WITH MIXED CONSTRAINTS

Ronald D. Miller
Assistant Dean of Systems
College of Business and Public Service
Governors State University
Park Forest South, Illinois

DESCRIPTION

Linear programming is a mathematical technique for finding the best use of a firm's limited resources. The adjective "linear" is used to describe a relationship between two or more variables, a relationship which is directly and precisely proportional. For example, linear means that a 10 percent change in the number of productive hours used in some operation will cause a 10 percent change in output. "Programming" refers to the use of certain mathematical techniques to get the best solution to a problem involving limited resources.

The APL function which initiates these linear programming calculations is entitled SIMPLEX (after the name of the technique which is employed). A complete list of all of the functions which are required for successful operation are given.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLEX</td>
<td>This function accepts and validates user input.</td>
</tr>
<tr>
<td>SIMPLEX1</td>
<td>This function develops the LP matrix and performs the computations necessary for the solution of the LP problem.</td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SOLI</td>
<td>This function solves the matrix by the Barnes M Method.</td>
</tr>
<tr>
<td>PRINT</td>
<td>This function provides the printed output as well as the format features.</td>
</tr>
<tr>
<td>REASON</td>
<td>This function contains author information.</td>
</tr>
<tr>
<td>DECSLP</td>
<td>This function gives a brief verbal description of linear programming (similar to what is contained in this section)</td>
</tr>
<tr>
<td>SAMLP</td>
<td>This function contains a sample problem, as well as explanations and examples of data input.</td>
</tr>
</tbody>
</table>

These functions have comprehensive built-in instructions and prompts for the user, and so there is no need to explain their detailed usage here. The user merely needs to type *SIMPLEX* at the terminal and the functions, through a series of interactive questions, will request the data necessary to execute the program. Thus the user need not have any special experience with either APL or linear programming in order to use this series of functions.
PROGRAM USAGE

SIMPLEX

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING - MIXED CONSTRAINTS

DO YOU KNOW HOW TO USE THIS PROGRAM? YES

ENTER A PROJECT TITLE IF YOU WISH. DOCUMENTATION RUN

IS THIS A MAXIMIZATION OR MINIMIZATION PROBLEM? REPLY MAX OR MIN. MIN

ENTER THE NUMBER OF CONSTRAINTS, AND THE NUMBER OF VARIABLES: 4 3

ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME: 1 1 1 = 1000

ENTER NEXT CONSTRAINT: 1 0 0 ≤ 300

ENTER NEXT CONSTRAINT: 0 1 0 ≤ 150

ENTER NEXT CONSTRAINT: 0 0 1 = 200

ENTER PROFIT OR COST FUNCTION COEFFICIENTS: 5 6 7

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MINIMIZATION

DOCUMENTATION RUN
### TABLEAU 1

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>300.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>150.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>200.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>13500000.00</td>
<td>100000.00</td>
<td>200000.00</td>
<td>200000.00</td>
<td>100000.00</td>
<td>0.00</td>
<td>100000.00</td>
<td>100000.00</td>
<td>100000.00</td>
<td>100000.00</td>
</tr>
</tbody>
</table>

### TABLEAU 2

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>850.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>300.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>150.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>200.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0</td>
<td>10500900.00</td>
<td>100000.00</td>
<td>6.00</td>
<td>200000.00</td>
<td>100000.00</td>
<td>0.00</td>
<td>99994.00</td>
<td>99994.00</td>
<td>99994.00</td>
<td>99994.00</td>
</tr>
</tbody>
</table>

### TABLEAU 3

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>650.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>300.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>150.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>200.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>65002300.00</td>
<td>100000.00</td>
<td>6.00</td>
<td>7.00</td>
<td>100000.00</td>
<td>0.00</td>
<td>99995.00</td>
<td>99995.00</td>
<td>99995.00</td>
<td>99995.00</td>
</tr>
</tbody>
</table>

### TABLEAU 4

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>350.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>150.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>200.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>35003800.00</td>
<td>5.00</td>
<td>6.00</td>
<td>7.00</td>
<td>100000.00</td>
<td>-99995.00</td>
<td>99994.00</td>
<td>99994.00</td>
<td>99993.00</td>
<td>99993.00</td>
</tr>
</tbody>
</table>

### TABLEAU 5

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>350.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>300.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>200.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>5900.00</td>
<td>5.00</td>
<td>6.00</td>
<td>7.00</td>
<td>6.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### OPTIMAL SOLUTION FOR MINIMIZATION

The optimum value of Z is 5900.00

<table>
<thead>
<tr>
<th>VAR</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>350.00</td>
</tr>
<tr>
<td>1</td>
<td>300.00</td>
</tr>
<tr>
<td>2</td>
<td>500.00</td>
</tr>
<tr>
<td>3</td>
<td>200.00</td>
</tr>
</tbody>
</table>

Linear programming completed.
SIMPLEX

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING - MIXED CONSTRAINTS

DO YOU KNOW HOW TO USE THIS PROGRAM? YES

ENTER A PROJECT TITLE IF YOU WISH. DOCUMENTATION RUN

IS THIS A MAXIMIZATION OR MINIMIZATION PROBLEM? REPLY MAX OR MIN. MAX

ENTER THE NUMBER OF CONSTRAINTS, AND THE NUMBER OF VARIABLES: 2 2

ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME: 4 2 ≤ 60

ENTER NEXT CONSTRAINT: 2 4 ≤ 48

ENTER PROFIT OR COST FUNCTION COEFFICIENTS: 8 5

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MAXIMIZATION

DOCUMENTATION RUN

TABLEAU 1

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>60.00</td>
<td>4.00</td>
<td>2.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>48.00</td>
<td>2.00</td>
<td>4.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

TABLEAU 2

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.00</td>
<td>1.00</td>
<td>0.50</td>
<td>-0.25</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>18.00</td>
<td>0.00</td>
<td>3.00</td>
<td>0.50</td>
<td>1.00</td>
</tr>
<tr>
<td>0</td>
<td>120.00</td>
<td>8.00</td>
<td>4.00</td>
<td>2.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

TABLEAU 3

<table>
<thead>
<tr>
<th>VAR</th>
<th>Z</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.00</td>
<td>1.00</td>
<td>0.00</td>
<td>-0.33</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>6.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.17</td>
<td>0.33</td>
</tr>
<tr>
<td>0</td>
<td>132.00</td>
<td>8.00</td>
<td>6.00</td>
<td>1.67</td>
<td>0.67</td>
</tr>
</tbody>
</table>

OPTIMAL SOLUTION FOR MINIMIZATION

THE OPTIMUM VALUE OF Z IS 132.00

<table>
<thead>
<tr>
<th>VAR</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.00</td>
</tr>
<tr>
<td>2</td>
<td>6.00</td>
</tr>
</tbody>
</table>

LINEAR PROGRAMMING COMPLETED
PROGRAM DESCRIPTION

[1] "REASON

[2] "THE LINEAR PROGRAMMING APPLICATION THAT FOLLOWS HAS BEEN WRITTEN AS ONE IN"

[3] "A SERIES OF PROGRAMS RESULTANT FROM:

[4] "INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS"

[5] "GRADUATE SCHOOL OF MANAGEMENT"

[6] "UNIVERSITY OF CALIFORNIA AT LOS ANGELES (UCLA)"

[7] "THIS PROGRAM WAS AUTHORED BY:

[8] "RONALD D. MILLER"

[9] "ASSISTANT DEAN OF SYSTEMS"

[10] "COLLEGE OF BUSINESS AND PUBLIC SERVICE"


[12] "PARK FOREST SOUTH, ILLINOIS 60466"

V DESC[[]]V

PA: PLEASE SIGN OFF BY 1530 FOR A SYSTEM RESTART.

V DESC[[]]V

[1] "LINEAR PROGRAMMING IS A MATHEMATICAL TECHNIQUE FOR FINDING THE BEST USE

[2] OF A FIRM'S LIMITED RESOURCES. THE ADJECTIVE LINEAR IS USED TO

[3] A RELATIONSHIP BETWEEN TWO OR MORE VARIABLES, A RELATIONSHIP WHICH IS

[4] DIRECTLY AND PRECISELY PROPORTIONAL. FOR EXAMPLE, LINEAR MEANS

[5] THAT A 10 PERCENT CHANGE IN THE NUMBER OF PRODUCTIVE HOURS USED IN SOME

[6] OPERATION WILL CAUSE A 10 PERCENT CHANGE IN OUTPUT. PROGRAMMING

[7] REFERS TO THE USE OF CERTAIN MATHEMATICAL TECHNIQUES TO GET THE

[8] BEST SOLUTION TO A PROBLEM INVOLVING LIMITED RESOURCES."

[9] ""
SIMPLEX

This program requires the functions DESCPP, REASON, SAMLP, SEPERATE, and SIMPLEX1; and the public functions PGS-AKI, AYN, and IPI.

PGS-AKI

Turn carriage to position paper at start of new page, then press return.

'LINEAR PROGRAMMING - MIXED CONSTRAINTS

Author: RONALD D. MILLER
Assistant Dean of Systems
College of Business and Public Service
GOVERNORS STATE UNIVERSITY
PARK FOREST SOUTH, ILLINOIS 60466

PGS-AKI 'Do you know how to use this program?'

*(1=PGS)/L1
REASON ◊ DESCPP ◊ SAMLP
L1: TITLE=AKI

L11: 'Enter a project title if you wish.'

L111: 'Reply a(1=REPLY)~('+REPLY~1+REPLY~2+REPLY)/ERR1

L112: 'Is this a maximization or minimization problem? Reply MAX or MIN.'

L113: '('A~REPLY)~('+REPLY~1+REPLY~2+REPLY)/ERR1

L114: 'Enter the number of constraints, and the number of variables:'

L115: 'Enter a project title if you wish.'

L116: 'Enter coefficients of simplex problem, one row at a time:'

L117: 'Enter profit or cost function coefficients:'

L118: 'Separate parm

L119: 'Invalid entry, please reenter'

L120: 'The objective function is not bounded above x = ; COL:'

L121: 'The reply must be MAX or MIN. Please reenter.'

L122: 'You should have entered two integers. Please reenter.'

L123: 'You have entered an incorrect number of coefficients or you have forgotten to enter the arithmetic operator(s). Please reenter.'

L124: 'You should have entered one profit or cost function coefficient for each variable. Please reenter.'

L125: 'UNBD:

The objective function is not bounded above x = ; COL:'

v
Solve the following linear programming problem:

Minimize \( 5X + 6Y + 7Z = \text{cost} \)

Subject to

\( X + Y + Z = 1000 \)
\( X \geq 300 \)
\( Y \geq 150 \)
\( Z \geq 200 \)

Reformat to this form for computer entry:

Minimize \( 5 \ 6 \ 7 \)

Subject to \( 1 \ 1 \ 1 \leq 1000 \)
\( 1 \ 0 \ 0 \leq 300 \)
\( 0 \ 1 \ 0 \geq 150 \)
\( 0 \ 0 \ 0 \geq 200 \)

The data will be entered in the following format:

Prompt: Do you know how to use this program?
Reply: Enter \( \text{YES} \) if you wish a sample problem plus an expansion of the entry of that problem. Enter \( \text{NO} \) if you already know how to use the program. You may use \( \text{Y} \) or \( \text{N} \).

Ex.: \( \text{YES} \)

Prompt: Enter a project title if you wish.
Reply: Enter any title that you wish to have printed on the result page.

Ex.: \( \text{OP512 - OPERATIONS RESEARCH - JOHN JONES} \)

Prompt: Is this maximization or a minimization problem? Reply \( \text{MAX} \) or \( \text{MIN} \).

Ex.: \( \text{MIN} \)

Prompt: Enter the number of constraints, and the number of variables.
REPLY: YOU MUST ENTER THE NUMBER OF CONSTRAINTS THAT THE PROBLEM CONTAINS. YOU WILL ALSO NEED TO ENTER THE NUMBER OF VARIABLES (SUCH AS X Y Z). YOU MUST ENTER TWO INTEGERS WITH A SPACE BETWEEN THEM.

EX.:  4 3

PROMPT: ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME:

OR: ENTER NEXT CONSTRAINT:

REPLY: ENTER ONE ROW OF THE CONSTRAINTS AT A TIME, MAKING SURE THAT YOU ENTER THE VALUES AND THE ARITHMETIC OPERATOR IN THE REFORMAT EXAMPLE GIVEN ABOVE. ONLY (= ≥ ≤) ARE VALID OPERATORS.

EX.:  1 1 1 = 1000 (FIRST CONSTRAINT ENTRY)

PROMPT: ENTER PROFIT OR COST FUNCTION COEFFICIENTS.

REPLY: ENTER THE PROFIT OR COST VALUES OF THE OBJECTIVE FUNCTION.

EX.:  5 6 7

PLEASE RETAIN THIS PRINTOUT FOR FUTURE REFERENCE.
V SIMPLEX1

1. THIS FUNCTION REQUIRES THE FUNCTION SOLI.

2. TURN CARTRIDGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

3. M1+1+(REPLY='A')

4. *(M1=1)/L7

5. LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MAXIMIZATION

6. OPTI+"OPTIMAL SOLUTION FOR MAXIMIZATION"

7. *L7+1

8. L7: "LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MINIMIZATION"

9. OPTI+"OPTIMAL SOLUTION FOR MINIMIZATION"

10. TITLE;

11. WHICH=\-1XPHIGH-100*/MAT

12. MAT=MAT[;O]C C1\+R,C1 CO CT+0 CO MAT+C1pMAT

13. ZEROS=\-R, (+/OPF+'z')+\+OP \+ MAT-MAT,ZEROSp0

14. C1\+MAT=1C1+C1 \+ C=C+1

15. MAT[;C]+MAT[;C]O MAT[;C]+R0

16. FC+C \+ PR+1

17. VIB+R0O \+ 2d+Clp0 \+ I+1 \+ Cd+Cd,(C1-C)p0

18. L13:+(OP[I]*'z')/L8 \+ MAT[PR;FC]+1 \+ MAT[PR;FC+1]+1

19. VIB[PR]+FC+1 \+ *(2=M1)/L9

20. CJ[FC+1]+PHIGH \+ *L9+1

21. CJPC+1]+PHIGH

22. PC+PC+2 \+ *L10

23. L8:MAT[PR;FC]+1 \+ VIB[PR]+PC

24. *(OP[I]*'z')/L11+1 \+ *(M1=1)/L11

25. CJ[FC]+PHIGH \+ *L11+1


27. PC+PC+1

28. L10:PR+PR+1 \+ *(R=X+I+1)/L13 \+ C=C+1

29. VPT+VIB[R] \+ CJ+VPT \+ PRINT

30. L12:*(M1=2)/L16

31. *(2\+V/((CZ\+O)\+0))/L17

32. S0LI

33. *0

34. L117:COL+.;CZi;L/CZ \+ *L17

35. L16:*(2\+V/((CZ\+O)\+0))/L18

36. S0LI

37. *0

38. L118:COL+.;CZi;[/CZ]

39. L17:*(2\+V/((M1;COL)+0))/UNBD

40. ANDED+(ANDED+0)/\+ANDED+;ANDED+(MAT[;COL]+0)A(MAT[;C]+0)

41. REMAIN+;MAT[ANDED;C1]A;MAT[ANDED;COL]

42. ROW+;REMAIN+(;/REMAIN) \+ ROW+ANDED[ROW] \+ VIB[ROW]+COL \+ CJC[ROW]+CJ[COL]

43. MAT[ROW]A;MAT[ROW]+COL \+ MAT[ROW;COL] \+ I+1

44. L18:PIV-MAT[I;COL] \+ *(I=ROW)/L19

45. MAT[I;I]+MAT[I] \+ PIV-MAT[ROW;]

46. L19:*(R\+I+1)/L18

47. PRINT

48. *(L12

49. UNBD:

THE OBJECTIVE FUNCTION IS NOT BOUNDED ABOVE X = ';COL:'

V
PRINT

THIS FUNCTION USES THE FOLLOWING SCIENTIFIC TIME SHARING FUNCTIONS: CENTER, &FMT, AND COLNAMES

CENTER+XMAT

CNTR+CNTR+CC+CC+CC

FORM='TABLEAU ',CODE[CNTR]


COLNM+'/VAR/Z ',((C1-1)X4)+COLNM

FSTR='I4,X1,F14.2,10F10.2'

FSTR CENTER FORM

FSTR COLNAMES COLNM

FSTR &FMT(VIB;MAT[;C1];MAT[;R;C1-1])

CC+CC+CC

FSTR &FMT(0;ZJ[C1];ZJ[CC])

V
THE INTERACTIVE EFFECTS OF THE MARGINAL
PROPENSITY TO CONSUME AND THE INVESTMENT
ACCELERATOR ON THE PRODUCTION OF INCOME

John Park
Chairman, Department of Economics
and Business Administration
Frostburg State College
Frostburg, Maryland

DESCRIPTION

This is a short, simple problem designed for use by undergraduate economic and business forecasting students to get them acquainted with an APL application in economics. The program attempts to generate a trend of national income (Gross National Product or "GNP"), based upon a set of assumptions. These assumptions are described in the program. Different combinations of assumptions would, of course, produce different income trends.

The object of the program is to attack a problem which is too tedious for hand calculation and solution. The project may be assigned to students as a homework assignment or for drill and practice. Outputs are printed out in a tabular form and, as an option, the trend of income (the last column in the table) can be plotted out on a chart.

To use the program, the student types ACTION and, when asked, enters the following values.

1. The autonomous investment (constant throughout the exercise)
2. The marginal propensity to consume (MPC)
3. The investment accelerator (alpha)
4. The number of periods the program is to run.
As an additional exercise, the instructor may wish to designate to students a particular shape (perhaps fluctuating) of income trends of the past and/or present as a starting point, and then ask them to suggest a particular combination of numerical figures for marginal propensity to consume and alpha which would bring the given initial trend of income to some planned or desired level.
ACTION
DO YOU NEED INSTRUCTIONS?
YES
HI ECONOMIST! I SEE THAT YOU ARE TRYING TO FORECAST THE NATIONAL INCOME TREND. WITH THE HELP OF AFL, AND YOU MAY SUCCEED IN DOING SO IN A JIFFY. LET US ASSUME THE FOLLOWING CONDITIONS:

1. AUTONOMOUS INVESTMENT (A) IS CONSTANT THROUGHOUT THE GIVEN PERIODS WE OBSERVE.

2. INDUCED CONSUMPTION (C) OF A PERIOD IS THE NATIONAL INCOME OF THE PREVIOUS PERIOD ADJUSTED BY YOUR MARGINAL PROPENSITY TO CONSUME (MPC).


4. TOTAL NATIONAL INCOME (Y) OF A PERIOD IS THE SUMMATION OF A, C, AND I FOR THAT PERIOD.

5. THE TOTAL NUMBER OF MULTIPLIER PERIODS (T'S) WE OBSERVE IS, SAY 20 (I.E., 20 T'S).

ENTER AUTONOMOUS INVESTMENT (A)  
\[ 1000 \]
ENTER MARGINAL PROPENSITY TO CONSUME (MPC)  
\[ .75 \]
ENTER ALPHA (ACCELERATOR)  
\[ 2.7 \]
ENTER NUMBER OF PERIODS  
\[ 20 \]
<table>
<thead>
<tr>
<th>PERIOD</th>
<th>INVESTMENT</th>
<th>CONSUMPTION</th>
<th>INVESTMENT</th>
<th>NATIONAL INCOME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(T) (A)</td>
<td>(C)</td>
<td>(I)</td>
<td>(Y)</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>1,000</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>750</td>
<td>3,025</td>
<td>3,775</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>2,831</td>
<td>5,619</td>
<td>9,451</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>7,088</td>
<td>11,493</td>
<td>19,581</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>14,686</td>
<td>20,514</td>
<td>36,200</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>27,150</td>
<td>33,653</td>
<td>61,603</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>46,353</td>
<td>51,847</td>
<td>99,199</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>74,400</td>
<td>75,727</td>
<td>151,126</td>
</tr>
<tr>
<td>9</td>
<td>1000</td>
<td>113,345</td>
<td>105,152</td>
<td>219,497</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
<td>164,623</td>
<td>138,450</td>
<td>304,073</td>
</tr>
<tr>
<td>11</td>
<td>1000</td>
<td>228,055</td>
<td>171,267</td>
<td>400,321</td>
</tr>
<tr>
<td>12</td>
<td>1000</td>
<td>300,241</td>
<td>194,903</td>
<td>495,144</td>
</tr>
<tr>
<td>13</td>
<td>1000</td>
<td>372,108</td>
<td>194,041</td>
<td>567,149</td>
</tr>
<tr>
<td>14</td>
<td>1000</td>
<td>425,361</td>
<td>143,785</td>
<td>570,146</td>
</tr>
<tr>
<td>15</td>
<td>1000</td>
<td>427,609</td>
<td>6,070</td>
<td>434,679</td>
</tr>
<tr>
<td>16</td>
<td>1000</td>
<td>326,009</td>
<td>274,320</td>
<td>601,330</td>
</tr>
<tr>
<td>17</td>
<td>1000</td>
<td>450,997</td>
<td>337,467</td>
<td>788,464</td>
</tr>
<tr>
<td>18</td>
<td>1000</td>
<td>592,098</td>
<td>380,973</td>
<td>973,071</td>
</tr>
<tr>
<td>19</td>
<td>1000</td>
<td>730,553</td>
<td>373,828</td>
<td>1,105,381</td>
</tr>
<tr>
<td>20</td>
<td>1000</td>
<td>829,036</td>
<td>265,904</td>
<td>1,095,940</td>
</tr>
</tbody>
</table>

Would you like a plot of the above data? Yes
HI ECONOMIST! I SEE THAT YOU ARE TRYING TO FORECAST THE NATIONAL INCOME TREND. WITH THE HELP OF APL, AND YOU MAY SUCCEED IN DOING SO IN A JIFFY. LET US ASSUME THE FOLLOWING CONDITIONS:

1. AUTONOMOUS INVESTMENT (A) IS CONSTANT THROUGHOUT THE GIVEN PERIODS WE OBSERVE.
2. INDUCED CONSUMPTION (C) OF A PERIOD IS THE NATIONAL INCOME OF THE PREVIOUS PERIOD ADJUSTED BY YOUR MARGINAL PROPENSITY TO CONSUME (MP C).
4. TOTAL NATIONAL INCOME (Y) OF A PERIOD IS THE SUMMATION OF A, C, AND I FOR THAT PERIOD.
V ACTION;A;ALPHA;C;CTR;FMT;I;MPC;N;T;Y
[1] * THIS FUNCTION REQUIRES THE FUNCTION DESCRIBE ACTION, WHICH DESCRIBES
[3] * FORMAT AND PLOT
[4] 'DO YOU NEED INSTRUCTIONS ?'
[5] +1*N8'=1+[]
[6] DESCRIBE ACTION
[7] 'ENTER AUTONOMOUS INVESTMENT (A)'
[8] Y+.,A+[]
[9] I=O+D
[10] 'ENTER MARGINAL PROPENSITY TO CONSUME (MPC)'
[12] 'ENTER ALPHA (ACCELERATOR) '
[13] ALPHA+[]
[14] 'ENTER NUMBER OF PERIODS'
[15] N+[]
[16] CTR+2
[17] LOOP:C+C,MPCXY[CTR-1]
[18] I=I+ALPHA1X[CCTR]-C[CTR-1]
[19] Y=Y+I+ACHR)+I[CCTR]
[20] CTR=CTR+1
[21] LOOP:(CTRSl=1)
[22] A+NDA
[23] T+1
[24] ' '
[25] ' '
[26] FMT="2112.30117"'
[27] FMT COLNAMES '/MULTIPLIER/AUTONOMOUS/INDUCED /INDUCED /TOTAL NATIONA
[28] L'
[29] ' '
[30] FMT COLNAMES '/PERIOD /INVESTMENT/CONSUMPTION/INVESTMENT/INCOME
[31] ' '
[32] FMT AFRMT(T;A;C;I;Y)
[33] ' '
[34] ' '
[35] 'WOULD YOU LIKE A PLOT OF THE ABOVE DATA?'
[36] +0*N8'=1+[]
[37] 60 80 PLOT Y
[38] ' '
[39] ' '
V
DESCRIPTION

This program is used to analyze Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) networks. Such networks are useful in planning, scheduling, and control of complex projects (such as research, development, and construction).

The program can be used to solve problems which would be too time consuming for hand calculation or to verify the results of manual calculations.

It was developed to be used in a junior year course in quantitative methods in business or by beginning graduate students in a prerequisite course in quantitative methods for an MBA program.

Instructions for Use

Input instructions are contained within the program and are printed out to the user as needed.

It is assumed that the user has a basic understanding of network theory as presented in a basic text such as Quantitative Approaches to Management by Levin and Kirkpatrick.

The user desiring a description of the workspace should execute the niladic function NETINFO. The user then has the option of analyzing a network after having read the workspace description. Those not desiring a description should execute the niladic function NETI directly in order to initiate a network.
analysis. The input variables are:

<table>
<thead>
<tr>
<th>PERT Networks</th>
<th>CPM Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginning Event</td>
<td>Beginning Event</td>
</tr>
<tr>
<td>Ending Event</td>
<td>Ending Event</td>
</tr>
<tr>
<td>Optimistic Time</td>
<td>Expected Time</td>
</tr>
<tr>
<td>Most Likely Time</td>
<td>Latest Allowable Date</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Latest Allowable Date</td>
<td></td>
</tr>
<tr>
<td>Completion Dates for</td>
<td></td>
</tr>
<tr>
<td>Probability Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Outputs from the program include:

1. List of input data
2. Number of activities input
3. Events listed in numerical order
4. Starting and finish events listed
5. Critical path
6. Expected completion date
7. Slack
8. Z values for probability analysis (optional) -- (PERT only)
9. Complete event analysis (optional)
   For each event:
   Event numbers
   Expected completion date
   Latest date
   Slack
   Critical path indicator
(10) Complete activity analysis (optional)
For each activity:
To-From events
Expected time
Expected start date
Expected completion date
Latest start date
Latest completion date
Standard deviation (PERT only)
Critical path indicator

Sample Use

In the next section the following uses are demonstrated:

(1) PERT Network
(2) CPM Network
(3) Error checking and recovery

![Figure 1](image)

Figure 1

The network shown in Figure 1 is used for these demonstrations. These data were used to illustrate that the critical path can change as the result of consideration of pessimistic or optimistic times, as is true with PERT analysis, while CPM considers only one time estimate.
PROGRAM USAGE

NETINFO

WORKSPACE NETWORK

This workspace contains three functions which operate together to solve problems in network analysis. It is assumed that the user is familiar with network theory.

The user may specify CPM or PERT analysis. Provision is made for the user to select abbreviated or complete output results.

Features include:
(1) Self-contained input instructions
(2) Extensive error checking and recovery
(3) Capability of correcting data in network after it has been input
(4) Optional computation of 'z' values for probability analysis with PERT networks

The functions in this workspace are described below:

NETINFO - contains a description of the functions. The user has the option of executing the analysis of a PERT or CPM network from this function by calling function NET1.

NET1 - handles input of network data to be analyzed, error checking, recovery, and network analysis. Calls function NET2 for output.

NET2 - produces output of the PERT or CPM network analysis. The user may specify short or long output format.

Do you want to input a network to be analyzed? Answer YES or NO. YES

Enter CPM or PERT to specify type of network to be analyzed
NET1
ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED
XYZ
ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED
PERT
ENTER PERT ACTIVITY DATA IN GROUPS OF FIVE: BEGINNING EVENT, ENDING
EVENT, OPTIMISTIC TIME, MOST LIKELY TIME, PESSIMISTIC TIME.
BEGINNING EVENT MUST BE A HIGHER NUMBER THAN ENDING EVENT FOR EACH ACTIVITY.
ENTER -1 TO INDICATE THAT YOU HAVE NO MORE DATA.
D:
1 2 6 7 8
D:
2 3 8 9
INPUT MUST BE 5 ITEMS.
D:
2 3 8 9 10
D:
3 4 4 5 6
D:
3 5 7 8 9
D:
-1

INPUT DATA
1 2 6 7 8
2 3 8 9 10
3 4 4 5 6
3 5 7 8 9

DO YOU WISH TO CHANGE ANY DATA? ANSWER YES OR NO.
YES
ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.
D:
1
INPUT MUST BE 3 ITEMS.
ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.
D:
10 10 8
ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.
D:
1 5 10

REVISED DATA:
1 2 6 7 10
2 3 8 9 10
3 4 4 5 6
3 5 7 8 9

DO YOU WISH TO CHANGE ANY DATA? ANSWER YES OR NO.
NO
PROGRAM DESCRIPTION

This function provides the user with a description of the contents of the workspace. The user may call the function which performs the network analysis after reading the description of the workspace by giving an affirmative response to the question, "Do you want to input a network to be analyzed?"

This workspace contains three functions which operate together to solve problems in network analysis. It is assumed that the user is familiar with network theory.

The user may specify CPM or PERT analysis. Provision is made for the user to select abbreviated or complete output results.

Features include:

1. Self-contained input instructions
2. Extensive error checking and recovery
3. Capability of correcting data in network after it has been input
4. Optional computation of 'Z' values for probability analysis with PERT networks

The functions in this workspace are described below:

- **NETINFO** - contains a description of the functions. The user has the option of executing the analysis of a PERT or CPM network from this function by calling function NET1.
- **NET1** - handles input of network data to be analyzed, error checking, recovery, and network analysis. Calls function NET2 for output.
- **NET2** - produces output of the PERT or CPM network analysis. The user may specify short or long output format.

Question: "Do you want to input a network to be analyzed? Answer yes or no."

Call function NET1 if the user wishes to perform a network analysis.
This function receives CPM or PERT network data from the user and analyzes the network to determine the expected times, expected start and completion dates, latest start and completion dates, and the critical path. In addition, activity standard deviations, network standard deviation, and Z values for selected completion dates are computed for PERT networks only. It is assumed that the user is familiar with network theory. The function should be used to analyze problems which are too complex for manual calculation or to check the results of manual calculations for classroom use.

Input instructions are contained within the program. Start: 'ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED'

- CPM: 'Z=3'
- Input data instructions for CPM networks
  'ENTER CPM ACTIVITY DATA IN GROUPS OF THREE:'
  'BEGINNING EVENT, ENDING EVENT, EXPECTED TIME.'
- CPM: +G01
- PERT: 'Z=5'
- Input data instructions for PERT networks
  'ENTER PERT ACTIVITY DATA IN GROUPS OF FIVE: BEGINNING EVENT, ENDING EVENT, OPTIMISTIC TIME, MOST LIKELY TIME, PESSIMISTIC TIME.'
- G01: 'BEGINNING EVENT MUST BE A HIGHER NUMBER THAN ENDING EVENT FOR EACH ACTIVITY.'
- 'ENTER -1 TO INDICATE THAT YOU HAVE NO MORE DATA.'
- NA+=0
- ACT+=0
- Network data read in here
- READIN: A=0
- +READIN
- (NEXT,NOMOREDATA)[1+(1+A)<1]
- Check that the appropriate number of data items is entered (3 for CPM, 5 for PERT)
- ERRI: 'INPUT MUST BE ';Z;' ITEMS.'
- +READIN
- ERR2: 'EVENTS MUST BE IN ASCENDING ORDER.'
- +READIN
- OK2: NA+=NA+1
- ACT=ACT+A
- +READIN
- NOMOREDATA: ACT+(NA,Z) pACT
- '+'
- 'Input data is displayed to user for correctness
- 'Input data'
- 'Again: '
- 'Do you wish to change any data? Answer yes or no.'
- +((YES,NO,AGAIN)[YN]=1,0]
YES: 'Enter row number, column number, and new value.'

* CHECK THAT THREE ITEMS HAVE BEEN ENTERED FOR DATA CORRECTION

* (ROW NO., COL. NO., NEW VALUE)

+GOOD1,BAD1+[1+(0A)*3]

BAD1: 'Input must be 3 items.'

+YES


DATA CORRECTIONS MADE HERE


REVISITED DATA:

DATA CORRECTIONS MADE HERE

ACT

BRANCH BACK TO SEE IF THERE ARE MORE CORRECTIONS TO BE ENTERED

AGAIN

NO: NA=1+PACT

ENTER LATEST ALLOWABLE DATE FOR COMPLETION OF PROJECT:

NA;' ACTIVITIES HAVE BEEN ENTERED.'

B1=ACT[;1],ACT[;2]

B1=B1[AB1]

B2+1+((B1,0)-(0,B1))

B2+(0B2)/B1

NE+pB2

INVOLVING 'nE;' DIFFERENT EVENTS, AS FOLLOWS:

B2

CHECK FOR MULTIPLE FINISH EVENTS

D1+1+1B2*ACT[;1]

+(D/D1)=1/OK3

MULTIPLE FINISH EVENTS IN THE NETWORK, USE DUMMY ACTIVITY:

IF MULTIPLE FINISH EVENTS, EXIT PROGRAM

0

OK3:D1+1B2*ACT[;2]

CHECK FOR MULTIPLE START EVENTS

+(D/D1)=1/OK4

MULTIPLE STARTING EVENTS IN THE NETWORK, USE DUMMY ACTIVITY:

IF MULTIPLE START EVENTS, EXIT PROGRAM

0

OK4:

THE STARTING EVENT FOR THE NETWORK IS 'n1B2;' AND THE FINISH EVENT IS 'n1B2

SET UP ARRAYS FOR NETWORK ANALYSIS. E FOR EVENTS AND C FOR ACTIVITIES.

E+(NE,5)0

C+(NA,7)0

+(Z=3)/OK5

CALCULATE EXPECTED TIMES AND SD'S FOR PERT NETWORKS
[103] C[;1]+(ACT[;3]+(4×ACT[;4])×ACT[;5])×6
[104] C[;6]+(ACT[;5]-ACT[;3])×6
[105] +GO2
[106] GO5:C[;1]+ACT[;3]
[107] PUT EVENT NUMBERS IN MATRIX E
[109] E([1;2]+E)
[110] E([1;5]+)
[111] N IS ACTIVITY INDEX
[112] GO2:N+1
[113] REFERENCE NUMBER IN E FOR BEGINNING EVENT OF ACTIVITY
[114] GO3:F1×B2×ACT[N;1]
[115] +(~(E[F1;5]=1)×(C[N;7]=0))/GO4
[116] C[N;2]+E[F1;2]
[118] C[N;7]=1
[119] GO4:×(N=NA)/GO5
[120] N=N+1
[121] +GO3
[122] M IS EVENT INDEX
[123] GO5:M+1
[124] GO6:×(E[N;5]=1)/GO7
[125] F2 IS LIST OF ACTIVITIES WHOSE ENDING EVENT IS EVENT BEING ANALYZED
[126] F2+.×(ACT[;2]=E[M;1])/NA
[127] IF FLAGS FOR ALL SUCH ACTIVITIES ARE SET, UPDATE EVENT.
[128] FLAGS+.C[F2;1]
[129] +(~(∀FLAGS))/GO7
[130] EXPECTED DATE FOR EVENT IS LATEST COMPLETION DATE OF ALL ACTIVITIES
[131] LEADING TO THE EVENT
[132] E[M;2]+[C[F2;1]
[133] E[M;5]=1
[134] GO7:+(M=NE)/GO8
[135] M=M+1
[136] +GO6
[137] IF ALL EVENT FLAGS ARE SET, ALL EXPECTED DATES ARE RESOLVED AND
[138] FORWARD PART OF ANALYSIS IS COMPLETE
[139] GO8:+(~(∀E[;5]))/GO02
[140] FORWARD ANALYSIS COMPLETED, REVERSE NEXT
[141] ZERO OUT FLAGS FOR RE-USE
[142] E[;5]=0
[143] C[;7]=0
[144] E[NE;3]+LATEST
[145] E[NE;5]=1
[146] N IS ACTIVITY INDEX
[147] GO20:N+1
[148] F1 IS REF. NO. IN E FOR ENDING EVENT OF ACTIVITY
[149] GO21:F1×B2×ACT[N;2]
[150] +(~(E[F1;5]=1)×(C[N;7]=0))/GO22
[151] C[N;5]+E[F1;2]
[152] C[N;4]+C[N;5]-C[N;1]
[153] C[N;7]=1
[154] GO22:×(N=NA)/GO23
[155] N=N+1
[156] +GO21
[157] M IS EVENT INDEX
[158] GO23:M+1
F2 is a list of activities whose beginning event is event being analyzed.

If flags for all such activities are set, update event flags.

Latest date for event is earliest start date of all activities coming from the event.

If all event flags are set, all latest dates are resolved and network analysis is nearly complete.

Compute slack.

Determine critical path.

Call Net2 for output report.

NET2

This function produces output of CPM or PERT network analysis. It accepts values from function Net1 which calls this function as its last statement. This function should not be used alone. Network analysis should be initiated by executing NetInfo or Net1. The former provides a description of the workspace prior to giving the user the option of performing network analysis. This function references AFMT which may not be available on some systems.

Results of the network analysis are contained in vector CRIT which is the critical path events and in arrays E and C. Array E contains event information. One row is assigned to each event.

Columns in array E are defined as follows:

1. Event number
2. Expected date
3. Latest date
4. Slack
5. Status flag

Array C contains activity information. One row is assigned to each activity. Columns in array C are defined as follows:

1. Expected time
2. Expected start date
3. Expected completion date
4. Latest start date
5. Latest completion date
6. Standard deviation (PERT only)
7. Status flag

Vectors 1 = on critical path 0 = not on critical path

CPAC is activity vector CPFV is event vector

CPAC = (ACT[;1] € CRIT) & (ACT[;2] € CRIT)

CPFV = E[;1] € CRIT

'CRITICAL PATH:'

CRIT = (1, pCRIT) & CRIT

'PD+MIS' AFMT(CRIT)
EXPECTED COMPLETION DATE FOR PROJECT: F7.2 \( \text{AFMT(E[NE;2])} \)

SLACK: F7.2 \( \text{AFMT(E[NE;4])} \)

CPM NETWORKS BRANCH TO LABEL GO30

\( +(Z=3) \)/GO30

PROBABILITY ANALYSIS FOR PERT NETWORKS ONLY

Q031: 'DO YOU WANT TO INPUT COMPLETION DATE(S) FOR COMPUTATION OF Z VALUES

FOR THE NORMAL PROBABILITY DISTRIBUTION? ANSWER YES OR NO.'

\( +(G031,GO30,Q031)'["YN",1+2]," \)

Q032: 'COMPUTE STANDARD DEVIATION FOR THE NETWORK

\( +(STDEV=0) \)/GO30

INPUT COMPLETION DATE(S):'

A+, 0

' NETWORK STANDARD DEVIATION:

STDEV=(+/C[;6]*2)*CPAC*0.5

Q033: '

' COMPUTE Z VALUES

\( +(STDEV=0) \)/GO30

' Z VALUES: '

\( +(STDEV=Q \) /G03

INPUT COMPLETION DATE(S):'

A+, 0

COMPUTE Z VALUES

\( +(STDEV=0) \)/GO30

Z+\( (A=E[NE;2]) \)*STDEV

Q034: '

' NETWORK STANDARD DEVIATION:

\( +(STDEV=Q \) /G03

' COMPUTE Z VALUES

\( +(STDEV=0) \)/GO30

Q035: 'FOR COMPLETE EVENT AND ACTIVITY ANALYSIS, TYPE MORE. '

TO EXIT PROGRAM TYPE STOP.'

\( +(G032,0,Q032)'["MS",1+2]," \)

Q036: ' EVENT ANALYSIS'

EVENT EXPECTED LATEST ON CRITICAL'

NUMBER COMPLETION DATE DATE SLACK PATH'

H AND J ARE VECTORS CONTAINING EVENT NUMBERS, H[N] + J[N] REPRESENTS AN ACTIVITY

\( H=\text{ACT}[;1] \)

\( J=\text{ACT}[;2] \)

ARRAY E WITH EVENT FLAG COLUMN REMOVED

\( DATA=(E[NE;4]+E \)

\( G=\text{NA} \)

ARRAY G IS * FOR 'ON CRITICAL PATH' AND BLANK FOR 'NOT ON CRITICAL PATH' USED IN EVENT ANALYSIS

\( N=1 \)

\( G034:=(CPEV[N]=0) \)/GO35

\( G[NE]=* \)

\( G035:=(N=NE) \)/GO36

\( N=N+1 \)

\( G=\text{NA} \)

\( X4,13,X12,F6.2,X9,F6.2,X6,F6.2,X11,41 \) \( \text{AFMT(DA"A;1;G)}`

G=\text{NA} \)
Array G is * for 'on critical path' and blank for 'not on critical path' used in activity analysis.

\[ 86 \] N=1
\[ 87 \] G038: N=0 \rightarrow G039
\[ 88 \] N=1 \rightarrow G040
\[ 89 \] G040: ;
\[ 90 \] G040: G[N]
\[ 91 \] CPM networks branch to G050
\[ 92 \] (Z=3) \rightarrow G050
\[ 93 \] PRODUCE PERT ACTIVITY REPORT
\[ 94 \] PERT ACTIVITY ANALYSIS'
\[ 95 \] PRODUCE CPM ACTIVITY REPORT
\[ 96 \] ARRAY C with activity flag column removed.
\[ 97 \] DATA1+(NA,6)+C
\[ 98 \] 'activity time start completion deviation path'
\[ 99 \] 'expected expected latest on critical'
\[ 100 \] 'expected expected latest path'
\[ 101 \] 'expected expected latest path'
\[ 102 \] 'expected expected latest path'
\[ 103 \] 'expected expected latest path'
\[ 104 \] 'expected expected latest path'
\[ 105 \] 'expected expected latest path'
\[ 106 \] 'expected expected latest path'
\[ 107 \] AfMt(H;J;DATA1;G)
INTERACTIVE OPTIMIZATION OF A PRODUCTION SYSTEM

Thomas J. Schriber
Professor of Management Science
Graduate School of Business Administration
The University of Michigan
Ann Arbor, Michigan

DESCRIPTION

The APL model presented here uses Monte Carlo methods to simulate the behavior of a simplified production system. The system involves two independent random variables, one dependent random variable, and two decision variables. Given particular cost information, the problem is to find decision-variable values which minimize the average daily cost of the production process. This implies the need to experimentally investigate the nature of a cost surface, searching across the surface in an attempt to find its low point. In general, such a search can be conducted according to one or more formalized searching strategies, or it can be based solely on informal strategies which the investigator devises intuitively. No searching strategy is built directly into the model given here; it consequently falls to the user to supply either a formal or an informal strategy for searching the cost surface efficiently and successfully. The model supports the search by letting the user interactively estimate the cost consequences of various alternative values (s)he chooses for the decision variables.

Assignments involving use of this model, or models similar to it, can be appropriate for a variety of audiences and in a variety of settings. Some of the possibilities are suggested below.

- The model can be used (as one in a series of computer models) in a "computer applications
in business" course to illustrate one way the computer can be employed to support decision making in the firm.

- As a more specific alternative to the above, the model can be brought into play in a course in production and operations management to suggest one mode of computer use in a production context.

- The model can be used in an introductory course in a discrete-event simulation as an example of the type of modeling in which the students themselves will engage.

- It can be used in a second-level course in computer-implemented algorithms to motivate the student of methods for seeking optima.

- The model may be of interest in courses dealing with probability and statistics. The production system on which the model is based has the flavor of a practical problem involving a combination of random variables whose complete analytic manipulation is difficult or impossible; in this sense, the underlying model demonstrates an alternative method for determining the properties of dependent random variables. In addition, for any given choice of decision-variable values, the model can be used to estimate the sampling distribution of a statistic of interest (e.g., average daily cost). Finally, the model can be used to explore the behavior of the distribution of the average daily cost estimator as a function of sample size (that is, as a function of the duration of each simulation).
Use of the model does not require previous computer experience. The user should understand the fundamental concepts of "relative frequency," but these concepts can be motivated intuitively for the purpose at hand; as a result, no course in probability and/or statistics is prerequisite to use of the model. Within these guidelines, and subject to whatever other requirements may be implied in the above list of suggested applications, the model could be used by students at any level in the curriculum.

It might be noted here that the model is designed to protect itself against "bad data" which the user might accidentally or intentionally feed to it. That is, the model examines all information inputted to it by the user, and determines whether the information is feasible in the context of the problem. If infeasible information is detected, the model asks the user to re-enter the information in question. For example, the model will not accept negative values for relative frequencies, it requires that relative frequencies sum to one hundred percent, and so on. A later section in this paper displays a sample session in which a hypothetical user deliberately tries to get the model into trouble, and shows how the model responds to such a user.

Provided below, under section headings entitled Instructions for Use and Sample Use, is a "user's writeup" which includes the following:

- A description of the production system
- An indication of the problem to be solved
- A description of a formalized searching strategy
- An illustration of model use

In spirit, the user's writeup is designed to be a self-contained problem statement from which copies can be made for distribution to students. Of course, instructors who decide to use the
problem might want to modify the user's writeup to suit their own purposes. For example, the writeup presented here describes the univariate searching strategy as one which might be employed to explore the cost surface in methodical fashion. Those who assign the problem might prefer to let their students develop their own heuristics spontaneously as a user-conducted search proceeds; or, it might be thought preferable to have the students follow some formalized searching strategy other than that of univariate search.

This problem was originally composed by the author to meet several differing needs for such a problem-model combination in the business curriculum at The University of Michigan. The system described here has been modeled in FORTRAN\(^1\) and in GPSS\(^2\). Variations on the problem described here have been presented in SIMULETTER, the quarterly publication of ACM's Special Interest Group for Simulation.\(^3\)

**Instructions for Use**

Statement of the Problem

In a certain factory, a goal of management is to operate 50 machines of a particular kind 8 hours a day, 5 days a week. Management owns 50 machines of the kind in question, and employs 50 machine operators (one per machine), so that this operating objective can be met, at least in theory. In practice, however, a problem arises in that each of the machines is subject to random failure. Whenever a machine fails, it must of course be taken out of operation. It is then routed to an in-house repair shop,

---

3 See SIMULETTER, Vol. 5, No. 3, April 1974, pp. 75-76.
where it is eventually repaired. Meanwhile, unless backup machines are available to supplement the 50 owned machines, management's operating objective is not being met.

Management estimates that the profit opportunity which is foregone whenever fewer than 50 machines are in operation at all times during the 40-hour work week is $22.50 per hour per machine. For example, if only 47 machines are being operated during a given 2-hour period, then the associated profit opportunity which is foregone is $135 ($22.50 times 3 times 2). For various reasons, any under-capacity operation which might come about cannot be made up by working overtime. Management does have the option, however, to supplement the 50 machines it owns by leasing (renting) machines of like kind on a long-term basis at a cost of $30 per day per machine. (The leasing charges are independent of the extent to which the leased machines are actually used.) Management also can decide how many repairmen to employ for the purpose of repairing failed machines. For example, it might be decided to hire 2 repairmen, and to lease 9 machines. The Figure 1 diagram then shows what the overall state of affairs might be in the system at some arbitrary time. Each square in the diagram represents a machine. Squares without an X in them denote machines which are either being operated, or are in backup status, ready for use when needed. Squares with an X in them signify machines which either are being repaired, or are queued up waiting to go into repair. Of the 59 machines in this hypothesized situation, then, 2 are in repair; 2 are queued for repair; 5 are in backup status; and the other 59 are being operated.

Assignment

Your task is to determine for management that combination of repairmen-hired and machines-leased which will minimize the average daily cost of the production process. To help you in
task, a computer model has been built to simulate the operation of the factory. You are to work with the model to conduct a search for that hire-lease combination which minimizes the average daily cost of operating the factory. Pertinent lifetime and repairtime data for the machines in question are given below, and various modeling assumptions are stated. A strategy is also described for methodically searching for the minimum-cost hire-lease combination. Finally, the history of a typical session involving use of the available computer model is shown.

After scanning the sections on data and modeling assumptions, you should do the following:

- Study the section on searching strategy
- Review the specifics of the sample session
- Go to a computer terminal and use the model to perform the requested optimization.

- Write up and hand in a brief report to management, summarizing the results of your investigation and providing your final recommendations.

**Data**

Based on past records, the lifetime of a machine which has just been repaired is distributed as follows:

<table>
<thead>
<tr>
<th>Lifetime, Hours</th>
<th>110</th>
<th>120</th>
<th>130</th>
<th>140</th>
<th>150</th>
<th>160</th>
<th>170</th>
<th>180</th>
<th>190</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Frequency, %</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>12</td>
<td>21</td>
<td>17</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

Records also indicate that the time required to repair a failed machine (not counting any time spent waiting to go into repair) follows the distribution given below.

<table>
<thead>
<tr>
<th>Repairtime, Hours</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Frequency, %</td>
<td>7</td>
<td>10</td>
<td>16</td>
<td>20</td>
<td>15</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>

**Modeling Assumptions**

The computer model which simulates operation of the factory is based on the following assumptions.

- The time required to replace a failed machine with a backup machine is negligibly small.

- Owned machines and leased machines are virtually identical to one another. They are characterized by the same lifetime and repairtime distributions. Like owned machines, when leased machines fail they are repaired in-house.

- The lifetime and repairtime distributions are independent of a machine's operating history.
- Machines are taken out of operation only when they fail.

- Repairmen always work on an individual basis, one repairman per machine being repaired. (Idle repairmen cannot team up with busy repairmen to speed up the process of completing a machine's repair.)

- Each simulation performed with the model starts with the factory at typical "steady state" conditions. For example, at startup time those machines being operated already have used up some randomly-chosen fraction of their current lifetime. (The random fraction is determined separately, machine by machine.) This means that transient phenomena do not have to be taken into account in interpreting statistics printed out by the model at the end of each simulation.

**Strategy for Use of the Model**

The model is to be used to estimate the hire-lease combination which minimizes the average daily cost of running the factory. Some thought should be given then, to the way the cost is expected to behave as various hire-lease combinations are considered. For a fixed number of repairmen, average daily cost will vary with "machines rented" as shown in Figure 2. With too few rented machines, costs are high because of the foregone profit opportunities resulting from under-capacity operation. With too many rented machines, costs are also high because the leasing fee is being paid to provide more supplemental machines than are necessary. Somewhere between these two extremes, the average
cost can be expected to pass through a minimum, as is suggested in Figure 2.

Similarly, for a given number of rented machines, the influence of "repairmen hired" will follow the pattern in Figure 3. With too few repairmen, high average costs are experienced because the cost component associated with under-capacity operation is large. With too many repairmen, costs are high because the wage component is larger than it need be. It is reasonable to expect that average daily cost goes through a minimum somewhere between these two extremes, as per Figure 3.

A three-dimensional perspective of the situation is suggested in Figure 4. Note that the mesh points in the grid laid down in the Figure 4 hired-rented plane represent feasible (that is, non-negative and integer-valued) choices of values for the decision-variable pair. Located above each of these mesh points, in the third dimension, is the expected value of the daily cost corresponding to that particular hire-rent alternative. (These cost points are not shown in Figure 4.) The collection of all such cost points forms a cost surface in the third
dimension. (The term "surface," although suggestive, is a bit misleading here. The points do not actually form a surface; instead, they simply form a collection of discrete values, with no continuum joining them.) Figures 2 and 3 are then simply contours through this cost surface.

It can be argued that the cost surface in this problem is like that of an irregularly shaped bowl. More specifically, it can be argued that the cost is a unimodal function of the two decision variables involved. A function is unimodal if the independent variables are altered in an (arbitrarily) given direction, and if the resulting values of the function are found initially to decrease down to some minimum, and then to rise again. In such functions, as the name suggests, there is only a single low point. Note the consistency between the concept of unimodality, and the shape of the Figure 2 and Figure 3 contours through the
cost surface. Given that the cost surface is unimodal, the search for the optimal hire-rent combination, then, is essentially a search for the only "lowest point" on the cost surface. This unimodality can be exploited by employing a univariate searching strategy to seek the conditions under which the cost function assumes its minimum value. Briefly, the univariate strategy involves moving through the search domain along successive contours until all more promising directions of movement have been exhausted. In the context of the problem at hand, the specifics of a univariate search are outlined below.

(1) Perform a simulation with the model to estimate the average daily cost associated with some first choice of a hire-rent combination. (In this problem, preliminary hand computations can be carried out to estimate a reasonable hire-rent combination with which to begin.)

(2) Holding the "number hired" constant, vary the "number rented" and perform corresponding simulations until the low point in the corresponding cost contour has been found. This corresponds to finding the low point in Figure 2.

(3) Holding the "number rented" constant at its low-point value from (2), vary the "number hired" and perform corresponding simulations until the low point in that cost contour has been found.

This is equivalent to finding the low point in Figure 3.

(4) Is the final hire-rent pair in (3) the same as that in (2)? If not, return to (2). Otherwise, go to (5).

(5) Call the best hire-rent combination found so far the tentative optimum. As a final check, simulate for the as-yet-uninvestigated hire-rent combinations (if any) surrounding this tentatively optimal combination.

(6) Is the lowest-cost point in (5) that of the tentative optimum? If so, the search is finished. If not, let the lowest-cost point be the new tentative optimum, and return to (5).

In carrying out the steps just described, it is helpful to enter the various cost points in a grid as the search proceeds. Figure 5 provides a grid which can be used for this purpose.

It should now be clear where the univariate searching strategy gets its name. Only one decision variable is permitted to vary at a time, while the other is held constant (or, more generally, while the others are held constant). Of course, the univariate searching strategy involves a well defined, step-by-step procedure, meaning it can be programmed and need not necessarily be carried out interactively by a person sitting at a computer terminal.

(For a FORTRAN subroutine which implements the generalized univariate searching strategy, see "Use of An External Optimizing Algorithm with a GPSS Model," by Robert M. Lefkowits and Thomas J. Schriber, Proceedings of the Fifth Conference on Applications of Simulation, 1971, pp. 1-6.)
Figure 5
Note that in a simulation context, a potential problem can arise with any searching strategy because each simulation only provides an estimate of the expected daily cost. If the variability of the estimator is high (because, for example, each simulation is of short duration), it is possible to draw a false conclusion about where the low cost point is located. (Strictly speaking, before the computer model is employed for optimization purposes, it should be used to study the behavior of the cost estimator as a function of the duration of simulation. Statistical methods should then be used either to determine point estimates of the expected cost values, or to construct interval estimates of the expected cost values at whatever is judged to be an acceptable confidence level. For simplicity here, it is simply recommended that each simulation be performed for 200 working days, and that the resulting average daily cost be taken as a point estimate of the expected cost value.)

Sample Use

Use of the computer model is illustrated in the sample sessions displayed in the following section. Lifetime and repair-time data used in these sessions are identical to those given earlier in the statement of the problem. This is also true of the cost data. The recommended simulation span of 200 days has been used for each of the two simulations performed in the sample session.

The first sample session should be largely self-explanatory. The model requests information from the user by issuing questions such as HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME? and commands such as ENTER THE VARIOUS LIFETIME VALUES. The user responds by typing in numbers on some occasions and by typing in words on other occasions. The dialogue between the model and the
user is easily followed simply by reading through this sample session. With the sample session in hand, even a first-time computer user can sit at a terminal and engage in a similar dialogue with the model almost without effort.

The second sample session is designed to demonstrate how the model analyzes the user inputs for feasibility. As in the first session, this second session should be self-explanatory. As this session suggests, every attempt has been made to design the model to trap infeasible data inputs, notify the user when infeasible data are encountered, and give the user an immediate opportunity to type in acceptable information.

Comments on the Program Listing

Following the sample sessions is a complete listing of the model. Except for using the public functions \textit{IPI}, \textit{INP}, and \textit{AFMT}, the model is self-contained. (Code for the input functions \textit{IPI} and \textit{INP} is provided in an appendix to this monograph. \textit{AFMT} is a proprietary function of the Scientific Time Sharing Corporation, and is available in their APL/Plus System. The user who does not have access to APL/Plus can substitute his own logic in place of \textit{AFMT}. In this model, that would not be unduly difficult to do.)

The model consists of 15 functions (modules). The first of these functions, \textit{MASTER\&CONTROL}, acts as a logical main program. \textit{MASTER\&CONTROL} systematically routes control to each of the other functions which make up the overall package. The names of these other functions have been chosen to suggest their role in context (e.g., \textit{GET\&LIFE\&DATA}, \textit{GET\&REPAIR\&DATA}, \textit{SIMULATE}, \textit{PRODUCE\&REPORT}). All variables in the model are global in type, which paves the way for ease of communication among the various modules.
The last point to be made involves the amount of CPU time required to simulate with this model. Each 200-day simulation performed in the first sample session required about 12 CPU seconds on UCLA's IBM 360/91 computer. Of course, Monte Carlo simulations are known for the computer time they may require, and this is compounded here by the fact that APL is an interpretive language. Those who might choose to use this model for pedagogical purposes should take this rather large CPU-time requirement into account.
How many values can the lifetime random variable assume? (Enter an unsigned integer > 0): 1

Enter the various lifetime values: 225

Enter the corresponding relative frequencies (as percentages): 100

How many values can the repair time random variable assume? (Enter an unsigned integer > 0): Furnishsticks

Please enter positive integer(s): 5

Please enter positive integer(s): 0

Try again

How many values can the repair time random variable assume? (Enter an unsigned integer > 0): 4

Enter the various repair time values: 50

Enter 3 more values 60 70 80 90

Too many values entered, should have entered 3 values

Enter 3 more values 60 70 80

Enter the corresponding relative frequencies (as percentages): 15 20 25 20

Your relative frequencies sum to 80 percent

Try again

Enter the corresponding relative frequencies (as percentages): 20 25 30 25

Enter the repairmen’s wages ($/man-hour): 5.75

Negative values are not valid, try again

Enter the repairmen’s wages ($/man-hour): They earn too much! None numeric inputs are invalid, try again

Enter the repairmen’s wages ($/man-hour): 7.50

Enter the machine leasing charge ($/machine-day): 25.50

None numeric inputs are invalid, try again

Enter the machine leasing charge ($/machine-day): 25.50

Enter the production capacity (i.e., the number of machines which can be operated simultaneously): 0

ENTER THE LOST PRODUCTION PENALTY ($/MACHINE-HOUR): $10
NONE NUMERIC INPUTS ARE INVALID, TRY AGAIN
ENTER THE LOST PRODUCTION PENALTY ($/MACHINE-HOUR): 10

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: NONE
PLEASE ENTER POSITIVE INTEGER(S): 0
TRY AGAIN

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: 5

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 0

FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: 0
TRY AGAIN

FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: 500
PLEASE ENTER POSITIVE INTEGER(S): 250

--- REPORT ON SIMULATION RESULTS ---

NO. OF DAYS SIMULATED: 250
REPAIRMEN HIRED: 5
AVERAGE REPAIRMEN IN USE: 4.97
DAILY PAYROLL EXPENSE: $300.00
MACHINES LEASED: 0
DAILY LEASING EXPENSE: $0.00
AVERAGE DAILY UNDER-PRODUCTION LOSSES: $2,548.85
AVERAGE DAILY COST: $2,874.35

WHAT DO YOU WANT TO DO NEXT?
YOUR CHOICES ARE AS FOLLOWS...

IF YOU TRY TO INSERT A NEW VALUE FOR...
Hired Men-Hired
Leased Machines-Leased
Run-Length Duration of simulation

WHAT WILL HAPPEN IS...
Simulate Another simulation will be performed
Done Your use of the model will end
Explain This list of choices will be typed out

(YOU ONLY NEED TYPE IN THE FIRST LETTER IN EACH CASE)

WHAT IS YOUR CHOICE?

HIRED

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: 3

WHAT DO YOU WANT TO DO NEXT?

LEASED

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 5

WHAT DO YOU WANT TO DO NEXT?

SIMULATE

REPORT ON SIMULATION RESULTS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of days simulated</td>
<td>200</td>
</tr>
<tr>
<td>Repairmen hired</td>
<td>3</td>
</tr>
<tr>
<td>Average repairmen in use</td>
<td>2.43</td>
</tr>
<tr>
<td>Daily payroll expense</td>
<td>$162.00</td>
</tr>
<tr>
<td>Machines leased</td>
<td>5</td>
</tr>
<tr>
<td>Daily leasing expense</td>
<td>$150.00</td>
</tr>
<tr>
<td>Average daily under-production losses</td>
<td>$58.08</td>
</tr>
<tr>
<td>Average daily cost</td>
<td>$250.08</td>
</tr>
</tbody>
</table>

WHAT DO YOU WANT TO DO NEXT?
HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): 10

ENTER THE VARIOUS LIFETIME VALUES: 110 120 130 140 150 160 170 180
ENTER 2 MORE VALUES 190 200

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 3 6 8 12 21 17 11 9 8 5

HOW MANY VALUES CAN THE REPAIRTIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): 9

ENTER THE VARIOUS REPAIRTIME VALUES: 4 5 6 7 8 9 10 11 12

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 7 10 16 20 15 11 10 8 3

ENTER THE REPAIRMEN'S WAGES ($/MAN-HOUR): 6.75

ENTER THE MACHINE LEASING CHARGE ($/MACHINE-DAY): 30

ENTER THE PRODUCTION CAPACITY (I.E., THE NUMBER
OF MACHINES WHICH CAN BE OPERATED SIMULTANEOUSLY): 50

ENTER THE LOST PRODUCTION PENALTY ($/MACHINE-HOUR): 22.50

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: 5

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 3

FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: 200

---REPORT ON SIMULATION RESULTS---

NO. OF DAYS SIMULATED: 200

REPAIRMEN HIRED: 5

AVERAGE REPAIRMEN IN USE: 2.39

DAILY PAYROLL EXPENSE: $270.00

MACHINES LEASED: 3

DAILY LEASING EXPENSE: $90.00

AVERAGE DAILY
UNDER-PRODUCTION LOSSES: $67.12

AVERAGE DAILY COST: $367.12

WHAT DO YOU WANT TO DO NEXT?
LEASED

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?:  6

WHAT DO YOU WANT TO DO NEXT?

SIMULATE

---REPORT ON SIMULATION RESULTS---

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Days Simulated</td>
<td>200</td>
</tr>
<tr>
<td>Repairmen Hired</td>
<td>3</td>
</tr>
<tr>
<td>Average Repairmen in Use</td>
<td>2.38</td>
</tr>
<tr>
<td>Daily Payroll Expense</td>
<td>$162.00</td>
</tr>
<tr>
<td>Machines Leased</td>
<td>6</td>
</tr>
<tr>
<td>Daily Leasing Expense</td>
<td>$180.00</td>
</tr>
<tr>
<td>Average Daily Under-Production Losses</td>
<td>$4.82</td>
</tr>
<tr>
<td>Average Daily Cost</td>
<td>$196.82</td>
</tr>
</tbody>
</table>

WHAT DO YOU WANT TO DO NEXT?

DONE
PROGRAM DESCRIPTION

\[ \text{PRODUCE REPORT} \]

1. \( \text{NO. OF DAYS SIMULATED: I10 FMT(DAYS)} \)
2. \( \text{REPAIRMEN HIRED: I16 FMT(HIRED)} \)
3. \( \text{AVERAGE REPAIRMEN IN USE: F10.2 FMT(MEN\_USE\_CLOCK)} \)
4. \( \text{PAY = WAGE \_ RATE \_ HIRED} \)
5. \( \text{DAILY PAYROLL EXPENSE: \text{CP} \_ F13.2 FMT(PAY)} \)
6. \( \text{MACHINES LEASED: I16 FMT(LEASED)} \)
7. \( \text{DAILY LEASING EXPENSE: \text{CP} \_ F13.2 FMT(LEASED \_ LEASE \_ RATE)} \)
8. \( \text{LP \_ LOSS = (PROD \_ CAPACITY \_ CLOCK) - MACH \_ USE \_ LOSS \_ RATE \_ DAYS} \)
9. \( \text{AVERAGE DAILY} \)
10. \( \text{UNDER-PRODUCTION LOSSES: \text{CP} \_ F11.2 FMT(LP \_ LOSS)} \)
11. \( \text{AVERAGE DAILY COST: \text{CP} \_ F16.2 FMT(+/PAY, LEASE \_ RATE, LP \_ LOSS)} \)

\[ \text{EXPLAIN CHOICES} \]

1. \( \text{WHAT IS YOUR CHOICE?} \)
2. \( \text{IF YOU WILL BE GIVEN A CHANCE TO} \)
3. \( \text{SET A NEW VALUE FOR } \)
4. \( \text{HIRED MEN- HIRED} \)
5. \( \text{LEASED MACHINES - LEASED} \)
6. \( \text{RUN-LENGTH DURATION OF SIMULATION} \)
7. \( \text{WHAT WILL HAPPEN IS } \)
8. \( \text{ANOTHER SIMULATION WILL BE PERFORMED} \)
9. \( \text{DONE YOUR USE OF THE MODEL WILL END} \)
10. \( \text{EXPLAIN THIS LIST OF CHOICES WILL BE TYPED OUT} \)
11. \( \text{(YOU ONLY NEED TYPE IN THE FIRST LETTER IN EACH CASE)} \)
12. \( \text{WHAT IS YOUR CHOICE?} \)
\[ \text{GET\_WAGE\_DATA} \]

1. Note the use of "INP" in this function.
2. \( \text{WAGE\_RATE+1 INP PROMPT7} \)

\[ \text{GET\_LEASE\_DATA} \]

1. Note the use of "INP" in this function.
2. \( \text{LEASE\_RATE+1 INP PROMPT8} \)

\[ \text{GET\_PROD\_CAPACITY} \]

1. Note the use of "INP" in this function.
2. \( \text{REPEAT: REPEAT+x\_PO+(1>PROD\_CAPACITY+1+INP PROMPT10)'/TRY AGAIN'} \)

\[ \text{GET\_LOST\_PROD\_PENALTY} \]

1. Note the use of "INP" in this function.
2. \( \text{LOSS\_RATE+1 INP PROMPT9} \)

\[ \text{GET\_HIRE\_HIRE} \]

1. Note the use of "INP" in this function.
2. \( \text{REPEAT: REPEAT+x\_PO+(1>HIRE\_HIRE+1+INP PROMPT11)'/TRY AGAIN'} \)

\[ \text{GET\_MACHINES\_LEASED} \]

1. Note the use of "INP" in this function.
2. \( \text{REPEAT: REPEAT+x\_PO+(0>LEASED+1+INP PROMPT12)'/TRY AGAIN'} \)

\[ \text{GET\_SIM\_SPAN} \]

1. Note the use of "INP" in this function.
2. \( \text{REPEAT: REPEAT+x\_PO+(1>DAYS+1+INP PROMPT13)'/TRY AGAIN'} \)
\[ K = \text{CH} \text{ MT}; G \]

1. \[ K = (\text{MT} = \text{CH})/\text{pMT} \]
2. \[ G = \sqrt{K + k + 1} \pm 1\]
3. \[ \text{MT} = (\text{MT} \times \text{CH})/\text{MT} \]
4. \[ K = ((\text{CH} - G)/(K + g)) \pm (\text{MT} \times \text{CH})/\text{MT} \]

\[ K = \text{NIP}; P; S; I; D; E; T \]

1. \# \text{NUMERIC INPUT}; \text{AUTHOR: ROY SYKES, STSC.}
2. \# \text{4, 0 = -21}
3. \# E: P = 'INCOMPLETE NUMBER...PLEASE ENTER ', (1 + 0 = \rho \text{E}) p' : ', E
4. \# E = (1 + T + O \text{P} ' \text{N'} ) + P = 'E
5. \# \text{P} = \text{T} + \text{P}
6. \# (0 1 = \text{P} = \text{AKI P)/ 6 9}
7. \# T = 0 + '0123456789' - 'E': T
8. \# (13 1= (S + (P + T) + T) = (12) + T) + T = 15 11
9. \# (11 10 0 = T + O + '0123456789' (T)/6, 10 20
10. \# 0_1_0 = \text{PH} + \text{P}
11. \# P = (1 + D + T + 1 = 13) + T
12. \# (\text{V}/10 + \text{P} = (1 + 11 = 1 + P) + P) + \text{P}
13. \# (0 + (P), \rho T + D + T) + \text{P}
14. \# P = 10 + (1 + 10) \text{P}
15. \# (\sqrt{V / T} = 11 \text{ V} 10 + (1 + T + 10 + T) + \text{P}
16. \# (0 + \text{P} = (1 + T), \text{D}) + \text{P}
17. \# T + \text{P} = (1 0.1 \times \cdot S), \text{P}, \text{D}) \times 10 + \text{P}
18. \# + 19 + \text{T} = \text{LT}
19. \# + \text{LT}
20. \# + \text{T} + \text{T}

\[ \text{DRAW} = \text{CDF} \text{ SAMFCW RVVALS} \]

1. \# THIS FUNCTION RETURNS A VALUE DRAWN FROM THE POPULATION 'RVVALS' FOR
2. \# WHICH THE CORRESPONDING CUMULATIVE DISTRIBUTION FUNCTION IS 'CDF'
3. \# \text{DRAW} = \text{R}, \text{RVVALS} = (((1 + 0 + 100000) + 100000) + \text{CDF}); 1]

\[ \text{SEPARATE} \leq \leq \text{B} \]

1. \# \text{NUM} = (\text{B} + \text{AVI} \leq \text{DFI} \leq \text{2})
2. \# \text{ALPH} = (\text{B} + \text{AMI} \leq \text{2})
DEFINE MAJOR PROMPTS

THESE PROMPTS ARE USED BY THE INPUT FUNCTIONS CALLED IN LINES 7-15 AND 27-29 IN THE MASTER CONTROL FUNCTION.

DEFINE CR TO BE A CARRIAGE RETURN CHARACTER

PROMPT1+CR,'HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME? '
PROMPT1+PROMPT1+CR,'(ENTER AN UNSIGNED INTEGER > 0): '
PROMPT2+CR,'ENTER THE VARIOUS LIFETIME VALUES: ' 
PROMPT3+CR,'ENTER THE CORRESPONDING RELATIVE FREQUENCIES (AS PERCENTAGES): '
PROMPT4+CR,'HOW MANY VALUES CAN THE REPAIRTIME RANDOM VARIABLE ASSUME? '
PROMPT4+PROMPT4+CR,'(ENTER AN UNSIGNED INTEGER > 0): '
PROMPT5+CR,'ENTER THE VARIOUS REPAIRTIME VALUES: ' 
PROMPT6+CR,'ENTER THE CORRESPONDING RELATIVE FREQUENCIES (AS PERCENTAGES): '
PROMPT7+CR,'ENTER THE REPAIRMEN'S WAGES (/MAN-HOUR): ' 
PROMPT8+CR,'ENTER THE MACHINE LEASING CHARGE (/MACHINE-DAY): ' 
PROMPT9+CR,'ENTER THE LOST PRODUCTION PENALTY (/MACHINE-HOUR): ' 
PROMPT11+CR,'HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL? ' 
PROMPT12+'HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: '
PROMPT13+'FOR HOW MANY A-HOUR DAYS DO YOU WANT TO SIMULATE?: '

GETLIFE DATA

THIS FUNCTION MAKES DIRECT USE OF THE INPUT FUNCTIONS ''IPI'' AND ''INP''. (SEE THE MONOGRAPH APPENDICES FOR DETAILS)

REPEAT:+REPEAT×10'=+(1>LTIMES+1+IPI PROMPT1)+'/TRY AGAIN' ' 
LTVALS+LTIMES INP PROMPT2 
AGAIN:LTDIST='\0.01×LTDIST+LTIMES INP PROMPT3 
+:(("-1×LTDIST<0.99)∧("1+LTDIST<1.01)))/SET20NE 
'YOUR RELATIVE FREQUENCIES SUM TO ';100×"1+LTDIST;' PERCENT' 
'TRY AGAIN' ◄ AGAIN 
SET20NE:LTDIST[LTIMES]+1 ◄ 0

GETREPAIRDATA

THIS FUNCTION MAKES DIRECT USE OF THE INPUT FUNCTIONS ''IPI'' AND ''INP''. (SEE THE MONOGRAPH APPENDICES FOR DETAILS)

REPEAT:+REPEAT×10'=+(1>RTIMES+1+IPI PROMPT4)+'/TRY AGAIN' ' 
RTVALS+RTIMES INP PROMPT5 
AGAIN:RTDIST='\0.01×RTDIST+RTIMES INP PROMPT6 
+:((-1×RTDIST<0.99)∧("1+RTDIST<1.01)))/SET20NE 
'YOUR RELATIVE FREQUENCIES SUM TO ';100×"1+RTDIST;' PERCENT' 
'TRY AGAIN' ◄ AGAIN 
SET20NE:RTDIST[RTIMES]+1 ◄ 0

151
MASTERCONTROL

[1] THE MASTER CONTROL FUNCTION ROUTES CONTROL DIRECTLY TO


[4] FUNCTION "SIMULATE". THE VARIOUS FUNCTION NAMES

[5] ARE DESIGNED TO REVEAL THE FUNCTION ROLES.

[6] DEFINE MAJOR PROMPTS

[7] GET LIFE DATA

[8] GET REPAIR DATA

[9] GET WAGE DATA

[10] GET LEASING DATA


[12] GET LOST PRODUCTION PENALTY

[13] GET MACHINES HIRED

[14] GET MACHINES LEASED

[15] GET SIM SPAN

[16] SET CODE TO CONTROL PRINTING OF EXPLANATIONS

[17] (SEE LINES 22-25)

[18] CODE = 0

[19] SIM: SIMULATE

[20] PRODUCE REPORT

[21] MORE: QA, 'WHAT DO YOU WANT TO DO NEXT?', QA

[22] (CODE = 1)/SKIP

[23] EXPLAIN CHOICES

[24] CODE = 1

[25] SKIP: SKIP: p = (1z(CHAR*1+1)*e('HLRSD'))/TRY AGAIN

[26] (HIRE, LEASE, SPAN, SIM, DONE, EXPLAIN)[ 'HLRSD': CHAR]

[27] HIRE: GET MACHINES HIRED ◇ MORE

[28] LEASE: GET MACHINES LEASED ◇ MORE

[29] SPAN: GET SIM SPAN ◇ MORE

[30] EXPLAIN: EXPLAIN CHOICES ◇ MORE

[31] DONE: ◇
SIMULATE

[1] THE SIMULATION ITSELF IS PERFORMED IN THIS MODULE.

[2] INITIALIZE PERTINENT VARIABLES

[3] CLOCK+FIXQ+MACHUSE+MENUSE=O :: RCTIMES+BDTIMES+10

[4] INUSE+PRODCAPACITY

[5] TOP:BDTIMES+BUTIMES,(LTDIST SAMFCN LTVALS)X(1+71000)X1000

[6] +(PRODCAPACITY+BDTIMES)/TOP

[7] MENIDLE+HIRED+READY+LEASED+STOPTIME+DAYSX8

[8] FIND IMMEDIATE EVENT TIME; UPDATE USE STATISTICS AND CLOCK

[9] NEXTEVENT:ETIME+l/BUTIMES,RCTIMES,STOPTIME

[10] MENUSE=MENUSE+(HIRED-MENIDLE)+(JUMP+ETIME-CLOCK)


[12] USAGE+USAGE+INUSE+(ETIME-CLOCK) :: CLOCK+ETIME

[13] BRANCH TO IMMEDIATE EVENT, OR BRANCH OUT WHEN DONE

[14] +(ETIME+RCTIMES),ETIME+BDTIMES)/FIXED,FAILED + O

[15] UPDATING IN RESPONSE TO A MACHINE BREAKDOWN FOLLOWS

[16] FAILED:BDTIMES+(-1+(HOLD+BDTIMES)),(HOLD+BDTIMES+ETIME)+BDTIMES

[17] BRANCH IF FAILED MACHINE MUST QUEUE UP FOR REPAIR

[18] +Q2FIX+MENIDLE+O :: MENIDLE+MENIDLE+1

[19] RCTIMES+RCTIMES,CLOCK+RTDIST SAMFCN RVALS

[20] BRANCH IF NO OTHER MACHINES ARE READY FOR USE

[21] TESTREADY+READY=O/NOWREADY+READY+READY+1

[22] NEXTEVENT,BUTIMES+BUTIMES,CLOCK+LTDIST SAMFCN LTVALS

[23] Q2FIX+TESTREADYQ,FIXQ+FIXQ+1

[24] NOWREADY+NEXTEVENT,INUSE+INUSE+1

[25] UPDATING IN RESPONSE TO REPAIR COMPLETION FOLLOWS

[26] FIXEU:RCTIMES+(-1+(HOLD+RCTIMES)),(HOLD+RCTIMES+ETIME)+RCTIMES

[27] BRANCH IF THE REPAIRED MACHINE IS NEEDED NOW

[28] NEEDED+INUSE+INUSE+1 :: BDTIMES+BDTIMES,CLOCK+LTDIST SAMFCN LTVALS

[29] BRANCH IF ANOTHER FAILED MACHINE IS WAITING FOR THIS REPAIRMAN

[30] TEST:READY+READY+1

[31] NEEDED:INUSE+INUSE+1 :: BDTIMES+BDTIMES,CLOCK+LTDIST SAMFCN LTVALS

[32] BRANCH IF OTHER FAILED MACHINE IS WAITING FOR THIS REPAIRMAN

[33] TEST::NEXTREPAIR+FIXQ+O

[34] NEXTEVENT,MENIDLE+MENIDLE+1

[35] NEXTREPAIR:FIXQ+FIXQ+1

[36] NEXTEVENT,RCTIMES+RCTIMES,CLOCK+RTDIST SAMFCN RVALS

V
INTERACTIVE INVENTORY SIMULATOR

Ralph H. Sprague, Jr.
Professor
College of Business Administration
University of Hawaii
Honolulu, Hawaii

DESCRIPTION

Introduction

This system of APL programs allows students to obtain an optimum decision for economic order quantity (Q) and reorder point (R) in a stochastic inventory system by iterative search. The iterative search procedure allows the student to experiment with alternative methods of convergence on the optimum solution. One of the functions can also be used to reinforce the concept of appropriate choice of simulation run length based on the sampling distribution of resultant cost. The major functions in the system are:

- **SET** - To set initial values
- **SETRL** - To experiment with simulation run length
- **SIM** - To perform one simulation run
- **SEARCH** - To search for optimum, Q, R
- **ENTDDIST** - To enter probability distributions
- **ENTLDIST**
- **HPLOT** - To display sampling distributions
- **SHIST**

For a detailed description of each major function and its purpose, see the listing of **DESCRIBE**.

This exercise is useful in a variety of courses, including those in operations research, production/operations management,
computer simulation, and those dealing with computer-based decision models. The flexibility of its use makes it appropriate for graduate or undergraduate courses.

It can also be used by students with a variety of backgrounds and capabilities in APL. The student who knows little or no APL will call the first function and respond to the questions and instructions given by the programs. A student with some APL knowledge may enter the required data input using the assignment operator, use calculator mode to do some preliminary calculations on EOQ, and execute each major function when needed. The fully knowledgeable APL user may write his own functions utilizing SIM to implement a search algorithm, or perform a sensitivity analysis.

**Instructions for Use**

Several types of data must be entered before running the system. See the listing of SETHOW for specific variable names and descriptions.

1. **Cost Coefficients** - (3 cost coefficients are required): order cost, carrying cost, and stockout cost. Note that the latter two are unit daily cost. Only positive numbers are acceptable.

2. **Probability Distributions** - (2 empirical probability distributions are required): one for daily demand and one for order lead time. These must be entered by using the function SET or the functions ENTDDIST and ENLTDIST (which are called by SET).
3. Initial conditions: beginning inventory and initial values for Q and R. All three may be entered through SET or by the assignment operator directly. Only positive integers will be accepted.

4. Run length: a positive integer which can be entered using SET or by using the assignment operator. Only a positive integer is acceptable. The function SETRL allows the student to replicate simulation runs of a given length and display the sample of resultant average daily costs. This reveals the variability of response from a simulation with a short run length and allows the choice of appropriate length run. Note that the program will not accept more than 20 replications, or a run length of more than 250 days. Even these limits may be excessively expensive in CPU time on some APL installations.

After the required data are entered, the user begins the search for an optimum Q and R decision by executing the function SEARCH. This function accepts the value for Q and R, calls SIM and returns the average daily cost of that run. The student must first identify the appropriate search region and decide whether to conduct a coarse search over a large area or a fine search over a smaller area. He may use a primitive search algorithm which iteratively scans a relevant range for Q while R is fixed, then increments R and scans again over Q, etc. Alternatively, he may use a more sophisticated search algorithm such as the simplex technique described in Meier, et al.\(^1\) The student may be encouraged

to use a Q, R grid to track the progress of his search.

If the student writes his own function to conduct the search, he may wish to use the simulation function SIM directly. It requires all the variables described above and calculates the average daily cost of a run. Total cost of each type is also available in global variables if needed (see below). For a further description of the logic of the simulation module, see a similar FORTRAN model in Schriber.²

Summary of Functions and Variables

The two following tables summarize the functions and the global variables in the INVSIM workspace.

FUNCTIONS

<table>
<thead>
<tr>
<th>FUNCTION NAME</th>
<th>PURPOSE</th>
<th>CALLED BY</th>
<th>CALLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIBE</td>
<td>Gives basic description of INVSIM</td>
<td>SETHOW</td>
<td></td>
</tr>
<tr>
<td>DISTHOW</td>
<td>Gives instructions for entering probability distributions</td>
<td>SETHOW</td>
<td></td>
</tr>
<tr>
<td>ENTDIST</td>
<td>Enters probability distribution for demand</td>
<td>SET</td>
<td>DISTHOW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FUNCTION NAME</th>
<th>PURPOSE</th>
<th>CALLED BY</th>
<th>CALLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTLDIST</td>
<td>Enters probability distribution for lead time</td>
<td>SET</td>
<td>DISTHOW</td>
</tr>
<tr>
<td>FR</td>
<td>Constructs frequency vector</td>
<td>HPLOT</td>
<td></td>
</tr>
<tr>
<td>HPLOT</td>
<td>Constructs scale for histogram of sampling distribution</td>
<td>SETRL</td>
<td>SHIST, VS</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Permits iterative simulations with different Q, R</td>
<td>SIM</td>
<td></td>
</tr>
<tr>
<td>SET</td>
<td>Accepts input data</td>
<td>SETHOW, ENTLDIST, ENTLDIST, SETRL</td>
<td></td>
</tr>
<tr>
<td>SETHOW</td>
<td>Gives input instructions for the use of SET</td>
<td>DESCRIBE, SET</td>
<td>DISTHOW</td>
</tr>
<tr>
<td>SETRL</td>
<td>Replicates SIM with varying run length</td>
<td>SET</td>
<td>SIM, HPLOT</td>
</tr>
<tr>
<td>SHIST</td>
<td>Prints histogram of sampling distribution</td>
<td>HPLOT</td>
<td></td>
</tr>
<tr>
<td>SIM</td>
<td>Simulates the inventory system</td>
<td>SETRL, SEARCH</td>
<td></td>
</tr>
<tr>
<td>VS, AND</td>
<td>Utilities used for plotting</td>
<td>HPLOT</td>
<td></td>
</tr>
<tr>
<td>AFI, ΔFMT, ΔVI</td>
<td>Input/Output utilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>DESCRIPTION</td>
<td>INITIALIZED IN</td>
<td>USED IN</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
<td>----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ADC</td>
<td>Average Daily Cost</td>
<td>SIM</td>
<td>SEARCH, SIM</td>
</tr>
<tr>
<td>BINV</td>
<td>Beginning INVentory</td>
<td>SET</td>
<td>SIM</td>
</tr>
<tr>
<td>CAT</td>
<td>Number of CATegories in histogram</td>
<td>HPLoT</td>
<td>SHIST</td>
</tr>
<tr>
<td>CC</td>
<td>Carrying Cost</td>
<td>SET</td>
<td>SIM</td>
</tr>
<tr>
<td>CO</td>
<td>Cost per Order</td>
<td>SET</td>
<td>SIM</td>
</tr>
<tr>
<td>CS</td>
<td>Cost of Stockout</td>
<td>SET</td>
<td>SIM</td>
</tr>
<tr>
<td>DDIST</td>
<td>Demand DISTRIBUTion as a 100 element vector</td>
<td>ENTDDIST</td>
<td>SIM</td>
</tr>
<tr>
<td>FLAG</td>
<td>A FLAG to fix scale of histogram during one session</td>
<td>SETRL, HPLoT</td>
<td>HPLoT</td>
</tr>
<tr>
<td>INC</td>
<td>INCrement of histogram scale</td>
<td>HPLoT</td>
<td></td>
</tr>
<tr>
<td>LDIST</td>
<td>Lead time DISTRIBUTion as a 100 element vector</td>
<td>ENTLDIST</td>
<td>SIM</td>
</tr>
<tr>
<td>MIN</td>
<td>MINimum range of the histogram scale</td>
<td>HPLoT</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Order Quantity</td>
<td>SET</td>
<td>SIM</td>
</tr>
<tr>
<td>R</td>
<td>Reorder point</td>
<td>SET</td>
<td>SIM</td>
</tr>
<tr>
<td>RL</td>
<td>Run Length in days</td>
<td>SET, SETRL</td>
<td>SIM</td>
</tr>
<tr>
<td>S</td>
<td>Vector of frequencies, S, for histogram</td>
<td>HPLoT</td>
<td>SHIST</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>Vector of SAMPLE costs from replication of</td>
<td>SETRL</td>
<td>HPLoT</td>
</tr>
<tr>
<td>SCALE</td>
<td>SCALE for histogram</td>
<td>HPLoT</td>
<td>SHIST</td>
</tr>
<tr>
<td>TCC</td>
<td>Total Carrying Cost for 1 run</td>
<td>SIM</td>
<td></td>
</tr>
<tr>
<td>TCO</td>
<td>Total Cost of Order for 1 run</td>
<td>SIM</td>
<td></td>
</tr>
<tr>
<td>TCS</td>
<td>Total Cost for Stockout for 1 run</td>
<td>SIM</td>
<td></td>
</tr>
</tbody>
</table>
PROGRAM USAGE

SET
ENTRY OF PARAMETERS FOR INVENTORY SIMULATION
DO YOU NEED INSTRUCTIONS?
YES OR NO N
ENTER CARRYING COST PER UNIT PER PERIOD
? 2
ENTER COST PER ORDER PLACED
? 2.
ENTER STOCKOUT COST PER UNIT PER PERIOD
? .75
ENTER BEGINNING INVENTORY IN UNITS
? 15
ENTER PROBABILITY DISTRIBUTION FOR DEMAND
DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY DISTRIBUTION?
N
ENTER D PD
? 0 8
ENTER D PD
? 1 13
ENTER D PD
? 2 16
ENTER D PD
? 3 20
ENTER D PD
? 4 17
ENTER D PD
? 5 11
ENTER D PD
? 6 9
ENTER D PD
? 7 6
ENTER D PD
? 8 0
DO YOU WISH TO DISPLAY THE DISTRIBUTION?
YES OR NO Y
--- --- --- ---
0 | 8
1 | 13
2 | 16
3 | 20
4 | 17
5 | 11
6 | 9
7 | 6
SHALL WE GO ON?
YES OR NO Y
ENTER PROBABILITY DISTRIBUTION FOR LEAD TIME
DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY DISTRIBUTION?
N
ENTER LT P(LT)
? 3 30
ENTER LT P(LT)
? 4 50
ENTER LT P(LT)
? 5 20
ENTER LT P(LT)
? 0 0
DO YOU WISH TO DISPLAY THE DISTRIBUTION?
YES OR NO Y

---

3 | 30
4 | 50
5 | 20

SHALL WE GO ON?
YES OR NO Y

ENTER INITIAL VALUE FOR C
? 15

ENTER INITIAL VALUE FOR R
? 13

DO YOU WANT TO EXPERIMENT IN ORDER TO DETERMINE THE APPROPRIATE RUN LENGTH?
YES OR NO Y

ENTER TEST VALUE FOR RUN LENGTH IN PERIODS
? 20

HOW MANY RUNS OF THAT LENGTH DO YOU WISH TO TRY?
? 15

REPLICATION 1  AVG DAILY COST = 3.67
REPLICATION 2  AVG DAILY COST = 2.65
REPLICATION 3  AVG DAILY COST = 2.75
REPLICATION 4  AVG DAILY COST = 1.8375
REPLICATION 5  AVG DAILY COST = 2.0275
REPLICATION 6  AVG DAILY COST = 2.09
REPLICATION 7  AVG DAILY COST = 2.3
REPLICATION 8  AVG DAILY COST = 2.305
REPLICATION 9  AVG DAILY COST = 2.7725
REPLICATION 10 AVG DAILY COST = 2.6825
REPLICATION 11 AVG DAILY COST = 2.09
REPLICATION 12 AVG DAILY COST = 2.075
REPLICATION 13 AVG DAILY COST = 2.7925
REPLICATION 14 AVG DAILY COST = 2.6275
REPLICATION 15 AVG DAILY COST = 2.73

SAMPLE MEAN = 2.502166667
SAMPLE STD = 0.4380295525

WOULD YOU LIKE TO SEE A PLOT OF THIS SAMPLING DISTRIBUTION? Y
DO YOU WISH TO TRY ANOTHER RUN LENGTH? Y
ENTER TEST VALUE FOR RUN LENGTH IN PERIODS
? 50
HOW MANY RUNS OF THAT LENGTH DO YOU WISH TO TRY?
? 15
REPLICATION 1 AVG DAILY COST = 2.459
REPLICATION 2 AVG DAILY COST = 2.428
REPLICATION 3 AVG DAILY COST = 2.141
REPLICATION 4 AVG DAILY COST = 2.644
REPLICATION 5 AVG DAILY COST = 2.386
REPLICATION 6 AVG DAILY COST = 2.354
REPLICATION 7 AVG DAILY COST = 2.047
REPLICATION 8 AVG DAILY COST = 2.3
REPLICATION 9 AVG DAILY COST = 2.282
REPLICATION 10 AVG DAILY COST = 2.302
REPLICATION 11 AVG DAILY COST = 2.393
REPLICATION 12 AVG DAILY COST = 2.379
REPLICATION 13 AVG DAILY COST = 2.429
REPLICATION 14 AVG DAILY COST = 2.381
REPLICATION 15 AVG DAILY COST = 2.157
SAMPLE MEAN = 2.368466667
SAMPLE STD = 0.1786059577
WOULD YOU LIKE TO SEE A PLOT OF THIS SAMPLING DISTRIBUTION? Y
DO YOU WISH TO TRY ANOTHER RUN LENGTH? N
ENTER LENGTH OF EACH RUN IN PERIODS
? 40
PARAMETER ENTRY COMPLETE. WOULD YOU LIKE TO REVIEW THE VALUES?
? Y
CO = 2
CC = 0.2
CS = 0.75
BINV = 15
Q= 15
R= 13
RL= 40
DO YOU WANT TO PROCEED WITH THE SIMULATION?
IF SO, ENTER SEARCH, OTHERWISE RESTART PARAMETER SETTING PROCESS
BY ENTERING SET
PROGRAM DESCRIPTION

DESCRIBE

INTERACTIVE INVENTORY SIMULATION

THIS SET OF APL FUNCTIONS ALLOWS THE USER TO SEARCH FOR OPTIMUM
VALUES OF ORDER QUANTITY (Q) AND REORDER POINT (R) IN A STOCHASTIC
INVENTORY SYSTEM. AFTER ENTERING COST COEFFICIENTS, PROBABILITY
DISTRIBUTIONS, AND SIMULATION RUN PARAMETERS, THE USER CONDUCTS AN
INTERACTIVE SEARCH FOR THE OPTIMUM Q R PAIR. THE PRIMARY FUNCTIONS ARE
DESCRIBED BELOW.

SET

THIS PROGRAM ALLOWS THE USER TO ENTER THE DATA FOR THE SIMULATION
RUN. SPECIFICALLY IT:

1. ACCEPTS VALUES FOR UNIT CARRYING COST, UNIT STOCKOUT COST,
ORDER COST, AND BEGINNING INVENTORY.

2. CALLS TWO FUNCTIONS (ENTDDIST AND ENTLDIST) TO ENTER AND SHAPE
THE PROBABILITY DISTRIBUTIONS FOR DEMAND AND LEAD TIME.

3. ALLOWS THE USER TO CALL A SEPARATE FUNCTION (SETRL—SEE BELOW)
TO HELP DETERMINE THE APPROPRIATE LENGTH OF EACH SIMULATION RUN.

SETRL

THIS PROGRAM ALLOWS THE USER TO EXPERIMENT WITH SEVERAL VALUES FOR
THE RUN LENGTH OF THE SIMULATION. SPECIFICALLY, IT PERFORMS SEVERAL
RUNS OF A GIVEN LENGTH AND DISPLAYS THE RESULTANT AVERAGE DAILY COST
FOR EACH RUN. THE USER MAY DISPLAY THE COST FOR EACH SET OF
REPLICATIONS AS A FREQUENCY HISTOGRAM.

SEARCH

THIS PROGRAM ALLOWS THE USER TO ALTER THE VALUES OF Q AND R AND
PERFORM REPEATED SIMULATION RUNS. THE VALUE FOR AVERAGE DAILY COST IS
DISPLAYED AFTER EACH ITERATION.

SIM

THIS PROGRAM IS CALLED BY SETRL AND SEARCH TO PRODUCE THE AVERAGE
DAILY COST FOR A SINGLE SIMULATION RUN. ALL THE COST COEFFICIENTS,
PROBABILITY DISTRIBUTIONS, AND RUN PARAMETERS MUST BE DEFINED IN ADVANCE.

ENTDDIST AND ENTLDIST

THESE PROGRAMS ACCEPT VALUES OF AN EMPIRICAL PROBABILITY
DISTRIBUTION FOR DEMAND AND LEAD TIME. SEE ADDITIONAL INSTRUCTIONS BELOW

V
SETLR; REPL; COUNT; SM; STD

REPLICATES THE INVENTORY SIMULATION TO DEVELOP

A SAMPLING DISTRIBUTION OF AVERAGE COST. CALLS

HEPLOT ON USER OPTION TO DISPLAY HISTOGRAM OF

SAMPLE MEANS. Requires the public functions AYN

AND INF.

FLAG = 0

START: 

"ENTER TEST VALUE FOR RUN LENGTH IN PERIODS" 

RL = 1 INF 

(RL > 250)/ERR1

PRINT 

"HOW MANY RUNS OF THAT LENGTH DO YOU WISH TO TRY?"

REPL = 1 INF 

(REPL > 20)/ERR2

COUNT = 0

SAMPLE = 10

LOOP: SIM

COUNT = COUNT + 1

"REPLICATION "; COUNT; 

AVG DAILY COST = "; ADC

SAMPLE = SAMPLE, ADC

(REPL > COUNT)/LOOP

SM = (+/SAMPLE)*pSAMPLE

"SAMPLE MEAN = "; SM

STD = (((SM - SAMPLE - 1)) * (+/(SM - SAMPLE) * 2)) / 0.5

"SAMPLE STD = "; STD

YNT = AYN 

"WOULD YOU LIKE TO SEE A PLOT OF THIS SAMPLING DISTRIBUTION?"

(YNT = 0)/P2

HEPLOT

P2: YNT = AYN 

"DO YOU WISH TO TRY ANOTHER RUN LENGTH?"

(YNT = 1)/START

+ 0

ERR1: 

"THAT'S TOO LONG A RUN. CHOOSE A NUMBER < 250"

+ START

ERR2: 

"THAT'S TOO MANY REPLICATIONS. 20 OR LESS SHOULD BE SUFFICIENT"

+ P1

SIM; INV; ORDQ; DAYS; LT

THIS FUNCTION IS CALLED BY SEARCH AND SETLR TO GENERATE

AN AVERAGE DAILY COST FOR A SINGLE SIMULATION RUN OF

THE INVENTORY SYSTEM.

DAYS = TCO + TCC + TCS = 0

INV = BINV

ORDQ = (LONG / LDIST) * 0

CYCLE: + (RL * DAYS = DAYS + 1) / END

ADD INCOMING ORDER TO INVENTORY

INV = INV + (1 + ORDQ)

ORDQ = (1 + ORDQ)

SUBTRACT DAILY DEMAND FROM INVENTORY

INV = INV - DDIST / 1001

PLACE ORDER IF INVENTORY + COMING ORDERS < R

(R = INV + ORDQ)

PLACE ORDER, UPDATE ORDER QUEUE AND TOTAL ORDER COST

LT = LDIST / 1001

ORDQ[LT] = ORDQ[LT] + Q

TCO = TCO + CO

UPDATE CARRYING COST OR STOCKOUT COST

ON = + (INV = 0), (INV > 0))/CYCLE, GT

CYCLE, TCS + TCS * CS * INV

GT = CYCLE, TCC + TCC * CC * INV

CALCULATE AVERAGE DAILY COST

END; ADC = +/(TCS, TCO, TCC) * RL

END;
\* V HPLOT; RANGE
\* ACCEPTS SAMPLE OF COST MEANS FROM SETRL, SETS A SCALE.
\* CONSTRUCTS A FREQUENCY DISTRIBUTION, AND CALLS SHIST
\* TO PRINT A HISTOGRAM.  FLAG IS SET TO
\* USE THE SAME SCALE FOR ALL RUNS OF A
\* SINGLE SESSION.  REQUIRES THE USE OF THE
\* PUBLIC FUNCTION VS.
\* \+ (FLAG=1)/PL
\* \+ (FLAG+1)/PL
\* RANGE=((/SAMPLE)-1)/SAMPLE
\* MIN=((/SAMPLE)-0.25*RANGE
\* CAT=((/SAMPLE)+1)
\* \+ (CAT=15)/FORCE;
\* \+ (CAT=15)/FORCE1
\* SETINC:INC= RANGE/CAT
\* SCALE=INC; ((1.25*CAT))=MIN
\* PL=MIN,INC,CAT) FR SAMPLE
\* \+ S=SCALE-pS)pO
\* \+ 40 SHIST S VS SCALE
\* \+ 0
\* \+ (CAT=6)/FORCE;
\* \+ (CAT=15)/FORCE1
\* \+ SETINC

\* V SEARCH; T
\* ACCEPTS A VALUE FOR Q AND R, CALLS SIM
\* TOO RETURN THE AVERAGE DAILY COST, USED TO ITERATIVELY
\* SEARCH FOR THE OPTIMUM VALUES OF Q AND R.
\* Requires the public functions AYN and IFF
\* T=0
\* 'HAVE YOU ENTERED ALL THE NECESSARY DATA?'
\* YNT=AYN 'Y'
\* \+ (YNT=1)/G01
\* 'PLEASE ENTER DATA.  TYPE DECBIEF FOR INSTRUCTIONS'
\* \+ 0
\* G01: 'INITIAL VALUES FOR Q AND R ARE:'
\* Q= 'Q0'; R= 'R0'
\* 'SHALL WE PROCEED?'
\* YNT=AYN 'Y'
\* \+ (YNT=1)/LOOP
\* END: 'DONE'
\* \+ 0
\* LOOP; SIM
\* 'AVERAGE DAILY COST = ' ABD
\* \+ '
\* \+ ENTER Q R'
\* \+ T=2 INP '?'
\* \+ (A/T=0)/END
\* \+ 0=+T
\* \+ R=-1+T
\* \+ LOOP
SET

1. IT REQUIRES THE FOLLOWING USER-DEFINED FUNCTIONS:
   1. ENTDDIST, ENLDDIST, SETINV, SETRL. ALSO REQUIRED ARE THE
   2. PUBLIC FUNCTIONS AYN AND IFI.

2. THIS FUNCTION ACCEPTS USER INPUT TO SET VALUES FOR THE SIMULATION
   RUN.

3. 'ENTRY OF PARAMETERS FOR INVENTORY SIMULATION'
4. 'DO YOU NEED INSTRUCTIONS?'
5. YNT-AYN 'YES OR NO'
6. →(YNT=O)/PROCEED1

7. SETINV
8. PROCEED1: 'ENTER CARRYING COST PER UNIT PER PERIOD'
9. CC+1 INF '?'
10. 'ENTER COST PER ORDER PLACED'
11. CO+1 INF '?'
12. 'ENTER STOCKOUT COST PER UNIT PER PERIOD'
13. CS+1 INF '?'
14. 'ENTER BEGINNING INVENTORY IN UNITS'
15. BINV+1 INF '?'
16. 'ENTER PROBABILITY DISTRIBUTION FOR DEMAND'
17. ENTDIST
18. 'ENTER PROBABILITY DISTRIBUTION FOR LEAD TIME'
19. ENLDDIST
20. PROCEED2: 'ENTER INITIAL VALUE FOR Q'
21. Q+1 INF '?'
22. 'ENTER INITIAL VALUE FOR R'
23. R+1 INF '?'
24. 'DO YOU WANT TO EXPERIMENT IN ORDER TO DETERMINE THE'
25. 'APPROPRIATE RUN LENGTH?'
26. YNT-AYN 'YES OR NO'
27. →(YNT=O)/PROCEED3

28. SETRL
29. PROCEED3: 'ENTER LENGTH OF EACH RUN IN PERIODS'
30. RL+1 INF '?'
31. 'PARAMETER ENTRY COMPLETE. WOULD YOU LIKE TO REVIEW THE VALUES?'
32. YNT-AYN 'YES OR NO'
33. →(YNT=O)/PROCEED4

34. 'CC =~ CO'
35. 'CC =~ CS'
36. 'BINV =~ FINV'
37. 'Q =~ Q'
38. 'R =~ R'
39. 'RL =~ RL'
40. PROCEED4: 'DO YOU WANT TO PROCEED WITH THE SIMULATION?'
41. 'IF SO, ENTER SEARCH, OTHERWISE RESTART PARAMETER SETTING PROCESS'
42. 'BY ENTERING SET'

43.
THE FOLLOWING DATA MUST BE ENTERED FOR THE INVENTORY SIMULATION.

DATA NAME DESCRIPTION
-------- --------
CO ORDER COST: THE COST OF PLACING AN ORDER
CC CARRYING COST: THE COST OF CARRYING ONE UNIT OF INVENTORY FOR ONE DAY
CS STOCKOUT COST: THE COST OF BACK ORDERING ONE UNIT FOR ONE DAY
BINV BEGINNING INVENTORY: THE NUMBER OF UNITS IN INVENTORY AT THE BEGINNING OF EACH SIMULATION RUN
Q QUANTITY ORDERED: THE NUMBER OF UNITS TO BE ORDERED EACH TIME AN ORDER IS PLACED
R REORDER POINT: THE STOCK LEVEL AT WHICH ORDER PLACEMENT IS TRIGGERED
RL RUN LENGTH: THE NUMBER OF PERIODS IN EACH SIMULATION RUN
DDIST THE PROBABILITY DISTRIBUTION FOR DEMAND
LDIST THE PROBABILITY DISTRIBUTION FOR LEAD TIME

All the data can be entered by typing "SET" and answering the questions. Enter only numeric data. To interrupt a run, hit the attention key, type the right arrow (>) or enter a backspace U.

As an alternative to using SET, all the data items (except the probability distributions) can be entered or changed by using the APL assignment operator (for instance CO=.25). To enter the probability distributions (without using the "SET" function) type ENTDDIST and ENTLDIST for demand and lead time distributions respectively and follow the instructions given below. After the data and distributions have been entered, type SETRL to determine the appropriate run length.

Finally, type SEARCH to interactively search for the optimum Q and R.
INSTRUCTIONS FOR ENTERING PROBABILITY DISTRIBUTIONS

ENTDDIST and ENTLDIST require entry of an empirical probability distribution of the following form.

<table>
<thead>
<tr>
<th>D</th>
<th>P(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.25</td>
</tr>
<tr>
<td>1</td>
<td>.50</td>
</tr>
<tr>
<td>2</td>
<td>.25</td>
</tr>
</tbody>
</table>

In response to the prompt "ENTER D PD", type one demand figure and its probability in whole percentage form (without decimal, two digits) with a space in between. For example:

ENTER D PD
0 25
ENTER D PD
1 50
ENTER D PD
2 25
ENTER D PD
0 0

After the last entry has been made, enter 0 0 to exit.

ENTDDIST

REQUIRES THE PUBLIC FUNCTIONS AYN, ΑΡΜΤ, AND INP.
ACCEPTS PROBABILITY DISTRIBUTION FOR DEMAND AND
CONSTRUCTS A 100 ELEMENT VECTOR FOR RANDOMLY
GENERATING DEMAND IN SIMULATION.
'DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY DISTRIBUTION?'

YNT=AYN 'Y'

START:DDIST+10

LOOP:'ENTER D PD'

T-2 INP 'Y'

DDIST-DDIST,(-T)D+T

TEST:=(pDDIST)#100)/ERR

'DO YOU WISH TO DISPLAY THE DISTRIBUTION?'

YNT-AYN 'YES OR NO'

IN=(((pIN)+2),2)pIN

'D | P(D)'

'SHALL WE GO ON?'

YNT=AYN 'YES OR NO'

*YNT=1)/O

'SHALL WE GO ON?'

YNT-AYN 'YES OR NO'

*YNT=1)/O

'SO, START OVER'

START

ERR: 'PROBABILITY PERCENTAGES DO NOT ADD TO 100, START OVER'

START
ENTLDIST;IN;T
[1] *REQUIRES THE PUBLIC FUNCTIONS AYN, @FMT, *AND INP.
[2] *ACCEPTS PROBABILITY DISTRIBUTION FOR LOAD TIME AND
[3] *CONSTRUCTS A 100 ELEMENT VECTOR FOR RANDOMLY
[4] *GENERATING LOAD TIME IN THE SIMULATION.
[5] 'DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY'
[6] 'DISTRIBUTION'
[7] YNT+AYN '?'
[8] *((YNT=1),YNT=0))/OUT,START
[9] OUT:DISTHOW
[10] START: LDIST+10
[12] LOOP:'ENTER LT P(LT)'
[13] T=2 INP '?'
[14] +((A/T=0)/TEST
[15] IN+IN,T
[16] LDIST=LDIST,(~1T)P1+T
[17] LOOP
[18] TEST:*((LDIST)*100)/ERR
[19] 'DO YOU WISH TO DISPLAY THE DISTRIBUTION?'
[20] YNT+AYN 'YES OR NO'
[21] *((YNT=0))/0
[22] IN+((P1N)+2),2P1N
[23] 'LT | PLT ' DFM IN
[24] 'SHALL WE GO ON?'
[25] YNT+AYN 'YES OR NO'
[26] *((YNT=1))/0
[27] 'OK, START OVER'
[28] *START
[29] STA:'PROBABILITY PERCENTAGES DO NOT ADD TO 100, START OVER'
[30] *START

171
FINANCIAL RATIO CALCULATIONS

Wilma D. Stricklin
Professor of Management
College of Business Administration
Northern Arizona University
Flagstaff, Arizona

DESCRIPTION

This program performs a number of ratio calculations which are typically used to gauge the financial effectiveness of an organization. These ratios are:

- \( \frac{\text{Net Income}}{\text{Total Assets}} \) - Return on investment as a function of assets
- \( \frac{\text{Net Income}}{\text{Total Common Equity}} \) - Return on investment as a function of equity
- \( \frac{\text{Net Income}}{\text{Number of Shares}} \) - Earnings per share
- \( \frac{\text{Net Income}}{\text{Total Sales}} \) - Profits as a function of sales
- \( \frac{\text{Long Term Debt}}{\text{Total Assets}} \) - Long term debt as a function of assets
- \( \frac{\text{Total Sales}}{\text{Total Assets}} \) - Sales as a function of assets

The program is initiated by typing `EFFRATIO` and then following the instructions given. Any number of years' worth of data can be entered; but, beyond five, the data inputting can be tedious and error prone.

In the sample shown in the following section, the data were taken from the Boise-Cascade Case found in Robert L. Katz, *Cases and Concepts in Corporate Strategy*, Englewood Cliffs, N. J.: Prentice-Hall, 1970.
**EFFRATIO**

**OBSERVE THE FOLLOWING INSTRUCTIONS IN ENTERING DATA:**
A. ROUND YOUR DATA TO THE NEAREST THOUSAND DOLLARS.
B. ENTER 0(ZERO) IN EACH INSTANCE WHERE DATA IS UNKNOWN OR UNAVAILABLE.
C. USE THE "-" SIGN (OVER THE 2-KEY) TO MARK NEGATIVE NUMBERS.
D. DO NOT USE COMMAS TO SEPARATE NUMBERS.

**ENTER YEARS FOR WHICH YOU HAVE DATA:** 1962 1963 1964 1965 1966

**ENTER TOTAL ASSETS FOR EACH YEAR:**
- 124382
- 130456
- 293475
- 370522
- 454250

**ENTER NET INCOME AFTER TAXES FOR EACH YEAR:**
- 5055
- 7390
- 14947
- 17888
- 17014

**ENTER TOTAL LONG TERM DEBT FOR EACH YEAR:**
- 71548
- 63815
- 117127
- 156921
- 193909

**ENTER TOTAL COMMON EQUITY FOR EACH YEAR:**
- 52834
- 64663
- 166496
- 192751
- 193909

**LENGTH ERROR...REENTER 5 VALUES.**
- 52834
- 64663
- 166496
- 172251
- 192751

**ENTER TOTAL SHARES OF COMMON STOCK FOR EACH YEAR:**
- 3741
- 4001
- 4387
- 4376
- 9005

**ENTER TOTAL SALES FOR EACH YEAR:**
- 176015
- 219361
- 363581
- 422911
- 492466

**ENTER HEADING FOR YOUR REPORT:** BOISE CASCADE CORP.

**EFFECTIVENESS RATIOS—BOISE CASCADE CORP.**

<table>
<thead>
<tr>
<th>YEAR</th>
<th>ROI-A</th>
<th>ROI-EQ</th>
<th>EPS</th>
<th>PROF/SLS</th>
<th>DT/ASSET</th>
<th>SLS/ASSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>1962</td>
<td>0.0406</td>
<td>0.0957</td>
<td>1.3512</td>
<td>0.0287</td>
<td>0.5752</td>
<td>1.4151</td>
</tr>
<tr>
<td>1963</td>
<td>0.0566</td>
<td>0.1149</td>
<td>1.8470</td>
<td>0.0337</td>
<td>0.4892</td>
<td>1.6815</td>
</tr>
<tr>
<td>1964</td>
<td>0.0599</td>
<td>0.0898</td>
<td>3.4071</td>
<td>0.0411</td>
<td>0.3991</td>
<td>1.2389</td>
</tr>
<tr>
<td>1965</td>
<td>0.0483</td>
<td>0.1038</td>
<td>4.0798</td>
<td>0.0423</td>
<td>0.4505</td>
<td>1.1414</td>
</tr>
<tr>
<td>1966</td>
<td>0.0375</td>
<td>0.0883</td>
<td>1.8894</td>
<td>0.0345</td>
<td>0.4269</td>
<td>1.0841</td>
</tr>
</tbody>
</table>

**DO YOU WISH TO CONTINUE? NO**
PROGRAM DESCRIPTION

\[ \text{EFFRATIO} \times \text{M} \times \text{YRS} \times \text{TA} \times \text{NP} \times \text{LTD} \times \text{CE} \times \text{CMS} \times \text{SA} \times \text{CO} \]

[1] THE FOLLOWING PUBLIC FUNCTIONS ARE REQUIRED IN ORDER
[2] TO USE THIS PROGRAM: AKI, CENTER, COLNAMES, AFMT, INP,
[3] AND IPI.
[5] \\
[6] OBSERVE THE FOLLOWING INSTRUCTIONS IN ENTERING DATA:
[7] A. ROUND YOUR DATA TO THE NEAREST THOUSAND DOLLARS,
[8] B. ENTER 0(ZERO) IN EACH INSTANCE WHERE DATA IS UNKNOWN OR UNAVAILABLE,'
[9] C. USE THE "SIGN (OVER THE 2-KEY) TO MARK NEGATIVE NUMBERS,'
[10] D. DO NOT USE COMMAS TO SEPARATE NUMBERS.'
[12] TOP:N+YRS+IPI 'ENTER YEARS FOR WHICH YOU HAVE DATA:
[13] THIS SECTION ACCEPT INPUT, TESTS EACH SET FOR ACCURACY OF NUMBER OF INPUTS, AND CYCLES FOR CORRECT DATA.
[14] ENTER TOTAL ASSETS FOR EACH YEAR:
[15] ENTER TOTAL INCOME AFTER TAXES FOR EACH YEAR:
[16] ENTER TOTAL LONG TERM DEBT FOR EACH YEAR
[17] ENTER TOTAL COMMON EQUITY FOR EACH YEAR:
[18] ENTER TOTAL SHARES OF COMMON STOCK FOR EACH YEAR:
[19] ENTER TOTAL SALES FOR EACH YEAR:
[20] THIS CREATES A MATRIX OF RATIOS RESULTING FROM A DIVISION OF VECTORS.
[21] \( \text{CO} = (\text{NI}, \text{NI}, \text{WI}, \text{NI}, \text{LTD}, \text{SA}) \times (\text{TA}, \text{CE}, \text{CMS}, \text{SA}, \text{TA}, \text{TA}) \)
[22] THIS ESTABLISHES REPORT FORMAT.
[23] TITLE=AKI 'ENTER HEADING FOR YOUR REPORT:
[24] FORMAT='I5,6F10.4'
[25] FORMAT CENTER 'EFFECTIVENESS RATIOS---',TITLE
[26] FORMAT COLNAMES 'XXEALKROI-AXKOI-EQXEPSXPROF/SLSXDT/ASSETXSL/ASSETX
[27] FORMAT AFMT(YRS;WCC)
[28] THIS PERMITS THE USER TO REPEAT THE PROGRAM FOR ANOTHER SET OF DATA, FOR
[29] A COMPETITOR, FOR THE INDUSTRY, ETC.
[30] \( \text{YN} = \text{'DO YOU WISH TO CONTINUE?'} \) / TOP
COMPUTATION OF A PRICE INDEX
BY THE METHOD OF THE WEIGHTED GEOMETRIC MEAN

Bernard A. Thielges
Assistant Professor
Management Department
College of Business Administration
Marquette University
Milwaukee, Wisconsin

DESCRIPTION

Statement of the Problem

The computation of a price index can be done by various averaging techniques. The arithmetic mean, however, is the average most often used because of its simplicity and ease in calculation. Various weighting methods can also be used with the arithmetic mean, but the one most often used is the method which uses the base year values as the weights.

The method of the arithmetic mean together with base year values as weights provides a good index. There are deficiencies, however. It is known that there is an upward bias to the arithmetic mean. On the other hand, the use of base year values as weights provides a downward bias by giving too much weight to those items having the smallest change in price. At times, one bias may tend to offset the other; but in other instances, this will not be the case.

In order to compute a price index which is consistently more accurate, the geometric mean, which is an unbiased mean, might be used. And, to provide for a more appropriate weighting system, it is suggested that the average of the values computed for all years, from the base year to the current year, be used. The computation of the price index by this method is considerably longer and more sophisticated. However, the initial computation and subsequent
updating are readily facilitated through the use of the computer.

Program Description

The program PINDEX computes a price index for selected items. The index is computed for the final year of data submitted, using the first year as the base year. The formula used in the computation is

\[
\text{Price Index} = \frac{\sum w \left( \log \frac{p_1}{p_0} \right)}{\sum w} \sum_{i=1}^{N} p_iq_i
\]

where, \(w\) (Average value of all years for each item in the index) = \(\frac{\sum_{i=1}^{N} p_iq_i}{N}\)

- \(p_i\) = price for the ith year of data
- \(q_i\) = quantity for the ith year of data
- \(p_1\) = price for the last year of data submitted
- \(p_0\) = price for the first (base) year of data submitted
- \(N\) = number of years for which data is submitted.

To begin the computation, the data must be entered in the following fashion. First, the years for which data are to be submitted are entered, i.e., 1971 1972 1973, and 1974. These data on years are needed for labeling the table in the print-out. The number of items to be included in the index is entered as a numberal followed by the individual item names when asked for by the program. Item names should not exceed 16 characters in length, should be only one word (compound words can be hyphenated or run together), and should be separated by blanks (do not use commas).
Quantity data is then entered by year for all items, followed by price data by year for all items. These entries constitute the quantity and price matrices. Data are entered for all items by year and appear as shown in the sample. The program puts the data in proper matrix form for calculations.

Internal computation of the index starts with the calculation of the ratio between the final year price data and the first year price data. The first year of data is therefore the base year. The logarithm to the base 10 of this ratio is then determined.

The weight to be used is calculated by first multiplying the price matrix by the quantity matrix and obtaining a value matrix. This matrix lists the value of each individual item in the index by years. The average value for each specific item is calculated by summing each year’s value and then dividing by the number of years. The resultant average is used as the weight.

Next, the logarithm of the price ratio for each item is multiplied by its weight. This weighted value is summed for all items and divided by the sum of the weights, i.e., the sum of the average values for all of the years for all of the items. The final computation of the price index is made by finding the antilog of the weighted data.

The print-out of data includes a repeat of the price and quantity matrix as well as the final price index as a percent. The user should be aware that if data for many years are submitted or if the data are fairly large, then the available space might not be sufficient to print out the price and quantity matrices. In this case, an asterisk will appear wherever there is insufficient space in the print-out.
PROGRAM USAGE

PINDEX

ENTER THE YEARS FOR WHICH THE INDEX IS TO BE COMPUTED, EG, 1973 1974 1975
0: 1973 1974 1975
ENTER THE NUMBER OF ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, 6
0: 6
ENTER THE NAMES OF THE ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, LUMBER CEMENT NAILS ASPHALTSHINGLES. NOTE THAT ANY ITEM CONTAINING TWO OR MORE WORDS IS TO BE JOINED TOGETHER TO MAKE ONE WORD. NO ITEM SHOULD EXCEED SIXTEEN CHARACTERS IN LENGTH.
CORN WHEAT OATS BARLEY RYE RICE
ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD, FOR 1973
0: 4084 1316 927 392 33 76
ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD, FOR 1974
0: 4760 1522 789 373 24 89
ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD, FOR 1975
0: 4375 1570 930
INCORRECT NUMBER OF ITEMS. TRY AGAIN
0: 4375 1570 930 418 23 105
ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 1973
0: 1.16 1.35 .62 1.03 .97 4.93
ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 1974
0: 1.04 1.39 .66 1.00 1.07 4.97
ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 1975
0: 1.05 1.22 .59 .88 .99 5
### Production Data

<table>
<thead>
<tr>
<th>Item</th>
<th>1973</th>
<th>1974</th>
<th>1975</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corn</td>
<td>4,084.00</td>
<td>4,760.00</td>
<td>4,375.00</td>
</tr>
<tr>
<td>Wheat</td>
<td>1,316.00</td>
<td>1,522.00</td>
<td>1,570.00</td>
</tr>
<tr>
<td>Oats</td>
<td>927.00</td>
<td>789.00</td>
<td>930.00</td>
</tr>
<tr>
<td>Barley</td>
<td>392.00</td>
<td>373.00</td>
<td>418.00</td>
</tr>
<tr>
<td>Rye</td>
<td>33.00</td>
<td>24.00</td>
<td>23.09</td>
</tr>
<tr>
<td>Rice</td>
<td>76.00</td>
<td>89.00</td>
<td>105.00</td>
</tr>
</tbody>
</table>

### Price Data

<table>
<thead>
<tr>
<th>Item</th>
<th>1973</th>
<th>1974</th>
<th>1975</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corn</td>
<td>1.16</td>
<td>1.04</td>
<td>1.05</td>
</tr>
<tr>
<td>Wheat</td>
<td>1.35</td>
<td>1.39</td>
<td>1.22</td>
</tr>
<tr>
<td>Oats</td>
<td>0.62</td>
<td>0.66</td>
<td>0.59</td>
</tr>
<tr>
<td>Barley</td>
<td>1.03</td>
<td>1.00</td>
<td>0.88</td>
</tr>
<tr>
<td>Rye</td>
<td>0.97</td>
<td>1.07</td>
<td>0.99</td>
</tr>
<tr>
<td>Rice</td>
<td>4.93</td>
<td>4.97</td>
<td>5.00</td>
</tr>
</tbody>
</table>

### Value Data

<table>
<thead>
<tr>
<th>Item</th>
<th>1973</th>
<th>1974</th>
<th>1975</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corn</td>
<td>4,737.44</td>
<td>4,950.40</td>
<td>4,593.75</td>
</tr>
<tr>
<td>Wheat</td>
<td>1,776.60</td>
<td>2,115.58</td>
<td>1,919.40</td>
</tr>
<tr>
<td>Oats</td>
<td>574.74</td>
<td>520.74</td>
<td>548.70</td>
</tr>
<tr>
<td>Barley</td>
<td>403.76</td>
<td>373.00</td>
<td>367.84</td>
</tr>
<tr>
<td>Rye</td>
<td>32.01</td>
<td>25.68</td>
<td>22.77</td>
</tr>
<tr>
<td>Rice</td>
<td>374.68</td>
<td>442.33</td>
<td>525.00</td>
</tr>
</tbody>
</table>

Price Index (Percent) is: 91.14895115
PROGRAM DESCRIPTION

\[ P \text{INDEX};R \]

\*PREPARED BY:

\*BERNARD A. THIELGES

\*ASSISTANT PROFESSOR OF MANAGEMENT

\*COLLEGE OF BUSINESS ADMINISTRATION

\*MARRIETT UNIVERSITY

\*MILWAUKEE, WISCONSIN 53233

\*THIS PROGRAM COMPUTES A PRICE INDEX BY THE METHOD OF THE WEIGHTED

\*GEOMETRIC MEAN.

\*INPUT: PRICE AND QUANTITY DATA FOR \( N \) YEARS. \( N \)OTE THAT BOTH ITEMS

\*ARE ENTERED AS ROW VECTORS. ITEM NAMES ARE TRANSPOSED AS NECESSARY INTO

A

\*COLUMN MATRIX FOR DATA PRINTOUT.

\*THE PRICE RATIOS FOR ALL ITEMS INCLUDED IN DATA ARE COMPUTED WITH THE

\*LAST YEAR OF DATA REFERENCED TO THE FIRST YEAR OF DATA AS BASE YEAR.

\*THE PRICE RATIOS ARE WEIGHTED BY THE AVERAGE OF ALL OF THE YEARS VALUE

\*OF THE ITEM FROM THE FIRST YEAR THROUGH THE LAST YEAR OF THE DATA

\*SUBMITTED. THE GEOMETRIC MEAN OF THE WEIGHTED PRICE RATIOS IS THEN

\*COMPUTED. THIS WEIGHTED PRICE RATIO IS THEN DIVIDED BY THE SUM OF THE

\*WEIGHTS (TOTAL OF THE AVERAGE VALUES) TO PRODUCE THE PRICE INDEX.

\*THIS PROGRAM REQUIRES THE USE OF \&FMT, LIST, AND MIN

\*CTR=1

\*ENTER THE YEARS FOR WHICH THE INDEX IS TO BE COMPUTED, EG, 1973 1974 19

\*75

\*N=pNN+[]

\*ENTER THE NUMBER OF ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, 6

\*M=[]

\*PRICE=QUAN=(M,N)p0

\*ENTER THE NAMES OF THE ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, 'LUMBER CEMENT NAILS ASPHALT SHINGLES'. \( N \)OTE THAT ANY ITEM CONTAINING \( N \)O MORE WORDS IS TO BE JOINED TOGETHER TO MAKE ONE \( N \)ORD. \( N \)O ITEM

\*SHOULD EXCEED SIXTEEN CHARACTERS IN LENGTH.'

\*LIST=MIN ';',m \( m \)CHANGES A ROW OF WORDS INTO A COLUMN MATRIX OF THE WORDS.

\*LIST=LIST,((1+pLIST),(16-1+pLIST))o' ' \( n \)AKES LITERAL MATRIX CONTAINING I

\*EM NAMES

\*AND PADS WITH BLANKS TO PRODUCE A LITERAL MATRIX OF THE FOLLOWING FORM: L

\*IST[N;16].

\*L1:ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD

\*FOR 'NN[CTR]

\*L11:=(M=pVV+,[])/AGAIN

\*N=A='INCORRECT NUMBER OF ITEMS. TRY AGAIN'

\* L1

\*AGAIN:QUAN[CTR]+((pVV),1)pVV

\*CTR=CTR+1

\*L11*(CTR)\leq N

\*CTR=1

\*L2:ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 'NN[CTR]

\*L111:=(M=pVV+,[])/AGAIN1

\*A

\*L11

\*AGAIN1:PRICE[;CTR]+((pVV),1)pVV

\*CTR=CTR+1

\*L111*(CTR)\leq N

\*VALUE=QUAN\cdot PRICE

\*LOGRAT=(M,1)\cdot 100(,PRICE[;N]\cdot PRICE[;1]

\*AVEVAL=(+/VALUE)\div N
\[ \text{WTRATIO} = \frac{\text{AVEVAL} \times \text{LOGRAT}}{\text{LOGPINDEX}} \]
\[ \text{R} = (10 \times \text{LOGPINDEX}) \times 100 \]

'PRODUCTION DATA'
\[ '4A1.X12.10112' \ \alpha \text{fmt}(\{1 \ 4 \ 'ITEM'\};\{1,NN\}NN) \]

'PRICE DATA'
\[ '4A1.X12.10112' \ \alpha \text{fmt}(\{1 \ 4 \ 'ITEM'\};\{1,NN\}NN) \]

'VALUE DATA'
\[ '4A1.X12.10112' \ \alpha \text{fmt}(\{1 \ 4 \ 'ITEM'\};\{1,NN\}NN) \]

'PRICE INDEX (PERCENT) IS: ' \ R
4.0

UCLA APL LIBRARY FUNCTIONS
SYNTAX: A PLOT B

THE FUNCTION PLOT WILL GRAPH ONE OR MORE FUNCTIONS SIMULTANEOUSLY, AUTOMATICALLY SCALING THE VALUES TO FIT APPROXIMATELY WITHIN SCALE DIMENSIONS SPECIFIED BY THE USER. IT WILL WORK ONLY IN 1-ORIGIN INDEXING.

THE FORM IN WHICH PLOT IS USED IS:

SCALESIZE PLOT FUNCTION

LEFT ARGUMENT: ONE OR TWO NUMBERS.
SCALESIZE[1]: HEIGHT OF PLOT ON PAPER IN LINES
SCALESIZE[2]: WIDTH OF PLOT ON PAPER IN CHARACTERS

THERE IS NO BUILT-IN LIMIT TO THE DIMENSIONS, AND A HORIZONTAL AXIS LARGER THAN THE WORKSPACE WIDTH WILL CAUSE SOME POINTS TO BE PRINTED ON THE NEXT LOWER LINE.

RIGHT ARGUMENT: A RECTANGULAR ARRAY WITH RANK ≤ 3.

SCALAR: WILL BE TREATED AS A VECTOR OF LENGTH ONE.
VECTOR: WILL BE PLOTTED AS ORDI NATE AGAINST ITS OWN INDICES AS ABSCISSA.
MATRIX: THE LEFTMOST COLUMN WILL BE TAKEN AS THE ABSCISSA AND ALL OTHER COLUMNS WILL BE PLOTTED AS ORDINATES. A DIFFERENT PLOTTING SYMBOL UP TO THE NUMBER OF SYMBOLS AVAILABLE WILL BE USED FOR EACH COLUMN. IN CASE TWO ORDINATES HAVE A COMMON POINT, THE SYMBOL FOR THE COLUMN FURTHEST TO THE RIGHT WILL BE USED.

3-DIMENSIONAL ARRAY: THE FIRST PLANE WILL BE PLOTTED AS A MATRIX, AND ALL OTHER PLANES WILL BE DISREGARDED.

AUXILIARY FUNCTIONS: THE FUNCTIONS AND AND VS CAN BE USED TO GENERATE THE RIGHT ARGUMENT IN THE PROPER FORM FOR PLOT. FOR EXAMPLE:

20 PLOT Z AND Y VS X
PLOT CHARACTERS: THE SYMBOLS USED ARE ASSIGNED TO THE VARIABLE PC IN LINE 1 OF PLOT. THE ALPHABET SUPPLIED IS 'O*oV60'. THIS ALPHABET MAY BE EXTENDED AND MODIFIED AS DESIRED, USING THE NORMAL FUNCTION-EDITING PROCEDURES: EITHER CHANGE LINE 1 OF THE FUNCTION, OR DELETE IT AND INDEPENDENTLY SPECIFY A VALUE FOR PC.

HISTOGRAMS: PLOT CAN BE USED TO GENERATE HISTOGRAMS BY SETTING THE VARIABLE HS TO 1 IN LINE 2 OF THE FUNCTION. ALTERNATIVELY, LINE 2 CAN BE DELETED, AND HS CAN BE SET EXTERNALLY.
--- 1 PLOTFORMAT VS ---

**SYNTAX:** 2*A VS B

The `<VS>` function is essentially a column-catenator, similar to AND, except that the right-hand argument must be of rank ≤ 1. It is designed primarily to provide convenient formation of input to PLOT function. Whether used by itself or with AND, VS will cause its right argument to appear as the leftmost column of the resulting array. (The resultant will be an array of rank three, consisting of a single plane).

```
V M=4 VS B; C; D
[1] +((1<ppB), B<ppB), 2 1 0 <ppA)/ 8 8 4 3
[2] A+((pB),1)pA
[3] A+((X/pA),1)pA
[4] +(A/(pB),1,1pA)/9
A
[7] *1<pM=1,pM)pM
[8] *1<1<AN ARGUMENT OF VS IS OF IMPROPER RANK.'
[9] 'ARGUMENTS OF VS ARE NOT CONFORMABLE.'
```

--- 1 PLOTFORMAT AND ---

**SYNTAX:** 2*A AND B

The `<AND>` function is essentially a column-catenator, with some extra effects when the arguments are not matrices. The function is designed to be used either independently, or in conjunction with VS. It provides a convenient way of forming input to DFT and EFT.

```
V AND[1;1;1]
V L=4 AND B; C; D
[1] +((2<ppA)v3<ppB), 0<ppB)/ 17 3
[2] B+ B
[3] +((3<ppA)v1<ppB), 2<ppA)/ 17 7
[4] A+ A
[5] +(A/(pA),1/z1pA),1pA)
[6] A+((D*pA),1)pA
[7] +(1<pB)/9
[8] B+((pB),(1<pB)x1pA),1pB
[9] *(A/B1,1pB),1pB/16 11
[10] B+((3<ppB)p1),(1ppA),1pB)
[12] L+((C�1pB)<p0), (1pB)p1)
[13] +0<ppL[1;1;1]+A
[14] L+((1+(C1pB)<p0), (1+1pB)p1)/E
[15] +0<ppL[1;1;1+1C]+A
[16] *D=0<D<AN ARGUMENT OF AND ARE NOT CONFORMABLE.'
[17] 'AN ARGUMENT OF AND IS OF IMPROPER RANK.'
```

190
SYNTAX: FS δFMT (V1;V2;V3;VN)

AUTHOR: STSC

REFERENCE: δFMT - REPORT FORMATTING IN APL PLUS, STSC

δFMT gives the APL PLUS user a simple, economical, and adaptable means of producing reports involving tabular formatting of numeric and character data. Tabular output produced using δFMT takes about one-twentieth the CPU time required using other methods. δFMT employs its left argument to control printing of its right in one or more columns.

RIGHT ARGUMENT

V1, V2, V3, ..., and VN are APL variables or expressions. They may be any combination of character or numeric scalars, vectors or matrices. A matrix is printed in one or more columns; a vector or scalar is treated as though it were a single column matrix.

LEFT ARGUMENT

FS is a character vector made up of one or more format phrases separated by commas. Each phrase controls the printing of one or more columns. The allowable format phrases are:

- MAW CHARACTER EDITING
- MW OR MQIW INTEGER EDITING
- MFW.D OR MQFW.D FIXED-POINT NUMERIC EDITING
- MFW.S FLOATING-POINT NUMERIC EDITING
- MXW INSERTION OF BLANKS
- MTEXT INSERTION OF TEXT

Here M, W, D, and S represent integer constants. O represents optional qualifiers and decorations.

M, which is optional, indicates how many times a phrase is to apply. If M is not present, it is taken to be 1. Thus, 3AI is equivalent to A1,A1,A1 and 214 is equivalent to 14,14.

W is the field is the field width, indicating the number of print positions occupied by the edited value (for A, I, F, and E) or by inserted blanks (for X).

For fixed-point editing, D is the number of digits to appear to the right of the decimal point. D must be less than W. For floating-point editing, S is the number of significant digits to be printed. S must be less than W - 4.
THE A PHRASE IS USED ONLY TO EDIT CHARACTER VALUES WHILE I, E, AND F PHRASES ARE USED ONLY TO EDIT NUMERIC VALUES.

CHARACTERS ENCLOSED BETWEEN ~ SYMBOLS ARE INSERTED DIRECTLY INTO THE EDITED LINE. THE INSERTED TEXT IS PRINTED EXACTLY AS IT APPEARS (INCLUDING SPACES) BETWEEN THE ~ SYMBOLS.

QUALIFIER CODES

B LEAVES THE RESULT FIELD BLANK IF THE EDITED VALUE IS ZERO.
C INSERTS COMMAS BETWEEN EACH GROUP OF THREE DIGITS IN THE INTEGER PART OF THE EDITED VALUE.
L LEFT-ADJUSTS THE VALUE IN THE RESULT FIELD.
Z FILLS UNUSED LEADING POSITIONS IN THE RESULT FIELD WITH ZEROS (AND INSERTED COMMAS, IF THE C QUALIFIER IS USED) INSTEAD OF BLANKS.

DECORATION CODES

M[TXT~ PLACES THE TEXT ON THE LEFT OF AN EDITED NEGATIVE VALUE. THE DEFAULT VALUE FOR THE M DECORATIONS IS THE NEGATIVE SIGN, "-".
N[TXT~ PLACES THE TEXT ON THE RIGHT OF AN EDITED NEGATIVE VALUE. THE DEFAULT VALUE IS NULL.
P[TXT~ PLACES THE TEXT ON THE LEFT OF AN EDITED POSITIVE OR ZERO VALUE. THE DEFAULT VALUE IS NULL.
Q[TXT~ PLACES THE TEXT ON THE RIGHT OF AN EDITED POSITIVE OR ZERO VALUE. THE DEFAULT VALUE IS NULL.
R[TXT~ FILLS THE RESULT FIELD WITH THE TEXT, PRIOR TO EDITING THE VALUE INTO THE FIELD. THE TEXT WILL APPEAR IN ALL POSITIONS OF THE RESULT FIELD NOT FILLED BY THE EDITED VALUE. THE TEXT IS USED AS MANY TIMES (REPLICATED) AS NEEDED TO FILL THE FIELD. THE DEFAULT VALUE IS BLANK.
FORMAT COLNAMES

SYNTAX: RESULT + FS COLNAMES TEXTVECTOR

REQUIRES: EWTD

REFERENCE: Afmt - REPORT FORMATTING IN APL PLUS, STSC


```apl
∇COLNAMES[[]]
∇ R+·A COLNAMES B;C;D;E;S
[1] C+5·0pA+((E+3T1)p0);,RWTD A
[2] +(12p4)p12
[5] 4+4A
[6] >2
[7] C+·C,((0.5+4x(A[3]))+x)/A[1 2]
[8] +(12p4+4A)p12
[10] C+((-1+C),(-1+C)+x)/A[1 2]
[12] A++//LC
[13] >S·0,E+(B=−1B)/1pB+·Φ,B
[14] +(1 0.5 =pS),11+C)/ 22 19
[15] R+R,((1+C)+((0D)>11+C)/(1+C)p**'),(1+C)+D+−1+S[1-F]+S[2-F]+B
[16] S+1+S
[17] C=1+C
[18] >14
[19] R+R,(11+C)**
[20] C=1+C
[21] >14
[22] R+(1,4)pA+R
∇
```

193
FUNCTION CENTER produces a centered title line for use with a "FMT display. FS is the vector of format phrases normally used by "FMT, and the TEXTVECTOR is a character vector holding the title. The result is a character matrix with one row and as many columns as a "FMT result would have if used with FS as the left argument. The characters given in TEXTVECTOR are centered in this matrix. If TEXTVECTOR is too long, it is truncated at the right.

```

```
SYNTAX: R+AKI P

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

WHEN EXECUTED, <AKI> PRINTS OUT THE PROMPT P AND AWAIT A USER RESPONSE ON THE SAME LINE. CONSECUTIVE SPACES ARE COMPRESSED OUT OF THE RESPONSE, WHICH IS THEN RETURNED AS THE EXPLICIT RESULT OF THE FUNCTION.

```plaintext
    R+AKI AKI;A
[1]  R+AKI
[2]  201;0
[4]  R' ' =A' ' ,A
[5]  R+1+(R*1+R,1)/A
[6]  51210.01-140*2*(pA)+-1+pAKI
[8]  'x' SUSPEND 'v, xA
[9]  '+(STOP',v,XA)pA
[10]  
```

SYNTAX: R+AYN P

REQUIRES:AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYN> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO RESPONSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYN> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYN> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.

```plaintext
    R+AYN AKI;A;N;T
[1]  * ACCEPT YES/NO INPUT. REQUIRES <AKI>, AND <ES>. AUTHOR: K. SYKES, STSC.
[2]  A+ 4 3 p"YES\NO Y N'
[3]  T+AYN
[4]  *(+0 3 =pN+AKI T,'YES\NO Y N',pBS)/ 3 6
[5]  'b',0T='PLEASE ANSWER YES OR NO:
[6]  +(v/T=AKI."YES\NO Y N")+5
[7]  R+2/T
```

--- 1 INFNS AYN ---

SYNTAX: R+AYN P

REQUIRES:AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYN> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO RESPONSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYN> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYN> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.

```plaintext
    R+AYN AKI;A;N;T
[1]  * ACCEPT YES/NO INPUT. REQUIRES <AKI>, AND <ES>. AUTHOR: K. SYKES, STSC.
[2]  A+ 4 3 p"YES\NO Y N'
[3]  T+AYN
[4]  *(+0 3 =pN+AKI T,'YES\NO Y N',pBS)/ 3 6
[5]  'b',0T='PLEASE ANSWER YES OR NO:
[6]  +(v/T=AKI."YES\NO Y N")+5
[7]  R+2/T
```

--- 2 INFNS AYK ---

SYNTAX: R+AYK P

REQUIRES:AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYK> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO RESPONSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYK> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYK> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.

```plaintext
    R+AYK AKI;A;N;T
[1]  * ACCEPT YES/NO INPUT. REQUIRES <AKI>, AND <ES>. AUTHOR: K. SYKES, STSC.
[2]  A+ 4 3 p"YES\NO Y N'
[3]  T+AYK
[4]  *(+0 3 =pN+AKI T,'YES\NO Y N',pBS)/ 3 6
[5]  'b',0T='PLEASE ANSWER YES OR NO:
[6]  +(v/T=AKI."YES\NO Y N")+5
[7]  R+2/T
```

--- 3 INFNS AYK ---

SYNTAX: R+AYK P

REQUIRES:AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYK> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO RESPONSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYK> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYK> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.

```plaintext
    R+AYK AKI;A;N;T
[1]  * ACCEPT YES/NO INPUT. REQUIRES <AKI>, AND <ES>. AUTHOR: K. SYKES, STSC.
[2]  A+ 4 3 p"YES\NO Y N'
[3]  T+AYK
[4]  *(+0 3 =pN+AKI T,'YES\NO Y N',pBS)/ 3 6
[5]  'b',0T='PLEASE ANSWER YES OR NO:
[6]  +(v/T=AKI."YES\NO Y N")+5
[7]  R+2/T
```

--- 4 INFNS AYK ---

SYNTAX: R+AYK P

REQUIRES:AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYK> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO RESPONSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYK> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYK> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.

```plaintext
    R+AYK AKI;A;N;T
[1]  * ACCEPT YES/NO INPUT. REQUIRES <AKI>, AND <ES>. AUTHOR: K. SYKES, STSC.
[2]  A+ 4 3 p"YES\NO Y N'
[3]  T+AYK
[4]  *(+0 3 =pN+AKI T,'YES\NO Y N',pBS)/ 3 6
[5]  'b',0T='PLEASE ANSWER YES OR NO:
[6]  +(v/T=AKI."YES\NO Y N")+5
[7]  R+2/T
```

--- 5 INFNS AYK ---

SYNTAX: R+AYK P

REQUIRES:AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYK> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO RESPONSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYK> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYK> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.
INFNS INP

SYNTAX: R+INP L

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <INP> ALLOWS THE USER TO SPECIFY THE ACCEPTABLE
NUMBER OF INPUTS. THE RIGHT ARGUMENT IS ONE OR MORE NUMBERS WHICH
REPRESENT THE ACCEPTABLE SIZE OF THE INPUT VECTOR. FOR INSTANCE, IF
YOU WOULD ACCEPT EITHER 1 OR 5 NUMERIC INPUTS, WHERE IN THE CASE
WHERE 1 WAS GIVEN IT WOULD BE USED FOR ALL FIVE NUMBERS, THE
FOLLOWING FORM WOULD BE USED:

R+INP 1 5

THE <INP> FUNCTION WOULD ACCEPT A SERIES OF NUMBERS AND CHECK
TO SEE IF THERE WAS EITHER 1 OR 5 VALUES GIVEN. IF THERE WAS, THE
RESULT WOULD BE RETURNED TO THE CALLING PROGRAM. IF THERE WAS A
LARGER OR SMALLER GROUP OF NUMBERS GIVEN, AN APPROPRIATE ERROR
MESSAGE WOULD BE GIVEN.

INFNS IPI

SYNTAX: R+IPI P

REQUIRES: AKI MIM

AUTHOR: CHRIS CLAUSEN, PHYSIOLOGY DEPARTMENT, UCLA

THE FUNCTION <IPI> ALLOWS THE MAIN PROGRAM TO PROMPT THE USER
FOR A SERIES OF POSITIVE INTEGERS. THE MAIN PROGRAM PRINTS THE
RIGHT ARGUMENT, WHICH IS AN ALPHA NUMERIC LITERAL VECTOR (PROMPT)
EXPLAINING TO THE USER WHAT NUMBERS HE SHOULD ENTER. THE FUNCTION
THEN WILL ACCEPT THE RESPONSE FROM THE USER, TYPED ON THE SAME LINE
AS THE PROMPT, AND RETURN THE RESULT TO THE MAIN PROGRAM. IF
ANYTHING OTHER THAN POSITIVE INTEGERS ARE ENTERED AN APPROPRIATE
ERROR MESSAGE WILL BE GIVEN. THIS PROGRAM WILL ACCEPT EITHER 1
VALUE, WHICH IS CONVERTED INTO A SCALAR; OR A SERIES OF VALUES,
WHICH ARE CONVERTED INTO A VECTOR.
SYNTAX: R+LINP S

AUTHOR: CHRIS CLAUSEN, PHYSIOLOGY DEPARTMENT, UCLA

THE FUNCTION <LINP> IS DESIGNED TO ACCEPT INPUT THAT WOULD
NORMALY NOT FIT ON ONE LINE. THE MAIN PROGRAM SPECIFIES THE NUMBER
OF INPUTS DESIRED, N, AND THE <LINP> CONTINUES TO ASK FOR MORE DATA
UNTIL IT HAS RECEIVED THE REQUIRED NUMBER. IF THE USER TYPES IN
MORE NUMBERS THAN ARE NEEDED, THE FUNCTION WILL ASK THE USER TO
REENTER JUST THE LAST LINE OF HIS INPUTS.
SYNTAX: R+NIP P
REQUIRES: AKI LF
AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <NIP> ACCEPTS A PROMPTED NUMERIC ANSWER AND
RETURNS THE RESULT TO THE CALLING PROGRAM. THE RIGHT ARGUMENT, P,
IS A ALPHANUMERIC LITERAL DEFINING TO THE USER WHAT INFORMATION IS
DESIRED. THIS FUNCTION WILL ACCEPT JUST ONE NUMBER, BUT THAT NUMBER
CAN BE ANY VALID REPRESENTATION ACKNOWLEDGEABLE IN APL.

V R+NIP P; 0; S; I; D; E; T
1. * NUMERIC INPUT; AUTHOR: ROY SYKES, STSC.
2. * O, T -> 71
4. E+(1+I+O F1 .m')+ P+, P
5. P+T+P
6. *(O 1 =pT+AKI P)/ 6 9
7. T+O+ '0123456789' 'E' T
8. *(13t=(S-((P=1)+T) E 11 12)+T) P 15 11
9. *(11 10 0 <=T+O+ '0123456789' 'T') E 10 20
10. *O,P+, '
11. P+(1+D+O+T) 13+T
12. *(V/10sF+(I=11=1+F)+P) pE
13. *(Oe(pP), pT+D+T) pE
14. P-10*(-1*I)X10T
15. *(S/10D+(1-I+O+T) 10+T) pE
16. *(0=pT+(I+T), D) pE
17. T-P+(1 0 .1 X.S, pD)x101T
18. *19+T*3T
19. T+L
20. R+T

198
THE BS CHARACTER CAN BE USED TO BACKSPACE THE PRINT ELEMENT. IT CAN BE STORED LIKE ANY OTHER APL CHARACTER IN A CHARACTER ARRAY. AN EXAMPLE OF ITS USE IS:

R←'THIS IS A TEST', (14p BS), 14p '_'

R NOW LOOKS LIKE THIS:

THIS IS A TEST

THE CR CHARACTER CAN BE USED TO CAUSE A CARRIAGE RETURN WITHIN A PROGRAM. IT CAN BE STORED LIKE ANY OTHER APL CHARACTER. FOR EXAMPLE IT CAN BE IMBEDDED WITHIN A CHARACTER ARRAY. FOR EXAMPLE:

R←'THIS', CR, 'IS', CR, 'A', CR, 'TEST'

R IS NOW THIS:

THIS IS A TEST

THE LINE FEED CHARACTER CAN BE USED TO CAUSE THE TERMINAL TO SKIP LINES WITHOUT RETURNING THE CARRIAGE. FOR INSTANCE, IF THE PRINT BALL WAS POSITIONED AT POSITION 50 ON THE PAGE, 4 LF'S WOULD CAUSE THE TERMINAL TO ADVANCE 4 LINES, BUT THE TYPE BALL WOULD STILL BE IN POSITION 50.
5.0

OUTLINE OF THE COURSE
5.1

Class Schedule and Reading Assignments
# CLASS SCHEDULE

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Group A</th>
<th>Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday, 30 June</td>
<td>9:00 - 10:00 a.m.</td>
<td>Course Introduction and Overview: Administrative Details of the Program (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10:00 - 11:00 a.m.</td>
<td>Tour of the UCLA Campus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11:00 - 11:30 a.m.</td>
<td>Tour of UCLA Campus Computing Network (CCN)</td>
<td>Visit Ackerman and Kerckhoff Halls</td>
</tr>
<tr>
<td></td>
<td>11:30 - 12:00 a.m.</td>
<td>Visit Ackerman and Kerckhoff Halls</td>
<td>Tour of UCLA Campus Computing Network (CCN)</td>
</tr>
<tr>
<td></td>
<td>12:00 - 1:00 p.m.</td>
<td>Lunch, UCLA Faculty Center</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1:00 - 2:30 p.m.</td>
<td>Assignment #1, Lecture (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2:30 - 4:00 p.m.</td>
<td>Assign. #2, Lecture (2232A GSM)</td>
<td>Assign. #1, Lab. (2035C GSM)</td>
</tr>
<tr>
<td></td>
<td>4:00 - 5:30 p.m.</td>
<td>Assign. #1, Lab. (2035C GSM)</td>
<td>Assign. #3, Lecture (2232A GSM)</td>
</tr>
<tr>
<td>Tuesday, 1 July</td>
<td>8:30 - 10:00 a.m.</td>
<td>Assign. #3, Lecture (2232A GSM)</td>
<td>Assign. #3, Lab. (2035C GSM)</td>
</tr>
<tr>
<td></td>
<td>10:00 - 10:15 a.m.</td>
<td>Break</td>
<td>Break</td>
</tr>
<tr>
<td></td>
<td>10:15 - 11:45 a.m.</td>
<td>Assign. #3, Lab. (2035C GSM)</td>
<td>Assign. #4, Lecture (2232A GSM)</td>
</tr>
<tr>
<td></td>
<td>11:45 - 12:50 p.m.</td>
<td>Lunch, UCLA Faculty Center</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12:50 - 2:20 p.m.</td>
<td>Assign. #4, Lecture (2232A GSM)</td>
<td>Assign. #4, Lab. (2035C GSM)</td>
</tr>
<tr>
<td></td>
<td>2:20 - 2:25 p.m.</td>
<td>Break</td>
<td>Break</td>
</tr>
<tr>
<td>Time</td>
<td>Activity</td>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>3:55 - 4:00 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:00 - 5:30 p.m.</td>
<td>Assign. #10, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab. (2035C GSM)</td>
<td></td>
</tr>
<tr>
<td>8:30 - 10:00 a.m.</td>
<td>Assign. #10, Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lecture (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td>10:00 - 10:15</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:15 - 11:45 a.m.</td>
<td>Assign. #11, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab. (2035C GSM)</td>
<td></td>
</tr>
<tr>
<td>11:45 - 12:50 p.m.</td>
<td>Lunch, Buffet in 2355 GSM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:50 - 2:20 p.m.</td>
<td>Assign. #11, Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lecture (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td>2:20 - 2:25 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:25 - 3:55 p.m.</td>
<td>Assign. #12, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab. (2035C GSM)</td>
<td></td>
</tr>
<tr>
<td>3:55 - 4:00 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:00 - 5:30 p.m.</td>
<td>Assign. #12, Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lecture (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td>Monday, 7 July</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30 - 10:00 a.m.</td>
<td>Assign. #13, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab. (2035C GSM)</td>
<td></td>
</tr>
<tr>
<td>10:00 - 10:15 a.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:15 - 11:45 a.m.</td>
<td>Assign. #13, Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lecture (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td>11:45 - 12:50 p.m.</td>
<td>Lunch, UCLA Faculty Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:50 - 2:20 p.m.</td>
<td>Assign. #14, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab. (2035C GSM)</td>
<td></td>
</tr>
<tr>
<td>2:20 - 2:25 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:25 - 3:55 p.m.</td>
<td>Assign. #14, Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lecture (2232A GSM)</td>
<td></td>
</tr>
<tr>
<td>3:55 - 4:00 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab. (2035C GSM)</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tuesday, 8 July</strong></td>
<td>Assign. #15, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:00 - 5:30 p.m.</td>
<td>Assign. #15, Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30 - 10:00 a.m.</td>
<td>Assign. #16, Lab. (2035C GSM) Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00 - 10:15 a.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:15 - 11:45 a.m.</td>
<td>Assign. #17, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:45 - 12:50 p.m.</td>
<td>Lunch, UCLA Faculty Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:50 - 2:20 p.m.</td>
<td>Assign. #18, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:20 - 2:25 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:25 - 3:55 p.m.</td>
<td>Assign. #16, Programming Project Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:55 - 4:00 p.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:00 - 5:30 p.m.</td>
<td>Programming Project Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Wednesday, 9 July</strong></td>
<td>Programming Project Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30 - 10:45 a.m.</td>
<td>Programming Project Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:45 - 11:00 a.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00 - 12:00 a.m.</td>
<td>Commercial Applications of APL, Scientific Time Sharing Corp., Lecture, (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00 - 1:00 p.m.</td>
<td>Lunch, UCLA Faculty Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:00 - 5:30 p.m.</td>
<td>Programming Project Lab (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Thursday, 10 July</strong></td>
<td>Programming Project Lab (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30 - 10:45 a.m.</td>
<td>Programming Project Lab (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:45 - 11:00 a.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00 - 12:00 a.m.</td>
<td>Commercial Applications of APL, Proprietary Computer Systems, Inc., Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00 - 1:00 p.m.</td>
<td>Lunch, UCLA Faculty Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:00 - 5:30 p.m.</td>
<td>Programming Project Lab. (2035C GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Friday, 11 July</strong></td>
<td>A Review of APL Usage, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30 - 10:00 a.m.</td>
<td>Pedagogical Approaches to Teaching APL, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00 - 10:15 a.m.</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:15 - 11:50 a.m.</td>
<td>Pedagogical Approaches to Teaching APL, Lecture (2232A GSM)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
11:00 - 12:00 a.m.  Programming Project, Lab. (2035C GSM)
12:00 - 1:00 p.m.  Lunch, UCLA Faculty Center
1:00 - 5:30 p.m.  Programming Project, Lab. (2035C GSM)

6/30/75
INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

Reading Assignments


<table>
<thead>
<tr>
<th>Assign. #</th>
<th>Required Reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Buckley et al., Chapter 1. G. &amp; R., Chapters 1 &amp; 2.</td>
</tr>
<tr>
<td>8.</td>
<td>G. &amp; R., pp. 131-137, 159-164, Chapters 22 &amp; 23</td>
</tr>
</tbody>
</table>
14. UCLA APL Library Description (Handout); PLOT Description (Handout); Buckley et al., Chapter 7.

15. STSC, APL Plus FMT Report Formatting Instructions Manual; G. & R., Chapter 21 (lightly)

16. No reading assignment (a discussion of programming techniques for use with APL).


6/30/75
5.2

Class Handouts
5.2.1

A Guide to Writing APL Programs
A GUIDE TO WRITING APL PROGRAMS

1. Define the Problem.
   Read the problem statement carefully and decide what it is that is really being asked for.

2. Identify the Steps Needed.
   Break down the problem into the logical steps needed to solve the problem (not the specific APL instructions).

3. Define the Logic of Each Step.
   Here the exact operations needed to accomplish the desired results are defined; e.g., "Input data from a user"--a step as defined in No. 2 above--would require (1) "Display instructions to the users of your program," (2) "Accept a literal input from the user, and (3) "Store the input in a variable," would be the No. 3 equivalent of No. 2.

4. Write the Code and Define the Function.
   Taking the detailed logic of Step No. 3, write the actual APL code needed to accomplish each step or task. If you are unsure about a particular operator, or how a line of code will actually execute, you may wish to test it out first before going into function definition mode.

5. When You Have Defined the Function, Test It with Sample Data.
   In testing your function, use "typical" data first, and then use more unusual data (e.g., vectors rather than scalars, negative numbers, zeros, etc.).

6. When the Function "Hangs," Examine the Indicators.
   The error messages (e.g., LENGTH ERROR, RANK ERROR, SYNTAX ERROR, etc.) and the location of the caret (the "^") can be very helpful in determining what is wrong.

7. Examine the Line of Code at Which Execution Stopped.
   It is possible that the problem was caused by a mistake in a preceding line of code (but not of the sort that would produce an error message), but you should first check out the line where execution stopped. In analyzing a line, the following procedure is a helpful guide:
START AT THE RIGHT AND WORK TO THE LEFT (the same way APL executes!)

Ask the following questions to yourself of each term encountered:

1) Is it an argument or an operator?
2) If an argument, is it a matrix, vector, or scalar?
3) If an operator, is it monadic or dyadic? (To tell, look to see if there is a left argument.)
4. If it is dyadic, is the left argument a matrix, vector, or scalar and is it compatible with the right argument?

Once the above have been determined, execute (in desk calculator mode) the portion of the line of code just analyzed.

Using the result of the preceding execution, proceed again to the left until the entire line of code has been checked out and produces the desired result. (N.B., the first error you find and correct may not be the only error in the line.)

Two of the most common errors are to overlook the strict right to left execution (which can frequently be corrected by the use of parentheses) and the failure to recognize that vector operations are involved (e.g., thus creating LENGTH, RANK, or INDEX errors). This latter problem may require a variety of solutions (e.g., reduction, ravelling, etc.) as determined by each situation.

8. Clear the Suspend Execution.

Although it is possible to continue the execution of the suspended program (by branching to the point in the program where the execution was suspended), it is probably easier to terminate execution and start again. You can clear out the suspended execution simply by typing a right arrow ("->"). (To check to see if all suspensions have been cleared, type ")SI". If there are none, nothing will print; if some still exist, one line will print for each suspension stored. As above, one right arrow is needed to clear each suspension.)

9. Correct the Error.

By use of function editing, go into function definition mode and change the incorrect line of code into its corrected version.

10. Retest the Function.

Quite frequently, another error will be detected, so Steps No. 6, 7, and 8 should be repeated.

11. When No Errors are Detected, "Desk Check" Your Results.

Even though no errors have been detected by the system, it does not necessarily mean that the results are what are desired. Therefore, manually calculate the answer for a few test data conditions to insure that the function is working as planned.
12. (Optional) When Correct, See If the Function Can Be Shortened.

Because APL executes in an interpretive mode, the more lines of code there are, the longer execution will take. Therefore—if you have time and/or the function is likely to be used frequently—investigate ways whereby one or more lines of code can be combined into a single line.

P.S. Don't forget to save a copy of your now-correct function in your stored workspace.
5.2.2

Signing-On to the UCLA APL System
<table>
<thead>
<tr>
<th>NAME</th>
<th>____________________________</th>
</tr>
</thead>
<tbody>
<tr>
<td>APL NUMBER</td>
<td>_______________</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>_______________</td>
</tr>
<tr>
<td>NEW PASSWORD</td>
<td>_______________</td>
</tr>
</tbody>
</table>
SIGNING-ON

1. TURN THE TERMINAL ON BY FLIPPING THE ON-OFF SWITCH ON THE LEFT OF THE KEYBOARD.


3. TYPE 'x' (UPPER CASE P) FOLLOWED BY 'a' (UPPER CASE A) FOLLOWED BY ')' AND PRESS 'RETURN'; EXAMPLE *a) (CARRIAGE RETURN). THE COMPUTER WILL RESPOND WITH THE MESSAGE 'INCORRECT SIGN-ON' AND PRINT A SHORT LINE OF OVERTYPED CHARACTERS TO OBSCURE THE NEXT TYPED ENTRY.

4. TYPE YOUR APL NUMBER AND YOUR PASSWORD; EXAMPLE 1234:MCLEAN (DO NOT FORGET THE RIGHT PARENTHESES BEFORE YOUR APL NUMBER AND THE COLON BEFORE YOUR PASSWORD).

ERROR CORRECTION

1. STRIKE THE BACKSPACE KEY TO MOVE THE CARRIAGE OVER TO WHERE THE ERROR BEGINS.

2. HIT THE INDEX KEY ON THE UPPER RIGHT SIDE OF THE KEYBOARD. AN INVERTED CARET (^) WILL APPEAR UNDER THE CHARACTER AT THAT POINT. EVERYTHING ABOVE AND TO THE RIGHT OF THE CARET IS WIPED OUT FROM THE MEMORY OF THE SYSTEM.

3. TYPE THE CORRECTION AND THE REMAINDER OF THE LINE.

SIGNING-OFF

1. TYPE )OFF OR )CONTINUE AND PRESS 'RETURN'.

2. TURN OFF THE TERMINAL.

3. PLACE THE PHONE HEADSET BACK IN THE CRADLE.
SIGNING-ON

1. TURN THE TERMINAL ON BY FLIPPING THE ON-OFF SWITCH ON THE RIGHT OF THE KEYBOARD.

2. CHECK THE SMALL SWITCH LOCATED IN THE FRONT LEFT CORNER, UNDER THE KEYBOARD, AND SWITCH IT TO THE RIGHT.


4. TYPE 'A' FOLLOWED BY '9' FOLLOWED BY ')' AND PRESS 'RETURN'; EXAMPLE: A9) (CARRIAGE RETURN). THE COMPUTER Responds WITH THE MESSAGE, 'Incorrect sign-on' AND PRINTS A SHORT LINE OF OVERSTRUCK CHARACTERS TO OBSERVE THE NEXT TYPED ENTRY.

5. TYPE YOUR APL NUMBER AND YOUR PASSWORD; EXAMPLE: )1234:Mclean (DO NOT FORGET THE RIGHT PARENTHESIS BEFORE YOUR APL NUMBER AND THE COLON BEFORE YOUR PASSWORD).

ERROR-CORRECTION

1. STRIKE THE BACKSPACE KEY TO MOVE OVER TO WHERE THE ERROR BEGINS.

2. HIT THE ATTENTION KEY (ATTN) IN THE UPPER RIGHT HAND CORNER. AN INVERTED CARET (↑) WILL APPEAR UNDER THE CHARACTER AT THAT POINT. EVERYTHING ABOVE AND TO THE RIGHT OF THE CARET IS WIPED OUT FROM THE MEMORY OF THE SYSTEM.

3. TYPE THE CORRECTION AND THE REMAINDER OF THE LINE.

SIGNING-OFF

1. TYPE: )OFF OR )CONTINUE AND PRESS 'RETURN'.

2. TURN OFF THE TERMINAL AND THE ACOUSTIC COUPLER.

3. REMOVE THE TELEPHONE HEADSET FROM THE COUPLER AND HANG UP.
SIGNING-ON

1. TURN THE TERMINAL ON BY FLIPPING THE ON-OFF SWITCH ON THE RIGHT OF THE KEYBOARD.

2. CHECK THE SWITCH ON THE LEFT SIDE OF THE TERMINAL. IT SHOULD BE SET TO 'ON NOT LCL'.


4. TYPE 'A' FOLLOWED BY ']' FOLLOWED BY ')' AND PRESS 'RETURN'; EXAMPLE: A]]) (CAPTAGE RETURN). THE COMPUTER RESPONDS WITH THE MESSAGE, 'INCORRECT SIGN-ON' AND PRINTS A SHORT LINE OF OVERSTRUCK CHARACTERS TO OBSCURE THE NEXT TYPED ENTRY.

5. TYPE YOUR APL NUMBER AND YOUR PASSWORD; EXAMPLE: )1234:MCLEAN (DO NOT FORGET THE RIGHT PARENTHESIS BEFORE YOUR APL NUMBER AND THE COLON BEFORE YOUR PASSWORD).

ERROR CORRECTION

1. STRIKE THE BACKSPACE KEY TO MOVE OVER TO WHERE THE ERROR BEGINS.

2. HIT THE ATTENTION KEY (ATTN) IN THE UPPER RIGHT HAND CORNER. AN INVERTED CARET (\) WILL APPEAR UNDER THE CHARACTER AT THAT POINT. EVERYTHING ABOVE AND TO THE RIGHT OF THE CARET IS WIPED OUT FROM THE MEMORY OF THE SYSTEM.

3. TYPE THE CORRECTION AND THE REMAINDER OF THE LINE.

SIGNING-OFF

1. TYPE: )OFF OR )CONTINUE AND PRESS 'RETURN'.

2. TURN OFF THE TERMINAL AND THE ACOUSTIC COUPLER.

3. REMOVE THE TELEPHONE HEADSET FROM THE COUPLER AND HANG UP.
5.2.3
An APL Bibliography

**APL: The Language and Its Use**
R.P. Polivka and S. Pakin
BIBLIOGRAPHY

This APL Bibliography is organized in sections as follows:

General: Articles that tell about APL or that describe it generally.

Texts: Books for learning APL.

Applications: Books and articles discussing uses of APL and applications in education, business, engineering and science, computers and computing, mathematics and statistics, and miscellaneous.

APL developments: Articles discussing extensions to APL and/or the APL terminal system.

Miscellaneous: Articles of interest to the APL programmer.

Collections: APL journals and conference proceedings.

The bibliography is rich in source materials with the following exceptions. No books or articles written prior to 1968 are included. 1968 was chosen as a cutoff date because APL/360 was not made publicly available by IBM until then. Earlier references tend to use Iverson notation (as formulated in A Programming Language, Wiley, New York, 1962) rather than APL. Neither articles nor manuals discussing specific implementations of APL on various machines or by various companies or universities nor how APL is used at various installations are included since such information tends to be transient. No foreign language articles are included. Finally no articles whose distribution is restricted are included.
Bibliography

General: Articles that Tell About APL—Describe or Discuss It Generally


Texts: Teaching APL


Bibliography


Education: Uses of APL and Applications in


Bibliography


Bibliography


Bibliography


Business: Uses of APL and Applications in


Bibliography


Bibliography


*Engineering and Science: Applications and Uses of APL in*


Bibliography


Computers and Computing: Uses and Applications of APL in


Bibliography


Bibliography


Bibliography


Bibliography


**Mathematics and Statistics: Applications and Uses of APL in**


Bibliography


Bibliography


Bibliography


Smith, R., "Interpretations of All Possible Reductions of a Boolean Vector Where the Result is 0 or 1" and "Similarities Between Reduction and Scan of Boolean Vectors," APL Quote-Quad, Jan. 1973, 34-35.


Bibliography


Other Applications and Uses of APL


Bibliography


**APL Developments**


Bibliography


Bibliography


Bibliography


Ryan, J., “Generalized Lists and Other Extensions,” APL Quote-Quad, June 1971, 8-10.


Miscellaneous: Of Interest to the APL-Programmer or APL-Related Information


Bibliography


Bibliography


Collections


*APL Quote-Quad*, an informal publication for APL-users, published quarterly.


Bibliography


*Colloque APL*, IRIA (Institut de Recherche d'Informatique et d'Automatique), Rocquencourt, France, Sept. 1971.

*Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, Coast Community College District.
5.2.4
"Things Go Better with APL"
Claire Stegmann
Think
May/June 1975
Tom Ryder, a Salt Lake City businessman, picks up his weekly payroll at a computer service bureau. He stops for a Coke. Then it’s back to his office in his Citroën for an appointment with an IBM salesman. He’s considering a computer of his own; and he’d like to have it by the time his son, just graduating from M.I.T., joins the firm.

Ryder might be surprised to learn that everything he has just done is involved, in some way, with the computer time-sharing language, APL.

The customer service bureau he uses, which offers APL service, is operated by Management Systems Corporation, a wholly owned subsidiary of the Church of Jesus Christ of Latter-day Saints. The Mormon Church, as it is more commonly known, also uses APL in many of its own operations.

- Both The Coca-Cola Company and Citroën are increasing users of APL.
- The IBM salesman, to help Ryder tailor a computer to his particular needs, will almost certainly consult a terminal in his branch office for design analysis and to find out what IBM products and services are available, and where. That terminal is tied into the Data Processing Division’s HONE (Hands-on Network) system, which makes significant use of APL.
- If Ryder decides to order, the System Products Division’s worldwide commodity planning system, based on APL, will enable SPD to provide the earliest possible delivery date.

Incidentally, Ryder’s son, back at M.I.T., uses APL to do his homework. And he and his father share another APL-based activity: the National Football League statistics they follow so closely are compiled with the language.

Ryder and his son are fictitious. But the APL applications are real. They illustrate the versatility of A Programming Language, which has come into its own as an important product for the computer user.

For most of its 13 years, APL’s primary habitat has been the classroom, where, as an extension of simple mathematical notation, it has helped high school and college students learn not only algebra, but also economics and electrical engineering. In Atlanta, students use it to design and compute the cost of making cabinets.

IBM Fellow Dr. Kenneth Iverson, who invented APL as a precise way to describe data processing, explains: “Every field of human knowledge that deals in quantity—mathematics, physics, economics—has a special vocabulary that must be put into a general-purpose language for computer use. APL defines these terms by their simple mathematical formulas, using a number of symbols, some of them (for example, ÷, –, ×, ÷) already familiar to every school child. Thus, as a student learns APL, he or she is also reinforcing knowledge of a particular discipline.”

Dr. Iverson believes that APL’s versatility as a teaching aid has barely been touched. Much of his work as a Fellow is devoted to helping educators in many fields study its application to their particular specialties.

But, according to Dr. Truman Hunter, APL manager, because APL is relatively simple to learn, versatile, and interactive (the user and the computer “converse” during the processing of a job), it is becoming more and more attractive to business for management decisions that need frequent updating and quick response.

“This has been especially true,” says Adin Falkoff, Dr. Iverson’s longtime collaborator. “Since the introduction of an APL program product called the APL Shared Variable System. This makes it efficient and convenient to work with files and data bases.” Both APLSV and APL/CMS (Conversational Monitor System), recently introduced by IBM, enable the APL user to do a greater variety of jobs on a much larger scale than before.

“It is the analytic power of APL, the language,” says Dr. Hunter, “plus the easy access to large stores of information, that I think is going to make it increasingly important to the businessman. Now, more than ever, he can
develop applications and get results while they are still timely.

APL and its conversational cousin BASIC are by far the most widely used languages specifically designed for terminal use, which allows the user to bypass such time-consuming steps as going to, and queuing up in the computer room. Of the two, BASIC is older and more widespread. But, according to Curt Bury, DPD senior product administrator, APL's inherent power is extending its effectiveness to areas beyond the scope of BASIC. An IBM operations planner agrees: "We planners have to sort out a great many complex relationships in a very short time. APL is the only game in town for this kind of thing."

IBM's accelerated use of APL since it was first introduced in the mid-Sixties, on the IBM 1130 and System/360 Model 40 and above, reflects the recognition of the language by data processing people as an effective way of getting applications written faster. "In the past several years," Data Processing Division President John F. Akers told a GUIDE user-group last year, "we have gone from 4,000 to 16,000 APL users within IBM. And the balance has shifted from scientific to commercial users."

Every IBM division relies on the language in some way, DPD's HONE system, which connects U.S. branch offices to a central data base and makes heavy use of APL programs, made a tremendous contribution to field productivity. In what is possibly the largest single use of APL, DPD's commodity planning system integrates production schedules so efficiently that decisions that used to take two months can now be made within a matter of minutes. The General Products Division depends on APL to write test cases for its new 3850 mass storage system. Many aspects of the business of IBM in Europe, according to Yves Le Borgne, advisory APL support representative in Europe, are managed from APL terminals. IBM World Trade Americas/Far East Corporation has used APL to develop applications for both its oper-
and magnetic keyhoard typewriters. OPD, incidentally, has another interest in the language. It makes the APL Selectric typing element.

In a move that brought APL into the mainstream of IBM software production, the APL group was transferred last year from OPD's Philadelphia Scientific Center into SDD. A design group, consisting of Iverson, Falkoff, Richard Lathwell, and a small staff remain in Philadelphia. The implementation group, under L. A. Morrow, joined the SDD programming center at San Jose. "We consider APL one of the primary interactive environments for current and future systems," says R. E. Ruthrauff, SDD manager of language and data facilities advanced system design.

One strong advocate of interactive computing, and of APL in particular, is Professor Ephraim McLean of U.C.L.A.'s Graduate School of Management. He introduced APL to the business school four years ago. Today, it is a requirement for all entering M.B.A. students.

This summer, with an IBM grant, Professor McLean will offer a course on interactive computing to a limited number of faculty members from other business schools. "I have been surprised at both the number and the variety of requests for applications," he comments. "They come from all over — a Jesuit priest, a woman marketing professor, a faculty member from Howard University, several individuals from both French and Mexican universities, and a surprisingly large number of business school deans and department chairmen."

Next, McLean hopes to do a study on the increased effectiveness of APL users. "If you study will find that APL is much more productive," he says. "For instance, I am convinced that with APL a programmer can accomplish in 10 instructions and a few hours what might take 200 instructions and several days in COBOL for FORTRAN."

One of the beauties of APL, says Dr. Iverson, "is that you have to learn so little to get started, like a child, or a grownup for that matter, learning a language. You tell a child two or three things and he builds his vocabulary gradually. With APL, you gradually accumulate a knowledge of the symbols as you do more and more work."

Several clerks and secretaries in town halls near Paris who probably agree. They have never seen a computer but are using terminals, and APL, to do payrolls they formerly did by hand. "Here in Europe," says Yves Le Borgne, "we are confident that APL has a great deal to offer our customers — especially now, with the economic crisis, when there is a need for better management to grasp a situation and react to a changing environment quickly."

In May, Iverson receives the annual Harry H. Goode Memorial Award from the American Federation of Information Processing Societies, for the impact APL is expected to have on programming language theory and practice in the future. The setting would be the envy of many youngsters: Disneyland, in California. But it probably wouldn't impress the son and daughter, both under nine, of IBM senior programmer Corey Fair in East Fishkill. Since the Outstanding Contribution Award winner (for his part in development of the SDD commodity planning system) brought a terminal home, his kids don't watch Walt Disney on TV much anymore. Instead, they compete for time on the terminal, when allowed, to practice APL.

Here's how it works

One of the virtues of APL is that it is interactive, that is, a person can "converse" with the computer during the processing of a job. Here is a simple example of how such a conversation might go:

Let's say a businessman wants to examine the feasibility of taking out an $8,000 loan at 9 1/2 percent interest. He wants to see what his monthly payment would be each month if he repaid the loan over an eight-year period. Also, for income tax purposes, he'd like to know what his annual interest would amount to each year.

Now, let's assume that the APL program to perform this function is identified in the computer as ME. The businessman begins by typing:

    ENTER PRINCIPAL, ANNUAL INTEREST (AS A DECIMAL), AND NO. OF YEARS

The computer responds:

    ENTER MONTH AND YEAR THAT FIRST PAYMENT IS DUE (E.G. 8 73 FOR AUG. 1973)

The businessman types:

    8 0 0 0 0 0 0 0 9 5 8

The computer comes back:

    PAYMENT AMOUNT IS 119.29

The businessman types:

    5 7 5

The computer prints out:

    END OF ANNUAL PRINCIPAL
    YEAR INT. REMAINING PRINCIPAL
    1975 494.05 7539.73
    1976 684.30 6792.55
    1977 610.15 5971.22
    1978 528.63 5068.37
    1979 439.02 4075.91

    END OF ANNUAL INTER. REMAINING INT.
    1980 340.52 2984.95
    1981 212.74 1785.71
    1982 113.23 1072.48
    1983 9.28 0.00

Think

Think is published by International Business Machines Corporation, Armonk, New York 10504. Frank T. Cary, Chairman of the Board. It is produced by IBM Corporate Communications, Jane P. Cahill, IBM vice president, communications. Think is distributed to all IBM employees in the U.S.A., and to a number of IBM employees abroad.

Chief Hansen, Editor; Woody Klein, Managing Editor; Benito P. Cianucci, Art Director; Quentin W. Johnson, Manager, Publications Production; Angelo M. Donohro, Special Features Editor; Ed Grimm, International Editor; Howard K. Janis, Senior Editor-At-Large; Geoffrey D. Austria, Emile Bauer, Editors-At-Large; Claire Siegmann, Senior Editor; Ruth S. Baker, Associate Editor; Walter W. Waszkowski, Photo Editor; Elizabeth W. Brown, Circulation. The word SelSric and all other similarly capitalized marks identify products manufactured by the IBM Corporation and distinguish them from similar products manufactured by others.

5.3

APL Programming Assignment Instructions
In applying to this program, each of you described, in varying degrees of detail, the problem or application that you wished to program in APL during the second week of the course. The resulting programs are to be published as a UCLA monograph.

In order for this aspect of the course to be successful, it is vitally important that you prepare for this exercise prior to arrival. Although you of course know nothing about APL, it is reasonable to expect that you should have a good understanding of the application which you intend to program. Toward this end, the following is a list of items that, to the maximum extent possible, should be prepared prior to arrival and brought with you.

1. A concise statement of the problem, application, game, student exercise, or other use you wish to make of the computer in an interactive mode. This should include how you expect the program to be used (as drill-and-practice, gaming or simulation, problem-solving, etc.), by whom (e.g., beginning undergraduates, managers, advanced M.B.A.'s, etc.), and to what end (i.e., what is the object of the program; e.g., to attack problems too complex for hand solution, to reinforce materials already presented by traditional means, etc.). Also included in this should the formula, algorithm, or other logic required to arrive at the solution, result, etc. This is similar to what you put in your application questionnaire, but unfortunately most of them were rather sketchy.

2. A general description of how the program works and what the user can expect from it. This statement, which will naturally be refined after you arrive, will be the kind of documentation you would like to see accompany your program in published form. It should include a specification of the inputs, parameters, and other variables that are necessary on the part of the user in order to make the program work. For example, in a simple program to calculate the terminal value of $1.00 invested at compound interest, i, for N periods, the user must be told that he needs to supply both i and N.

3. Data which can be used to test the program. These data should be of two types: one should be sample or typical data such as would be used to illustrate the use of your program; the other should be a wide range of data designed to test all possible conditions (e.g., what if a user tries to run the program with zero — or an unreasonably large number? what happens if some input is missing? etc.) In other words, try to think of all the ways the program might be used — correctly or incorrectly — and then devise data to test these possibilities.
For some of you with little or no experience in computing, you may feel that it is extremely difficult for you to compile such information prior to the beginning of the course; but it is important that you try, for without having carefully thought through the problem that you wish to undertake, you may find that the two weeks are over before you have half begun.
6.0
PROGRAM ADMINISTRATION
AND
PROMOTIONAL MATERIALS
6.1

Program Announcement Brochure
THE PROGRAM

An intensive two-week program, supported by a grant from the IBM Corporation, will be conducted by the Graduate School of Management at UCLA from June 29 to July 11, 1975. Its purpose is to introduce faculty members from business schools and colleges throughout the country to the use of interactive computing in management education.

Unlike other programs which have been offered elsewhere and which have dealt with the use of FORTRAN and computer batch processing, this new program will focus upon the advantages offered by time-sharing languages for the teaching as well as for the practice of management. Specifically, it will cover the following four areas:

1. The use of conversational, interactive computing in business.
2. The APL programming language as a new and powerful means for providing this interactive capability.
3. The methods for teaching APL to business students.
4. The ways in which APL can be used for teaching accounting, finance, operations research, and other management subjects.

For many faculty members who have made little or no use of computing in their teaching, the mastery of a computer language represents a substantial hurdle. They feel that in order to use computers they must first learn a batch processing language, with all the accompanying intricacies of the job control language, before they can move on to a time-sharing language like APL.

Such views are unfortunate, for they preclude many teachers from taking advantage of the assistance of the computer because of this perceived high start-up cost. Therefore, it is the intention of this program to help overcome this barrier and to make the transition from non-use to informed use as easy and painless as possible.

Toward this end, the use of APL is particularly appropriate, for it is both simple to learn and yet powerful to use. Rapidly growing in popularity, this conversational, interactive computer language is being used in such dissimilar industries as manufacturing, insurance, and transportation. It is also widely available through commercial time-sharing services.

As a prerequisite to attendance, each participant will be expected to define a potential application in his or her own area of specialization prior to arrival. (This involves specifying a learning objective, structuring the problem or exercise, and gathering the data necessary to test and illustrate the use of the resultant program.) After arrival, this application will be coded into APL. In this way, not only will the participants be able to acquire a deeper understanding of APL relevant to their own discipline, but they will also have produced working programs (or series of programs) which can be put into use immediately upon return.

These programs, identified with the author's name and accompanied by a description of their use, will be published as a monograph after the completion of the program. Copies will be furnished to all participants, to the IBM Corporation, to member schools of the American Assembly of Collegiate Schools of Business (AACSB), and to other interested parties.

Participation is limited to 24 faculty members. The cost of the program, as well as all of the living expenses of the participants while they are at UCLA, will be borne by the sponsoring grant from IBM. Travel costs to and from Los Angeles are not included.

TOPICS TO BE COVERED

1. Introduction to Computing Concepts
   A brief overview of computer terminology and technology, covering the general functioning of computer systems, especially those supporting time-sharing services.

2. Introduction to APL
   A description of the concept and background of APL, including a demonstration of its various features.

3. The APL Functions
   The complete set of APL primitive functions, showing their use in conjunction with scalars, vectors, and matrices.

4. User-Defined APL Functions
   The writing of programs, or, in APL terminology, the defining of one's own functions.
5. **Systems Commands and Workspace Management**
   The concepts of public and private libraries, the manipulation of workspaces and their contents, and the functioning of the various systems commands.

6. **Report Generation**
   The preparation of reports, both in a tabular and a graphical mode.

7. **File Systems**
   The concepts of shared variables (SV) and file systems (such as STSC's APL Plus File System).

8. **Programming Techniques**
   A number of APL programming procedures and techniques, exploring the best of current practice.

9. **Application of APL to Business**
   The applications of interactive computing to diverse business problems, drawing upon the experience of the instructional staff, outside speakers, and the program participants.

10. **APL as an Aid in Management Education**
    The use of APL to help teach accounting, finance, operations research, and other disciplines; also, the participants' development of their own applications.

11. **Methods for Teaching APL**
    The teaching strategies used at UCLA over the last four years in introducing APL to well over a thousand M.B.A. students.

**THE FORMAT**

The program will begin on Sunday, June 29th, and end on Friday, July 11th. A reception and welcoming dinner will be held on Sunday evening, with classes beginning the following morning.

All classes will be held at the Graduate School of Management (GSM) from 8:30 A.M. to 5:30 P.M. each day, with the evenings and middle weekend free for studying or recreation.

The participants will be split into two groups of twelve each. One group will attend a lecture while the other works in the computer laboratory. Every two hours the two groups will switch. This allows for small class sizes and maximum individual attention.

Of the three instructors associated with the program, at least two will be lecturing or assisting in the lab at all times. Over the two-week period, each attendee will have 34 hours of lecture and 44 hours of laboratory time, including the programming of his or her own application program.

**WHO SHOULD ATTEND**

The program is designed for those business school faculty members who wish to enhance their teaching skills by applying the power of interactive computing to their area of specialization. There is no need to have had any previous experience in computing. Therefore, those who have never used computers, as well as those who had had some programming experience, are invited to apply.

The selection of the 24 participants will be governed, at least in part, by the desire to obtain a broad cross section of disciplines, functional areas, and representation from among AACSB member schools. In the event the program is oversubscribed, all applications for attendance will be retained in case a subsequent offering is possible.

Requests for applications should be sent by March 15th to:

Dr. Ephraim R. McLean  
Graduate School of Management  
University of California  
Los Angeles, California 90024
THE FACULTY

EPHRAIM R. McLEAN, Program Coordinator
Professor McLean is presently the Director of the Center for Information Studies, and was formerly the Chairman of the Computers and Information Systems Curriculum Committee, both at GSM. Instrumental in bringing APL onto the UCLA campus, he taught the first regularly-scheduled course in APL at UCLA and has taught it every year since. In conjunction with Professor Siler, he has taught APL to over a thousand graduate students. When the widely-used book APL: An Interactive Approach (which is to be the primary text for the program) was recently revised in preparation for the second edition, Professor McLean was retained by John Wiley & Sons as a consultant on the revision.

KENNETH F. SILER
Professor Siler, the other faculty member associated with the program, is the current Chairman of the Computers and Information Systems Curriculum Committee and has worked closely with APL for the last two years.

JAMES W. SCHENCK
Mr. Schenck, the third member of the instructional staff, was formerly the Coordinator of Computer Services for GSM and is currently a doctoral student in the School. He is one of the co-authors (with J. W. Buckley, M. R. Nagaraj, and D. L. Sharp) of Management Problem-Solving with APL, (Los Angeles: Melville Publishing Co., 1974), another text in the program.

ACCOMMODATIONS AND FACILITIES

All of the participants will be housed at the Holiday Inn in Westwood, immediately adjacent to the UCLA campus. A special bus will furnish transportation between the motel and GSM. Meals will be provided at the Holiday Inn and at the UCLA Faculty Center, or participants may choose to dine at one of the many restaurants in Westwood.

The lecture and laboratory sessions at GSM will be conducted "on-line." For lectures, the air-conditioned classroom will be equipped with a computer terminal and with closed circuit television so that everyone in the room can follow the output directly.

The computer laboratory, a carpeted, sound-deadened room equipped with 12 computer terminals, will be reserved for the exclusive use of the participants during the program. In addition, a number of other terminals located within GSM will be available on a first-come, first-served basis. A portable terminal will be placed in the Holiday Inn so that participants may work in the evening without having to return to GSM.

With regards to the terminals themselves, GSM has a wide assortment, including A.J. 841's, IBM 2741's, G.T.&E. 5741's, and Tektronix 4013 graphic terminals. The main support computer is the UCLA Campus Computer Network's IBM 360/91 with four million bytes of main storage. Two commercial companies, Proprietary Computer Systems and Scientific Time Sharing Corporation, are available as back-up, if necessary, to insure uninterrupted service.
6.2

Invitation Letter to

Business School Deans
DEHKELEY • DAVIS • IHE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO

OFFICE OF THE DEAN
GRADUATE SCHOOL OF MANAGEMENT
LOS ANGELES, CALIFORNIA 90024

15 January 1975

Dear Dean : 

It is my pleasure to invite your participation in a program that I know will be of interest to you.

With the use of computers expanding on every front, it is difficult for those of us in the academic world to keep up with these developments, let alone employ them in our own educational programs. Significant among these developments is the use of conversational, interactive computing. With this capability being increasingly employed in government and industry, it is unfortunate when the students we are graduating are not familiar with this powerful aid to management. But before our students can use this technology, our faculty must first become comfortable with it. This brings me to the purpose of this letter.

Next summer, a program is being offered at the Graduate School of Management here at UCLA on the subject of Interactive Computing for Management Education. The enclosed brochure explains this program in detail.

Made possible by a generous grant by the IBM Corporation, this two-week program provides an opportunity for 24 faculty members to attend an intensive workshop in the use of APL for management education. All of the expenses connected with attending will be borne by the IBM grant, with the exception of the travel expenses to and from Los Angeles. In this way one or more of your faculty members can substantially upgrade his or her skills by participating in a program which, if taken commercially, would cost the individual several hundred dollars.

In offering this program, we believe that we bring to it a perspective and a measure of experience that is unique in the country. The program will be directed by Dr. Ephraim R. McLean who was instrumental in bringing the interactive computer language APL to the UCLA campus several years ago and is widely regarded as one of the leaders in the country in its application to management education.
As is described in the brochure, there are only 24 openings in the program, so it is possible that there may not be room for everyone who wishes to attend. May I urge you therefore to bring the enclosed material to the attention of those of your faculty whom you feel would most benefit by attending and have them forward their application to Dr. McLean. It is not necessary that they have any previous experience with computing or that APL is currently being used on your campus.

I am excited by the promise of this program and hope that you share my enthusiasm. May we hear from you, or from your designated faculty member, in the near future?

Sincerely,

Harold M. Williams
Dean

Enclosure
6.3

Publicity
ADVANCES IN INTERACTIVE COMPUTING

"Today the computer is as natural an educational tool as paper and pencil for the Tuck students, and their ability to solve operational problems with quantitative computer techniques is highly attractive to prospective employers."  
**Tuck Today**

Seventy-five percent of AACSB schools responding to our 1974 survey have time-sharing capability. That compares to 43 percent in the 1970 survey. What are the advantages of T/S over batch processing? Each year our March issue is dedicated to that question.

We begin our comparison with an update of Dartmouth's T/S approach. During our 1966 survey, 11 schools were visited. Of those 11, Dartmouth's Tuck School of Business proved to be a strong user of T/S. We've revisited that campus during each of our triennial surveys. As in previous visits, Tuck revealed some characteristics which can be emulated by other schools—even schools with very modest computing budgets.

**Objective in Computer Education**

When asked the principal objective in educating Tuck students about the computer, Dean John Hennessey responded, "To acquaint students with the power of the computer as an aid to decision making analysis—an aid to thought—for the practicing manager."

Dean Hennessey then expanded this concept: "We want to make sure our students are not in any way frightened by the computer. . . . We hope that at least students will become familiar with the idea of computers as a natural part of the managerial environment. Also, we want students to know how computers will be directly and indirectly useful to them in their work here at Tuck."

**Motivating the Faculty to Computer Use**

For a decade, Tuck's faculty has been highly computer-oriented. In the early 1960's Dean Karl Hill acquired an IBM 1620 and installed it in the Tuck School basement. His approach was to make computing resources available—both equipment and advisors. Gradually the faculty began to utilize those resources. I asked Dean Hill's successor the approach to motivating new faculty to acquire a computer capability. Dean Hennessey replied, "When a new faculty member arrives on our campus he recognizes that everybody else is using the computer as a natural part of their environment . . . almost as easily as telephones are used. I use it myself—it's required of me in order to access the management information system which President Kemney has installed as a part of Dartmouth's administration . . . It's a rare faculty member who wouldn't want to be a part of our computer activities."

**What's New at Dartmouth?**

All three AACSB surveys indicated the importance of faculty participation for achieving a viable computer curriculum. Even an extremely capable instructor in the core computer course will not be able to convince students of the value of the computer—that value must be demonstrated through assignments in subsequent courses. Dartmouth has shown the way to create such an environment. Computer proficient faculty in the functional area courses encourage students to use the computer to acquire greater depth of knowledge in each subject area.

Such an environment is not dependent upon, but is facilitated by time-sharing. Ease of access is a principal motivator—to both faculty and students. Page 2 will describe some T/S innovations at Tuck School.

SUMMER FACULTY EDUCATION PROGRAMS

**UCLA**

A two-week faculty education program, supported by a grant from the IBM Corporation, will be conducted by the Graduate School of Management at UCLA from June 29 to July 11, 1975. Its purpose is to introduce faculty members from business schools and colleges throughout the country to the use of interactive computing in management education.

Specifically, it will cover the following four areas:

1. The use of conversational, interactive computing in business.
2. The APL programming language as a new and powerful means for providing this interactive capability.
3. The methods for teaching APL to business students.
4. The use of APL in teaching accounting, finance, operations research, and other management subjects.

**No Cost To Participants**

Participation is limited to 24 faculty members. The cost of the program, as well as all of the living expenses of the participants while they are at UCLA, will be borne by the sponsoring grant from IBM. Travel costs to and from Los Angeles are not included.

**Who Should Apply**

The program is designed for those business school faculty members who wish to enhance their teaching skills by applying the power of interactive computing to their area of specialization. There is no need to have had any previous experience in computing.

Selection of the 24 participants will be governed, at least in part, by the goal to obtain a broad cross section of disciplines, functional areas, and representation from among AACSB members schools.

Although the cutoff date is March 15, if you immediately telephone Dr. Ephraim McLean, it is possible to have your application considered. (213) 825-7982.

**SUNY-BINGHAMTON**

Although not formulated at this time, another 3-week Summer Institute in APL will be offered at SUNY-Binghamton in 1975. The institute is largely motivated by numerous requests from APL users and potential APL users from academia and industry for various APL techniques, APL applications, and APL curricular development topics. The three-week syllabus involves: Elementary APL, APL packages, and Advanced APL.

Sessions are suitable for anyone from academia or industry. For complete details, session outlines, fee sheets, and registration forms contact Geraldine McDonald, Computer Center, SUNY-Binghamton, Binghamton, New York 13901.
In reviewing use of APL with other business educators across the country, the question which most commonly arises is: "But why don't you use BASIC?" UCLA first hoped that both would be available. The campus computer center agreed to support only one. The school selected APL, for the following reasons:

1. APL is easy to learn. Although the special symbols and character set are a little strange at first, they are quickly mastered and their power and conciseness are soon appreciated.

2. APL is unique. APL's powerful operations and matrix-handling ability set it apart from other languages. But this supposed "disadvantage" arises only if students do, in fact, go on to some other language. However, most students find little reason to move on to some other language. And again, UCLA's emphasis is on developing problem-solvers and decision-makers—not journeymen programmers.

3. APL is widely available. APL has been implemented, in one way or another, by almost all computer manufacturers. It is true that the number of different machines for which BASIC interpretive compilers presently exist is greater than for APL. Also, because of APL's greater power and correspondingly greater memory requirements, it is unlikely that it will ever be implemented on any of the mini-computers. Finally, in terms of the number of service bureaus, again far more are offering BASIC than APL. However, the average user typically does not care whether a language is available on five machines or fifty; his major concern is: "Is it available to me?" The answer, in most major cities in the U.S., is yes; and the availability of APL appears to be growing.

4. APL has large libraries of programs. The number of APL programs in the public domain is quite large. It is true, of course, that they are nowhere near as numerous as those in BASIC, but the APL libraries are growing rapidly, fueled by such major users as IBM itself and by American Airlines' comprehensive APL-based management information system.

Although BASIC might have an edge at this point, APL is gaining so rapidly that soon the number of available programs, as well as the number of companies offering time-sharing services, will result in differences that are more apparent than real.

Business School Applications

1. Finance
   - Capital budgeting is one area in which a fair amount of work has been done. Such variables as cost of capital, differing depreciation schedules and tax treatments, variable income streams, and provisions for risk can all be taken into account and tested for their relative sensitivity.
   - Two examples of specific assignments which have been undertaken involve computer acquisition and property tax evaluation. In the first case, the frequently-encountered problem of whether to lease or buy computer hardware was studied. Discount rates, equipment configurations, expected operating schedules (e.g., one shift or two), and charging and pricing algorithms were all varied to arrive at the most desirable solution. In the second case, a real estate company's capital asset management was partially dependent upon local property tax rates, so the extent of this sensitivity was analyzed.
   - Another application in the financial area is the use of corporate financial data from such sources as Standard and Poor. Financial ratio analysis across companies or over time has been undertaken. Also, various types of financial forecasting and trends analysis have been studied through the use of regression techniques and time series analysis, in the field of investment portfolio selection, the study of risk by means of beta factor analysis has been done.

2. Statistics
   - With the availability of STATPACK2 from the University of Alberta, as well as other statistical tools in the APL public library, a variety of problems can be investigated. In a course on measurement theory, analysis of variance and problems of data transformation were studied. For statistics classes, both classical statistical techniques, such as correlation, regression, and analysis of variance applications of various types are undertaken; and, in decision theory, much of the calculation drudgery has been eliminated through the use of APL modules without sacrificing the mastery of the concepts.

3. Planning and Control
   - Of the several classes within the school that focus on planning and control, one is built almost entirely around the use of APL. Student teams have developed planning models, each in a setting of their own choice. In creating these planning systems, stress is placed on the importance of the user-interface and upon such issues as data and file management, security features, and real-world feasibility.

4. Operations Research
   - By using a variety of programs in the public library, students are able to attack nonlinear and integer programming problems which, because of the need to solve a linear programming subprogram for each iteration, are usually so long that only the simplest are assigned. Now, with APL routines to handle the L.P. calculations, students can devote their attention to understanding the underlying nonlinear or integer algorithm. Resource allocation decisions and transportation and shortest route problems are regularly assigned and students can perform a number of sensitivity analyses on objective functions, constraint variables and coefficients of the objective functions.

5. Management Gaming
   - Probably the most extensive use of APL is in connection with the planning of the general-purpose management game which is an integral part of the M. B. A. students' first year of study. Externally, it is quite similar to the IBM APL Business Simulation. Internally, it is constructed rather differently. The availability of the APL Plus-batch file interface permits the best of both worlds.

   In almost all business games which involve more than two teams, the inputs (in the form of various management decisions) must be provided by all teams before the game can be run for the next period. This is intrinsically a batch operation. On the other hand, it is very helpful if each team can prepare input, and analyze output, in an interactive mode.

   There are 36 teams divided into 12 industries of 6 firms each. The teams prepare input at the APL terminals, making use of whatever tools they find helpful: resource allocation models, financial pro forma income projections; forecasts based upon regression techniques; and, in a few cases, specially-prepared simulation models of the industry and the competition (i.e., a simulation of a simulation).

   Each of the teams then stores its decision in a specially-accessed and protected--APL Plus file. When all the decisions are in, they are passed to a temporary batch file. From here they are input into a FORTRAN-coded batch program which actually executes the game for the next period of play. The results of this execution are printed on the high-speed remote printer in the School as well as being returned to each team's APL stored workspace; there they can analyze the results for the next round of competition.

6. Employment
   - The campus computer center agreed to support APL. The decision was made following a study of risk by means of belief factor analysis has been done.
6.4

Application Questionnaire
and Cover Letter
Thank you for your interest in our program, Interactive Computing for Management Educators, to be held here at UCLA during the period June 29 - July 11, 1975. The response has been such that we expect to have many more applicants than we have openings. For this reason, we will be forced to select from among those expressing interest in order to obtain the desired class size of 24. As stated in the descriptive brochures, it is our desire "to obtain a broad cross section of disciplines, functional areas, and representation from among AACSB member schools."

Toward this end, I am enclosing an application questionnaire which I would like for you to fill out and return to me by April 5th. I know this questionnaire may appear formidable, but only in this way can we obtain the broad base of information we need in order to make intelligent decisions. Therefore, I beg for your indulgence.

As of the writing of this letter, it appears that there will be approximately five applicants for every one opening. On the surface, this may discourage you from completing the questionnaire, especially when you see the questions relating to computing background. You may assume that because you have little or no experience, your chances are correspondingly lower. Actually, if anything, the reverse is true. As stated in the brochure "there is no need to have had any previous experience in computing." Also, "it is the intention of this program...to make the transition from non-use [emphasis added] to informed use as easy and painless as possible."

A second reason for completing and submitting the questionnaire — even if you are not selected — is that your application, together with others who were unable to be accepted, will form a "data base" of evidence of the widespread interest for programs such as this. Hopefully, such a large list of individuals would move IBM or some other sponsoring organization to underwrite a second or third offering. Of course, there is no guarantee of this, but I would be remise if I did not mention it as a possibility.
Therefore, may I enlist your care and assistance in completing the application questionnaire and returning it to me here at UCLA. All applicants will be notified of the final selections by the second week in April. Thanks again for your interest and help.

Sincerely,

EPHRAIM R. McLEAN
Assistant Professor of
Information Systems
Director, Center for
Information Studies

ERM/mc
Enclosure
INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California
Los Angeles, California 90024

29 June - 11 July 1975

Application Questionnaire

Please fill out this questionnaire as completely as possible and mail to the address below by April 5, 1975.

Professor Ephraim R. McLean
Graduate School of Management
University of California
Los Angeles, California 90024
I. Information Concerning Applicant

1. Full Name ____________________________

2. Present Title __________________________

3. Institution ____________________________

4. Address (include ZIP Code):
   a. Business ______________________________________________________
      ______________________________________________________
      ______________________________________________________
      ______________________________________________________
      Telephone ____________________________
   b. Home ______________________________________________________
      ______________________________________________________
      ______________________________________________________
      ______________________________________________________
      Telephone ____________________________

5. Date of Birth __________________________ Birth Place __________________________


8. Education: Type of Degree Date Obtained Institution Where Obtained
   a) Undergraduate: ________ ________ __________________________
   b) Graduate: ________ ________ __________________________

9. Discipline or Area of Specialization: ____________________________________________

10. List the three courses which you teach most regularly:
    Title Topic (if not clear from title)
    a. __________________________
    b. __________________________
    c. __________________________

    In which of these courses do you anticipate making the greatest use of interactive computing? a. b. c.
    If some other course, please describe: ____________________________________________
    ____________________________________________
    ____________________________________________
II. Information Concerning Applicant's Computer Background (if any).

1. Have you ever written a computer program or taken a course in computer programming?
   No:___________ If no, please go on to Question II.3. below.
   Yes:___________ If yes, please state in which language(s) these programs were written
   ____________________________________________________________
   ____________________________________________________________

2. List the computer courses you have taken (do not list more than 5):

<table>
<thead>
<tr>
<th>Title</th>
<th>Topic (if not clear from title)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td></td>
</tr>
<tr>
<td>c.</td>
<td></td>
</tr>
<tr>
<td>d.</td>
<td></td>
</tr>
<tr>
<td>e.</td>
<td></td>
</tr>
</tbody>
</table>

3. Have you ever used the computer in a conversational, interactive mode (e.g., APL, BASIC, TSO, etc.)?
   No:________
   Yes:_______ If yes, please describe:
   ____________________________________________________________
   ____________________________________________________________

4. Have you used the computer in any other way not described above?
   No:________
   Yes:_______ If yes, please describe:
   ____________________________________________________________
   ____________________________________________________________

III. Information Concerning Applicant's Institution

1. Number of Students Enrolled:

<table>
<thead>
<tr>
<th>Business School or Department</th>
<th>Total Campus</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Undergraduate:</td>
<td></td>
</tr>
<tr>
<td>b) Masters:</td>
<td></td>
</tr>
<tr>
<td>c) Doctorate:</td>
<td></td>
</tr>
<tr>
<td>d) Total:</td>
<td></td>
</tr>
</tbody>
</table>

2. Year in which business school or department was founded:______________________________
3. Total number of faculty F.T.E.'s (Full Time Equivalents): _______________________

4. Type of computer most frequently used by business school faculty and students. (Please give manufacturer's name and model number. E.g. IBM 360/50, IBM 370/145, Burroughs 5700, Xerox Sigma 7, etc.): ____________________________

5. Computer available to business school faculty and students, if different from Question III.4. above. (Again, please give manufacturer's name and model number): ____________________________

6. Does the business school currently have a terminal-based interactive computing service available to it on a regular basis?
   No:_______ If no, please go on to Question III.9. below.
   Yes:_______

7. What type of interactive service is available?

   BASIC: ___________  APL (Burroughs): ___________
   APL/360: ___________  APL/CMS: ___________
   APL * PLUS: ___________  APL/SV: ___________
   APL (UNIVAC): ___________  APL (Xerox): ___________
   Other (please describe): ____________________________

8. Number of computer terminals available for student and faculty use: ___________

9. What plans are there for the future with regards to acquiring or enhancing the interactive computing capability for the business school and/or the campus as a whole (please be as specific as you can)? ____________________________

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

IV. Information Concerning Use of Interactive Computing

1. Do business school students currently take a required course in computing?

   No:_______
   Yes:_______ If yes, please describe this course: ____________________________

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

288
2. Are there other courses in the curriculum which make use of the computer?
   No:_____
   Yes:_____ If yes, please list them (if more than 5, list only those closest to your discipline or area of specialization):

<table>
<thead>
<tr>
<th>Title</th>
<th>Topic (if not clear from title)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a._________</td>
<td></td>
</tr>
<tr>
<td>b._________</td>
<td></td>
</tr>
<tr>
<td>c._________</td>
<td></td>
</tr>
<tr>
<td>d._________</td>
<td></td>
</tr>
<tr>
<td>e._________</td>
<td></td>
</tr>
</tbody>
</table>

3. As part of the program, Interactive Computing for Management Educators, each participant will be required to develop and program (in APL) an application relating to his or her discipline. If selected, describe what your application would be:
4. Recognizing that there may be many more applicants than there are openings in this program, please make a brief statement as to why you wish to attend and what you believe you will be able to do with the experience once you return to your own campus:

5. Although all expenses while at UCLA will be covered by the sponsoring grant from IBM, travel to and from Los Angeles will not be covered. Please indicate the source of such travel funds:

School or university sources:____________________
From your own pocket:____________________
Other (please describe):____________________

6. If you are unable to get into the June 29th program because of the limited number of openings, would you be interested in attending this program at some later date (assuming that additional support could be obtained)?

No:__________
Yes:__________ If yes, please indicate your preference as to time:

Late July 1975 :__________
Mid-August 1975:__________
Summer 1976 :__________
Other :__________

Thank you for your care and patience in completing this application questionnaire.
6.5

Letter of Acceptance
On behalf of Ken Siler, Jim Schenck, and myself, I would like to take this opportunity to welcome you personally to the upcoming program "Interactive Computing for Management Educators" here at UCLA.

At final count, there were 105 faculty members who made application to the program and so the selection process proved to be very difficult indeed. I believe, however, that those who were selected represent a broad cross section in terms of disciplines, schools, and geographical distribution. I hope that the two weeks that you spend with them—and with us—will prove to be enjoyable as well as productive.

The program itself is described in a fair amount of detail in the announcement brochure and so I won't repeat that information here. However, a few additional details may be helpful.

The program will open with a reception-cocktail party, followed by dinner, beginning at 6:00 P.M. on Sunday, June 29th, at the Holiday Inn of Westwood, 10740 Wilshire Boulevard, in West Los Angeles. A color brochure, describing the Holiday Inn is enclosed. We have made reservations for you at this Holiday Inn for the 12 nights, June 29th through July 10th. If you wish to arrive earlier or stay later (at your expense), please let us know so that we may notify the Holiday Inn of your plans. All of the rooms will be double occupancy, but each room has two queen-sized beds and so I am sure you will be quite comfortable.

The Inn is about 12 miles from Los Angeles International Airport and there is a bus which goes directly to the Inn from either Building 2 or Building 7 at the Airport. The fare is $1.70 and the schedule is as follows:
<table>
<thead>
<tr>
<th>L.A. International Airport</th>
<th>Holiday Inn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bldg. 2</td>
<td>Bldg. 7</td>
</tr>
<tr>
<td>9:25 A.M.</td>
<td>9:35 A.M.</td>
</tr>
<tr>
<td>11:50 A.M.</td>
<td>12:00 A.M.</td>
</tr>
<tr>
<td>1:40 P.M.</td>
<td>1:50 P.M.</td>
</tr>
<tr>
<td>3:10 P.M.</td>
<td>3:20 P.M.</td>
</tr>
<tr>
<td>5:10 P.M.</td>
<td>5:20 P.M.</td>
</tr>
</tbody>
</table>

If you miss the bus, there is of course taxi service; the fare is approximately $6.00 - $7.00.

Breakfasts will be at the Holiday Inn (just charge it to your room) and lunches at the UCLA Faculty Club. A cash stipend will be provided upon arrival so that you can have your choice of dinner at either the Inn or at one of the many restaurants in Westwood. Laundry, dry cleaning, or other personal expenses can also be charged to your room. A special bus will provide transportation between the Inn and the Graduate School of Management. Westwood itself is within walking distance, so you will not require a car unless you wish to plan a trip on your own (e.g., a visit to Disneyland on the middle weekend). Naturally, such trips, as well as any other private entertainment, are not covered by the IBM grant.

The weather in Los Angeles is quite pleasant in June-July (mid-80's), and the dress for classes will be informal. However, the evenings can be cool and so you will probably want something warmer for evening wear.

Classes will be Monday through Friday for both weeks (including the 4th of July), 8:30 A.M. to 5:30 P.M. We plan to finish the formal instruction early in the second week and devote the balance of the time to working on your individual programs. As mentioned in the announcement, these APL programs are to be published in a monograph, and so we are most anxious that everyone have their program completed by Friday morning. Therefore, you may make plans to leave Los Angeles sometime Friday afternoon or evening. More information about the preparation for this programming assignment is covered on an enclosed sheet.

Finally, I am enclosing a list of the course materials and a copy of one of the books we will be using: Marilyn Bohl's *Information Processing*. Although none of the participants are knowledgeable about APL, several do have some background in computing. Therefore, in order that we all have the same vocabulary, I am sending this book in advance so that you can "come up to speed" on basic computer concepts prior to the beginning of the program. It is a well-written, straight-forward text and the enclosed book list indicates those pages...
that will be most useful. If you do not have any previous background in computing, please read the assigned sections before June 30th.

I believe that I have covered most of the key points of the program; however, if you have any questions between now and its beginning, please feel free to call Miss Maddie Contraras, the administrative assistant, at (213) 825-2001. The number for the Holiday Inn is (213) 475-3711. I am looking forward to meeting you personally and to the two weeks we will be spending together.

Sincerely,

Ephraim R. McLean
Assistant Professor of Information Systems
Director, Center for Information Studies

ERM/kw

Enclosures (5)
INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

READING LIST

The following books and materials are to be used in conjunction with the UCLA program on APL for management:


* The Information Processing book is to be read prior to arrival. Although all of the book is informative and potentially useful, the most important parts are found on the following pages: 1-37, 49-93, 106-171, 189-220, 231-235, and 247-282. The other books and materials will be distributed at the beginning of the course.

6/9/75
6.6

Program Evaluation Questionnaire
INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

Program Questionnaire

I. Accommodations

<table>
<thead>
<tr>
<th>Poor</th>
<th>Outstanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Rooms at the Holiday Inn</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>2. Meals</td>
<td></td>
</tr>
<tr>
<td>Breakfasts at the Holiday Inn</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>Lunches at the UCLA Faculty Center</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>Arrangements for independent dinner dining</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>3. Transportation</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>4. Administrative and clerical support (Maddie Contreras)</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>5. Program support (Bob Bradspies)</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>6. Comments and/or general appraisal of housing, meals and other living arrangements:</td>
<td></td>
</tr>
</tbody>
</table>

II. Course

<table>
<thead>
<tr>
<th>Poor</th>
<th>Outstanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Text (Gilman &amp; Rose)</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>2. Other reading materials and handouts</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>3. Lecture content and sequencing</td>
<td>/ / / / / / / / /</td>
</tr>
<tr>
<td>4. Lab assignments</td>
<td>/ / / / / / / / /</td>
</tr>
</tbody>
</table>
5. Presentations by outsiders

Didn't Attend

<table>
<thead>
<tr>
<th></th>
<th>Poor</th>
<th>Outstanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>STSC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xerox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bill Bergquist</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Overall course content

<table>
<thead>
<tr>
<th></th>
<th>Poor</th>
<th>Outstanding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Pace of presentation

<table>
<thead>
<tr>
<th></th>
<th>Poor</th>
<th>Outstanding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. What would you like to see added or expanded?

9. What would you like to see deleted or reduced?

10. Other suggested changes to the structure or content of the program:

11. Other suggestions or comments:

III. Instruction

1. Quality and effectiveness of classroom sessions

<table>
<thead>
<tr>
<th></th>
<th>Poor</th>
<th>Outstanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eph McLean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ken Siler</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jim Schenck</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Quality and effectiveness of lab sessions (Jim Schenck) Poor / / / / / / / / Outstanding

3. Quality and effectiveness of individual coaching and assistance
   Bob Bradspies Poor / / / / / / / / Outstanding
   Eph McLean Poor / / / / / / / / Outstanding
   Jim Schenck Poor / / / / / / / / Outstanding
   Ken Siler Poor / / / / / / / / Outstanding

4. Comments and/or general appraisal of the quality of instruction:

IV. Summary

1. Your overall evaluation of the program as a whole Poor / / / / / / / / Outstanding

2. A brief statement summarizing your overall appraisal of the program:

3. Any other suggestions for improvement?

THANK YOU FOR YOUR ASSISTANCE

9/11/75
6.7

Post Conference Follow-Up Letter
23 July 1975

Now that a week or so has past since the end of the program, I hope you have had a chance to recuperate from the rigors of our two weeks together. It was a demanding experience for all concerned; but I, for one, thoroughly enjoyed myself. I hope you did too.

Jim, Ken, and I are in the process of reviewing the programs that were left with us. On the attached sheet is a checklist of the materials that we have from you. If you want—or need—to work on your program, merely go to one of the APL terminals on your campus and then dial into our system here at UCLA. The numbers again are as follows: 213-825-0911, -0921, 0931, -0941, and -0721. Naturally, you will have to identify to the system the type of terminal you are using (e.g., A1 or A9) for IBM 2741's), just as you did here at UCLA. After that, use your APL number and password and then go to town!

The numbers will be kept active until Monday, August 18th. At that time we hope to have everyone's work in hand and will then begin the detailed editing. If you have any questions or problems, don't hesitate to give one of us a call.

Eph McLean 213-825-1449
Ken Siler 213-825-1252
Jim Schenck 213-825-4196 or 213-825-2870

I hope that the balance of the summer will be pleasant for you and that you will be able to put your newly acquired APL skills to use in the Fall.

Sincerely,

EPHRAIM R. MCLEAN
Associate Professor of
Information Systems

Attachment

ERM/ss