





# Operations Research using Open-Source Software

---

Jeffrey Strickland



*"Global Leaders in Training"*

Lulu.com



**Operations Research using Open-Source Software:**

**No Cost, No Fuss**

Copyright © 2015 by Jeffrey Strickland

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a book review.

All plots and graphs appearing in the text were generated using SCILAB, R, and Excel. Excel is a trademark of the Microsoft Corporation

Printed in the United States of America

First Printing, 2015

ISBN 978-1-329-00404-7

Lulu Press, Inc.

3101 Hillsborough Street

Raleigh, NC 27607

[www.Lulu.com](http://www.Lulu.com)



## **Acknowledgement**

---

This book is dedicated to my nephews Andrew Buff and Daniel Plunkett, two gentlemen and scholars.

I would like to thank my bride, Laurie, and our two daughters, Mariah and Evie, for tolerating my love for writing.



# Table of Contents

---

|   |           |
|---|-----------|
| ACKNOWLEDGEMENT .....   | I         |
| TABLE OF CONTENTS.....  | III       |
| PREFACE .....   | XV        |
| <b>1. WHAT IS OPERATIONS RESEARCH? .....</b>                                | <b>1</b>  |
| 1.1. HOLISTIC ANALYSIS AND OPERATIONS RESEARCH.....                         | 1         |
| 1.2. HISTORY OF OPERATIONS RESEARCH .....                                   | 1         |
| 1.3. WHAT DO OPERATIONS RESEARCH ANALYSTS DO?.....                          | 2         |
| 1.4. GETTING THE QUESTION RIGHT.....  | 4         |
| 1.5. MODELING SYSTEMS AND OPTIMIZATION SERVICES .....                       | 5         |
| 1.6. THE OPERATIONS RESEARCH TOOLKIT .....                                  | 6         |
| 1.7. WHAT IS SCILAB ?.....  | 7         |
| 1.8. WHAT IS R? .....   | 8         |
| 1.8.1. <i>Introduction</i> .....  | 8         |
| 1.8.2. <i>The R environment</i> .....                                       | 9         |
| 1.9. WHAT IS LPSOLVE AND WHAT IS IT NOT? .....                              | 10        |
| 1.9.1. <i>It contains full source, examples and manuals.</i> .....          | 10        |
| 1.9.2. <i>Using the LPSolve an IDE</i> .....                                | 11        |
| <b>2. WHAT IS LINEAR ALGEBRA? .....</b>                                     | <b>13</b> |
| 2.1. VECTORS .....  | 14        |
| 2.2. THE SCALAR PRODUCT OF TWO VECTORS.....                                 | 15        |
| 2.3. MATRIX OPERATIONS.....   | 16        |
| 2.3.1. <i>The Scalar Multiple of a Matrix</i> .....                         | 16        |
| 2.3.2. <i>Addition of Two Matrices</i> .....                                | 17        |
| 2.3.3. <i>The Transpose of a Matrix</i> .....                               | 19        |
| 2.3.4. <i>Matrix Multiplication</i> .....                                   | 19        |
| 2.3.5. <i>Properties of Matrix Multiplication</i> .....                     | 23        |
| 2.4. MATRIX MULTIPLICATION WITH EXCEL .....                                 | 24        |
| 2.5. MATRICES AND SYSTEMS OF LINEAR EQUATIONS .....                         | 25        |
| 2.6. THE GAUSS–JORDAN METHOD FOR SOLVING SYSTEMS OF LINEAR EQUATIONS        | 28        |
| 2.6.1. <i>Elementary Row Operations</i> .....                               | 28        |
| 2.6.2. <i>Finding a Solution by the Gauss–Jordan Method</i> .....           | 30        |
| 2.6.3. <i>Special Cases: No Solution or an Infinite Number of Solutions</i> | 36        |
| 2.6.4. <i>Summary of the Gauss–Jordan Method</i> .....                      | 38        |
| 2.6.5. <i>Basic Variables and Solutions to Linear Equation Systems</i> .... | 40        |
| 2.7. THE INVERSE OF A MATRIX .....  | 42        |

|             |   |           |
|-------------|---|-----------|
| <b>2.8.</b> | <b>BASIC LINEAR ALGEBRA IN SCILAB .....</b>   | <b>46</b> |
| 2.8.1.      | <i>A Note on Notation .....</i>   | 47        |
| 2.8.2.      | <i>Vector Algebra .....</i>   | 47        |
| 2.8.3.      | <i>Matrix Algebra .....</i>   | 49        |
| 2.8.4.      | <i>Solving Linear Equations .....</i>   | 51        |
| 2.8.5.      | <i>Gauss-Jordan.....</i>  | 53        |
| 2.8.6.      | <i>Quadratic Optimization.....</i>  | 54        |
| 2.8.7.      | <i>Indexing.....</i>  | 55        |
| 2.8.8.      | <i>Ranges.....</i>  | 56        |
| 2.8.9.      | <i>Where to find more information .....</i>   | 57        |
| 2.8.10.     | <i>SCILAB Functions and Operators .....</i>   | 57        |
| 2.8.11.     | <i>Factorizations.....</i>  | 59        |
| 2.8.12.     | <i>Matrix manipulations.....</i>  | 60        |
| <b>2.9.</b> | <b>REVIEW PROBLEMS .....</b>  | <b>60</b> |
| <b>3.</b>   | <b>WHAT IS A MODEL? .....</b>   | <b>63</b> |
| 3.1.        | <i>INTRODUCTION .....</i>   | 63        |
| 3.2.        | <i>MODEL NOMENCLATURE .....</i>   | 64        |
| 3.2.1.      | <i>Static and Dynamic Models.....</i>   | 65        |
| 3.2.2.      | <i>Linear and Nonlinear Models .....</i>  | 66        |
| 3.2.3.      | <i>Integer and Noninteger Models .....</i>  | 66        |
| 3.2.4.      | <i>Deterministic and Stochastic Models .....</i>                                    | 67        |
| 3.3.        | <i>MODEL FUNCTIONAL FORMS .....</i>   | 67        |
| 3.4.        | <i>MODELING IN OPERATIONS RESEARCH.....</i>   | 70        |
| <b>4.</b>   | <b>WHAT IS LINEAR PROGRAMMING?.....</b>   | <b>73</b> |
| 4.1.        | <i>WHAT IS A LINEAR PROGRAMMING PROBLEM? .....</i>                                  | 73        |
| 4.1.1.      | <i>The Proportionality and Additivity Assumptions .....</i>                         | 79        |
| 4.1.2.      | <i>The Divisibility Assumption .....</i>  | 80        |
| 4.1.3.      | <i>The Certainty Assumption .....</i>   | 81        |
| 4.1.4.      | <i>Feasible Region and Optimal Solution .....</i>                                   | 81        |
| 4.1.5.      | <i>Problems .....</i>   | 83        |
| 4.2.        | <i>THE GRAPHICAL SOLUTION OF TWO-VARIABLE LINEAR PROGRAMMING<br/>PROBLEMS .....</i> | 83        |
| 4.2.1.      | <i>Finding the Feasible Solution .....</i>  | 85        |
| 4.2.2.      | <i>Finding the Optimal Solution.....</i>  | 87        |
| 4.2.3.      | <i>Binding and Nonbinding Constraints .....</i>                                     | 88        |
| 4.2.4.      | <i>Convex Sets, Extreme Points, and LP.....</i>                                     | 88        |
| 4.2.5.      | <i>The Graphical Solution of Minimization Problems .....</i>                        | 90        |
| 4.2.6.      | <i>Problems .....</i>   | 94        |
| 4.3.        | <i>SPECIAL CASES .....</i>  | 94        |

|         |  |     |
|---------|--|-----|
| 4.3.1.  | <i>Alternative or Multiple Optimal Solutions</i>             | 95  |
| 4.3.2.  | <i>Infeasible LP</i>   | 98  |
| 4.3.3.  | <i>Unbounded LP</i>  | 100 |
| 4.3.4.  | <i>Problems</i>  | 102 |
| 4.4.    | A WORK-SCHEDULING PROBLEM                                    | 103 |
| 4.4.1.  | <i>Problems</i>  | 107 |
| 4.5.    | A CAPITAL BUDGETING PROBLEM                                  | 107 |
| 4.5.1.  | <i>Computing NPV with Excel</i>                              | 109 |
| 4.5.2.  | <i>Problems</i>  | 114 |
| 4.6.    | SHORT-TERM FINANCIAL PLANNING                                | 115 |
| 4.7.    | MULTIPEIOD FINANCIAL MODELS                                  | 120 |
| 4.7.1.  | <i>Problems</i>  | 123 |
| 4.8.    | MULTIPEIOD WORK SCHEDULING                                   | 124 |
| 4.8.1.  | <i>Problems</i>  | 127 |
| 4.9.    | SUMMARY  | 129 |
| 4.9.1.  | <i>Linear Programming Definitions</i>                        | 129 |
| 4.9.2.  | <i>Graphical Solution of Linear Programming Problems</i>     | 129 |
| 4.9.3.  | <i>LP Solutions: Four Cases</i>                              | 130 |
| 4.9.4.  | <i>Formulating LPs</i>                                       | 130 |
| 4.9.5.  | <i>Review Problems</i>                                       | 130 |
| 4.10.   | LINEAR PROGRAMMING USING SCILAB                              | 137 |
| 4.10.1. | <i>Linear programming SCILAB using quapro</i>                | 137 |
| 4.10.2. | <i>Graphical illustration of linear programming solution</i> | 143 |
| 4.10.3. | <i>Lagrange multipliers</i>                                  | 145 |
| 4.10.4. | <i>What are Lagrange multipliers?</i>                        | 146 |
| 4.10.5. | <i>SCILAB's karmarkar function</i>                           | 147 |
| 5.      | WHAT IS THE SIMPLEX ALGORITHM?                               | 153 |
| 5.1.    | HOW TO CONVERT AN LP TO STANDARD FORM                        | 153 |
| 5.1.1.  | <i>Problems</i>  | 157 |
| 5.2.    | PREVIEW OF THE SIMPLEX ALGORITHM                             | 158 |
| 5.2.1.  | <i>Basic and Nonbasic Variables</i>                          | 159 |
| 5.2.2.  | <i>Feasible Solutions</i>                                    | 160 |
| 5.2.3.  | <i>Direction of Unboundedness</i>                            | 163 |
| 5.2.4.  | <i>Why Does an LP Have an Optimal bfs?</i>                   | 167 |
| 5.2.5.  | <i>Problems</i>  | 168 |
| 5.3.    | THE SIMPLEX ALGORITHM  | 168 |
| 5.3.1.  | <i>Convert the LP to Standard Form</i>                       | 170 |
| 5.3.2.  | <i>Is the Current Basic Feasible Solution Optimal?</i>       | 171 |
| 5.3.3.  | <i>Determine the Entering Variable</i>                       | 172 |
| 5.3.4.  | <i>The Ratio Test</i>  | 173 |

|         |  |     |
|---------|--|-----|
| 5.3.5.  | <i>Find a New Basic Feasible Solution: Pivot in the Entering Variable</i>            | 174 |
| 5.3.6.  | <i>In Which Row Does the Entering Variable Become Basic?...</i>                      | 174 |
| 5.3.7.  | <i>Is a Canonical Form Optimal (Max Problem)? .....</i>                              | 179 |
| 5.3.8.  | <i>Summary of the Simplex Algorithm for a Max Problem.....</i>                       | 181 |
| 5.3.9.  | <i>Representing Simplex Tableaus.....</i>  | 182 |
| 5.3.10. | <i>Problems .....</i>  | 182 |
| 5.4.    | USING THE SIMPLEX ALGORITHM TO SOLVE MINIMIZATION PROBLEMS.....                      | 183 |
| 5.4.1.  | <i>Method 1.....</i>   | 183 |
| 5.4.2.  | <i>Method 2.....</i>   | 185 |
| 5.4.3.  | <i>Problems .....</i>  | 186 |
| 5.5.    | UNBOUNDED LPs .....  | 186 |
| 5.5.1.  | <i>Unbounded LPs and Directions of Unboundedness.....</i>                            | 190 |
| 5.5.2.  | <i>Problems .....</i>  | 191 |
| 5.6.    | THE BIG M METHOD.....  | 192 |
| 5.6.1.  | <i>Description of Big M Method .....</i>   | 195 |
| 5.6.2.  | <i>How to Spot an Infeasible LP.....</i>   | 199 |
| 5.6.3.  | <i>Problems .....</i>  | 201 |
| 5.7.    | MULTIATTRIBUTE DECISION MAKING IN THE ABSENCE OF UNCERTAINTY: GOAL PROGRAMMING ..... | 201 |
| 5.7.1.  | <i>Preemptive Goal Programming.....</i>  | 206 |
| 5.7.2.  | <i>Problems .....</i>  | 212 |
| 5.8.    | LINEAR PROGRAMMING WITH SCILAB .....   | 214 |
| 5.9.    | EXCEL EXAMPLE: RADIO BARN.....   | 219 |
| 5.9.1.  | <i>Complete Solution.....</i>  | 220 |
| 5.10.   | GRAPHICAL SOLUTION TO A LINEAR PROGRAMMING PROBLEM.....                              | 221 |
| 5.10.1. | <i>Iso-Profit Line Solution Method.....</i>  | 222 |
| 5.11.   | INTRODUCTION TO SENSITIVITY ANALYSIS .....   | 229 |
| 5.11.1. | <i>Sensitivity Report .....</i>  | 230 |
| 5.12.   | USING EXCEL SPREADSHEETS.....  | 231 |
| 6.      | WHAT IS SENSITIVITY ANALYSIS?.....   | 235 |
| 6.1.    | WHAT AND WHY IS SENSITIVITY ANALYSIS .....   | 235 |
| 6.2.    | REDUCED COSTS.....   | 243 |
| 6.2.1.  | <i>Problems .....</i>  | 246 |
| 6.3.    | DUAL PROBLEM.....  | 248 |
| 6.3.1.  | <i>Finding Dual .....</i>  | 248 |
| 6.3.2.  | <i>Economic Interpretation of Dual .....</i>   | 254 |
| 6.4.    | IMPORTANCE OF SENSITIVITY ANALYSIS .....   | 258 |
| 6.5.    | SUMMARY. GRAPHICAL SENSITIVITY ANALYSIS.....   | 258 |
| 6.5.1.  | <i>Shadow Prices .....</i>   | 259 |

|           |   |            |
|-----------|---|------------|
| 6.5.2.    | <i>Objective Function Coefficient Range</i> .....                               | 259        |
| 6.5.3.    | <i>Reduced Cost</i> .....   | 259        |
| 6.5.4.    | <i>Right-Hand Side Range</i> .....  | 260        |
| 6.5.5.    | <i>Signs of Shadow Prices</i> .....   | 260        |
| 6.5.6.    | <i>Optimal z-Value as a Function of a Constraint's Right-Hand Side</i> .....    | 260        |
| 6.5.7.    | <i>Finding the Dual of an LP</i> .....  | 260        |
| 6.5.8.    | <i>Optimal z-Value as a Function of an Objective Function Coefficient</i> ..... | 261        |
| 6.6.      | REVIEW PROBLEMS.....  | 262        |
| <b>7.</b> | <b>WHAT IS CPM AND PERT?</b> .....  | <b>271</b> |
| 7.1.      | COMPUTATION OF EARLY EVENT TIME.....  | 275        |
| 7.2.      | COMPUTATION OF LATE EVENT TIME .....  | 277        |
| 7.3.      | TOTAL FLOAT.....  | 279        |
| 7.4.      | FINDING A CRITICAL PATH .....   | 280        |
| 7.5.      | FREE FLOAT.....   | 281        |
| 7.6.      | USING LINEAR PROGRAMMING TO FIND A CRITICAL PATH .....                          | 282        |
| 7.7.      | CRASHING THE PROJECT.....   | 283        |
| 7.8.      | PERT: PROGRAM EVALUATION AND REVIEW TECHNIQUE .....                             | 285        |
| 7.9.      | DIFFICULTIES WITH PERT .....  | 288        |
| <b>8.</b> | <b>WHAT IS INTEGER LINEAR PROGRAMMING?</b> .....                                | <b>291</b> |
| 8.1.      | MIXED INTEGER LINEAR PROGRAMMING TERMINOLOGY .....                              | 291        |
| 8.1.1.    | <i>Pure Integer Program</i> .....   | 291        |
| 8.1.2.    | <i>Binary Integer Program</i> .....   | 291        |
| 8.1.3.    | <i>Mixed Integer Program</i> .....  | 292        |
| 8.2.      | BRANCH-AND-BOUND METHOD .....   | 292        |
| 8.2.1.    | <i>General Branch-And-Bound Algorithm</i> .....                                 | 298        |
| 8.2.2.    | <i>Problems</i> .....   | 299        |
| 8.3.      | TRAVELING SALESMAN PROBLEM .....  | 299        |
| 8.4.      | FIXED-CHARGE PROBLEMS .....   | 308        |
| 8.4.1.    | <i>Problems</i> .....   | 317        |
| 8.5.      | SET-COVERING PROBLEMS .....   | 318        |
| 8.5.1.    | <i>Problems</i> .....   | 321        |
| 8.6.      | USING THE EXCEL SOLVER TO SOLVE IP PROBLEMS .....                               | 322        |
| 8.7.      | REVIEW PROBLEMS.....  | 325        |
| <b>9.</b> | <b>WHY PROBABILITY AND STATISTICS?</b> .....                                    | <b>331</b> |
| 9.1.      | INTRODUCTION .....  | 331        |
| 9.2.      | BASIC DEFINITIONS .....   | 331        |

|            |  |            |
|------------|--|------------|
| 9.2.1.     | <i>Population</i> .....  | 331        |
| 9.2.2.     | <i>Sample</i> .....  | 331        |
| 9.2.3.     | <i>Parameter</i> .....   | 332        |
| 9.2.4.     | <i>Statistic</i> .....   | 332        |
| 9.2.5.     | <i>Expected Value</i> .....  | 332        |
| 9.2.6.     | <i>Variance</i> .....  | 332        |
| 9.2.7.     | <i>Sample Mean</i> .....   | 332        |
| 9.2.8.     | <i>Sample Variance</i> .....   | 333        |
| 9.2.9.     | <i>Random Variable</i> .....   | 333        |
| 9.3.       | CONDITIONAL PROBABILITY AND BAYES RULE .....                               | 334        |
| 9.3.1.     | <i>Basic Rules of Probability</i> .....                                    | 334        |
| 9.3.2.     | <i>Bayes' Rule</i> .....   | 337        |
| 9.3.3.     | <i>Problems</i> .....  | 340        |
| 9.4.       | PROBABILITY DISTRIBUTIONS.....   | 340        |
| 9.4.1.     | <i>Probability Density Function</i> .....                                  | 341        |
| 9.4.2.     | <i>The Two Statistical Representations of a Population</i> .....           | 342        |
| 9.4.3.     | <i>Normal Distributions</i> .....  | 344        |
| 9.4.4.     | <i>Sampling Distributions</i> .....  | 346        |
| 9.4.5.     | <i>Problems</i> .....  | 348        |
| 9.4.6.     | <i>Probability Distribution with SCILAB</i> .....                          | 349        |
| 9.5.       | THE CENTRAL LIMIT THEOREM .....  | 355        |
| 9.5.1.     | <i>An Illustration of CLT</i> .....  | 357        |
| 9.5.2.     | <i>Properties of the Sampling Distribution of the Sample Means:</i><br>360 |            |
| 9.6.       | EXPONENTIAL DENSITY FUNCTION.....  | 361        |
| 9.7.       | POISSON PROCESS .....  | 361        |
| 9.8.       | GOODNESS-OF-FIT TESTS .....  | 362        |
| 9.8.1.     | <i>Distribution Fitting</i> .....  | 365        |
| 9.9.       | INTRODUCTION TO R .....  | 369        |
| 9.9.2.     | <i>Hypothesis Testing Primer</i> .....                                     | 372        |
| 9.10.      | REVIEW PROBLEMS.....   | 375        |
| <b>10.</b> | <b>WHAT IS PREDICTIVE MODELING?</b> .....                                  | <b>377</b> |
| 10.1.      | PROPENSITY MODELS .....  | 377        |
| 10.2.      | CLUSTER MODELS.....  | 380        |
| 10.3.      | CONCLUSION .....   | 382        |
| 10.4.      | PRESENTING AND USING THE RESULTS OF A PREDICTIVE MODEL .....               | 382        |
| 10.5.      | APPLICATIONS.....  | 383        |
| 10.5.1.    | <i>Uplift Modeling</i> .....   | 383        |
| 10.5.2.    | <i>Archaeology</i> .....   | 384        |
| 10.5.3.    | <i>Customer relationship management</i> .....                              | 384        |

|            |   |            |
|------------|---|------------|
| 10.5.4.    | <i>Auto insurance</i> .....   | 385        |
| 10.5.5.    | <i>Health care</i> .....  | 385        |
| 10.6.      | NOTABLE FAILURES OF PREDICTIVE MODELING .....                                 | 385        |
| 10.7.      | POSSIBLE FUNDAMENTAL LIMITATIONS OF PREDICTIVE MODEL BASED ON DATA<br>FITTING | 386        |
| 10.8.      | MORE ON CLUSTER MODELS .....  | 387        |
| 10.9.      | DEFINITION .....  | 388        |
| 10.10.     | ALGORITHMS .....  | 390        |
| 10.10.1.   | <i>Connectivity based clustering (hierarchical clustering)</i> .....          | 390        |
| 10.10.2.   | <i>Metric</i> .....   | 391        |
| 10.10.3.   | <i>Linkage criteria</i> .....   | 392        |
| 10.11.     | EXAMPLES USING R .....  | 393        |
| <b>11.</b> | <b>WHAT IS DATA FITTING?</b> .....  | <b>413</b> |
| 11.1.      | INTRODUCTION .....  | 413        |
| 11.2.      | GRAPHICS .....  | 413        |
| 11.3.      | MODEL CHOICE.....   | 417        |
| 11.4.      | PARAMETERS' ESTIMATE .....  | 421        |
| 11.5.      | MEASURES OF GOODNESS-OF-FIT .....   | 425        |
| 11.6.      | GOODNESS OF FIT TESTS.....  | 427        |
| 11.7.      | NORMALITY TESTS .....   | 432        |
| 11.8.      | APPENDIX .....  | 436        |
| 11.9.      | REVIEW PROBLEMS.....  | 437        |
| <b>12.</b> | <b>WHAT IS ORDINARY LEAST SQUARES?</b> .....                                  | <b>439</b> |
| 12.1.      | LINEAR MODEL.....   | 440        |
| 12.1.1.    | <i>Assumptions</i> .....  | 440        |
| 12.1.2.    | <i>Classical linear regression model</i> .....                                | 441        |
| 12.1.3.    | <i>Independent and identically distributed</i> .....                          | 443        |
| 12.1.4.    | <i>Time series model</i> .....  | 443        |
| 12.2.      | ESTIMATION.....   | 444        |
| 12.2.1.    | <i>Simple regression model</i> .....  | 446        |
| 12.3.      | ALTERNATIVE DERIVATIONS .....   | 446        |
| 12.3.1.    | <i>Geometric approach</i> .....   | 446        |
| 12.3.2.    | <i>Maximum likelihood</i> .....   | 447        |
| 12.3.3.    | <i>Generalized method of moments</i> .....                                    | 448        |
| 12.4.      | FINITE SAMPLE PROPERTIES .....  | 448        |
| 12.4.1.    | <i>Assuming normality</i> .....   | 449        |
| 12.4.2.    | <i>Influential observations</i> .....   | 451        |
| 12.4.3.    | <i>Partitioned regression</i> .....   | 451        |
| 12.4.4.    | <i>Constrained estimation</i> .....   | 452        |

|            |   |            |
|------------|---|------------|
| 12.5.      | LARGE SAMPLE PROPERTIES.....  | 453        |
| 12.5.1.    | <i>Sensitivity to rounding</i> .....                                    | 459        |
| 12.6.      | EXAMPLE USING R .....   | 460        |
| 12.7.      | LEAST-SQUARE DATA FITTING WITH SCILAB FUNCTION DATAFIT .....            | 472        |
| 12.7.1.    | <i>The method of least squares</i> .....                                | 472        |
| 12.7.2.    | <i>Function datafit</i> .....   | 473        |
| 12.8.      | REVIEW PROBLEMS.....  | 483        |
| <b>13.</b> | <b>WHAT IS A GENERALIZED LINEAR MODEL?</b> .....                        | <b>485</b> |
| 13.1.      | INTUITION .....   | 485        |
| 13.2.      | OVERVIEW .....  | 487        |
| 13.3.      | MODEL COMPONENTS .....  | 487        |
| 13.3.1.    | <i>Probability distribution</i> .....                                   | 487        |
| 13.3.2.    | <i>Linear predictor</i> .....   | 489        |
| 13.3.3.    | <i>Link function</i> .....  | 489        |
| 13.4.      | FITTING .....   | 491        |
| 13.4.1.    | <i>Maximum likelihood</i> .....   | 491        |
| 13.4.2.    | <i>Bayesian methods</i> .....   | 492        |
| 13.5.      | EXAMPLES .....  | 492        |
| 13.5.1.    | <i>General linear models</i> .....                                      | 492        |
| 13.5.2.    | <i>Linear regression</i> .....  | 492        |
| 13.5.3.    | <i>Binomial data</i> .....  | 493        |
| 13.5.4.    | <i>Multinomial regression</i> .....                                     | 494        |
| 13.5.5.    | <i>Count data</i> .....   | 495        |
| 13.6.      | EXTENSIONS .....  | 495        |
| 13.6.1.    | <i>Correlated or clustered data</i> .....                               | 495        |
| 13.6.2.    | <i>Generalized additive models</i> .....                                | 496        |
| 13.6.3.    | <i>Generalized additive model for location, scale and shape</i> ... 497 | 497        |
| 13.7.      | CONFUSION WITH GENERAL LINEAR MODELS .....                              | 498        |
| 13.8.      | EXAMPLE USING R .....   | 499        |
| 13.8.1.    | <i>Setting up the Model</i> .....                                       | 499        |
| 13.8.2.    | <i>Model Quality</i> .....  | 501        |
| 13.9.      | PROBLEMS .....  | 504        |
| <b>14.</b> | <b>WHAT IS LOGISTIC REGRESSION?</b> .....                               | <b>505</b> |
| 14.1.      | FIELDS AND EXAMPLES OF APPLICATIONS .....                               | 505        |
| 14.2.      | BASICS.....   | 506        |
| 14.3.      | LOGISTIC FUNCTION, ODDS RATIO, AND LOGIT.....                           | 507        |
| 14.3.1.    | <i>Multiple explanatory variables</i> .....                             | 509        |
| 14.4.      | MODEL FITTING.....  | 509        |
| 14.4.1.    | <i>Estimation</i> .....   | 509        |

|  |            |
|--|------------|
| 14.4.2. <i>Evaluating goodness-of-fit</i> .....                        | 511        |
| 14.5. COEFFICIENTS .....   | 515        |
| 14.5.1. <i>Likelihood ratio test</i> .....                             | 516        |
| 14.5.2. <i>Wald statistic</i> .....                                    | 516        |
| 14.5.3. <i>Case-control sampling</i> .....                             | 517        |
| 14.6. FORMAL MATHEMATICAL SPECIFICATION .....                          | 517        |
| 14.6.1. <i>Setup</i> .....   | 518        |
| 14.6.2. <i>As a generalized linear model</i> .....                     | 521        |
| 14.6.3. <i>As a latent-variable model</i> .....                        | 523        |
| 14.6.4. <i>As a two-way latent-variable model</i> .....                | 525        |
| 14.6.5. <i>As a “log-linear” model</i> .....                           | 528        |
| 14.6.6. <i>As a single-layer perceptron</i> .....                      | 530        |
| 14.6.7. <i>In terms of binomial data</i> .....                         | 531        |
| 14.7. BAYESIAN LOGISTIC REGRESSION .....                               | 531        |
| 14.7.1. <i>Gibbs sampling with an approximating distribution</i> ..... | 533        |
| 14.8. EXTENSIONS .....   | 537        |
| 14.9. MODEL SUITABILITY .....  | 538        |
| 14.10. EXAMPLES USING R .....  | 539        |
| 14.11. PROBLEMS.....   | 550        |
| <b>15. WHAT IS A SIMULATION MODEL? .....</b>                           | <b>563</b> |
| 15.1. MONTE CARLO SIMULATION MODEL .....                               | 563        |
| 15.2. DYNAMIC SIMULATION MODELS .....                                  | 564        |
| 15.3. DISCRETE TIME SIMULATION MODELS .....                            | 564        |
| 15.4. DISCRETE EVENT SIMULATION MODELS .....                           | 565        |
| 15.5. SIMULATION ARCHITECTURES .....                                   | 565        |
| 15.6. CONCLUSION .....   | 565        |
| 15.7. REFRESHER ON RANDOMNESS.....                                     | 566        |
| 15.7.1. <i>Uniform Density Function</i> .....                          | 566        |
| 15.7.2. <i>Random Number Generators</i> .....                          | 566        |
| 15.7.3. <i>Additive Congruence Generator</i> .....                     | 567        |
| 15.7.4. <i>Linear Congruential Generator (LCG)</i> .....               | 570        |
| 15.7.5. <i>Random Function in Excel</i> .....                          | 571        |
| 15.7.6. <i>Testing for Randomness</i> .....                            | 574        |
| 15.7.7. <i>Statistics for Correlated Output Data</i> .....             | 575        |
| 15.8. REVIEW PROBLEMS.....   | 577        |
| <b>16. WHAT IS QUEUING THEORY?.....</b>                                | <b>579</b> |
| 16.1. HISTORY OF QUEUING THEORY .....                                  | 580        |
| 16.2. THE PURPOSE OF QUEUING MODELS .....                              | 580        |
| 16.3. QUEUING SYSTEM COMPONENTS .....                                  | 580        |

|            |   |            |
|------------|---|------------|
| 16.4.      | THE ARRIVAL PROCESS.....  | 581        |
| 16.4.1.    | <i>Arrival Characteristics .....</i>  | 582        |
| 16.4.2.    | <i>Review of the Exponential Distribution .....</i>                         | 583        |
| 16.4.3.    | <i>Relationship between the Poisson and Exponential Distributions .....</i> | 584        |
| 16.5.      | SERVICE PROCESS .....   | 585        |
| 16.6.      | QUEUE DISCIPLINE.....   | 585        |
| 16.7.      | MODELING THE ARRIVAL PROCESS .....  | 587        |
| 16.8.      | QUEUING SYSTEMS TAXONOMY .....  | 588        |
| 16.9.      | QUEUING SYSTEM THEORY .....   | 588        |
| 16.9.1.    | <i>Erlang Distribution .....</i>  | 589        |
| 16.9.2.    | <i>Lindley's Formula .....</i>  | 589        |
| 16.9.3.    | <i>A Spreadsheet Simulation of M/M/1 Queue Waiting Times</i>                | 589        |
| 16.10.     | CHARACTERISTICS OF DATA FROM DYNAMIC SYSTEMS .....                          | 590        |
| 16.10.1.   | <i>Influences on Mean Estimation.....</i>                                   | 592        |
| 16.10.2.   | <i>The Initial Transient Period .....</i>                                   | 592        |
| 16.10.3.   | <i>Determination of the Warm-up Period .....</i>                            | 594        |
| 16.10.4.   | <i>Batch Means Computation .....</i>  | 594        |
| 16.10.5.   | <i>Determination of the Desirable Number of Simulation Runs</i>             | 602        |
| 16.11.     | PROBLEMS .....  | 603        |
| <b>17.</b> | <b>WHAT IS SIMULATION? .....</b>  | <b>605</b> |
| 17.1.      | SIMULATION SOFTWARE SELECTION .....   | 605        |
| 17.1.1.    | <i>SIMSCRIPT II.5 .....</i>   | 605        |
| 17.1.2.    | <i>SIMAN .....</i>  | 607        |
| 17.2.      | ANIMATION IN SYSTEMS SIMULATION.....  | 609        |
| 17.3.      | SYSTEM DYNAMICS AND DISCRETE EVENT SIMULATION.....                          | 609        |
| 17.4.      | PARALLEL AND DISTRIBUTED SIMULATION .....                                   | 611        |
| 17.5.      | "WHAT-IF" ANALYSIS TECHNIQUES .....   | 612        |
| 17.5.1.    | <i>Introduction.....</i>  | 612        |
| 17.5.2.    | <i>Likelihood Ratio (LR) Method .....</i>                                   | 614        |
| 17.5.3.    | <i>Exponential Tangential in Expectation Method .....</i>                   | 615        |
| 17.5.4.    | <i>Interpolation Techniques .....</i>                                       | 616        |
| 17.5.5.    | <i>Conclusions.....</i>   | 618        |
| 17.6.      | SIMULATION USING R .....  | 620        |
| 17.7.      | SIMULATION USING EXCEL .....  | 625        |
| 17.8.      | SIMULATION WITH SCILAB .....  | 632        |
| 17.9.      | REVIEW PROBLEMS.....  | 658        |
| <b>18.</b> | <b>WHAT IS DISCRETE EVENT SIMULATION? .....</b>                             | <b>661</b> |
| 18.1.      | DISCRETE EVENT SIMULATION MODEL DEVELOPMENT .....                           | 661        |

|  |  |            |
|--|--|------------|
| 18.1.1.                                      | <i>Event and Event Sequencing</i> .....                        | 661        |
| 18.1.2.                                      | <i>Events and State Changes</i> .....                          | 662        |
| 18.1.3.                                      | <i>Event Scheduling</i> .....                                  | 662        |
| 18.1.4.                                      | <i>The System Clock</i> .....                                  | 663        |
| 18.1.5.                                      | <i>QueueTaxonomy</i> ..... <i>Error! Bookmark not defined.</i> |            |
| 18.1.6.                                      | <i>Random Number Generation</i> .....                          | 663        |
| 18.1.7.                                      | <i>Random Variates</i> .....                                   | 664        |
| 18.1.8.                                      | <i>A Generic Simulation Language</i> .....                     | 664        |
| 18.2.  | <b>A SINGLE-SERVER (M/M/1) QUEUE</b> .....                     | 666        |
| 18.2.1.                                      | <i>M/M/1 Queue Event Routines</i> .....                        | 667        |
| 18.2.2.                                      | <i>M/M/1 Queue</i> .....                                       | 667        |
| 18.2.3.                                      | <i>Event Execution and State Changes Summary</i> .....         | 673        |
| 18.2.4.                                      | <i>Estimating Mean Number in Queue</i> .....                   | 673        |
| 18.2.5.                                      | <i>Event Graphs</i> .....                                      | 674        |
| 18.3.  | <b>SUMMARY</b> .....   | 674        |
| 18.4.  | <b>EXERCISES</b> .....   | 675        |
| 18.5.  | <b>DISCRETE EVENT SIMULATION USING SCILAB</b> .....            | 675        |
| 18.6.  | <b>REVIEW PROBLEMS</b> .....                                   | 681        |
| <b>APPENDIX A – INSTALLING R</b> .....       |  | <b>685</b> |
| <b>APPENDIX B – INSTALLING SCILAB</b> .....  |  | <b>687</b> |
| <b>APPENDIX C – INSTALLING LPSOLVE</b> ..... |  | <b>688</b> |
| <b>REFERENCES</b> .....                      |  | <b>689</b> |
| <b>WORKS CITED</b> .....                     |  | <b>690</b> |
| <b>INDEX</b> .....                           |  | <b>699</b> |



# Preface

---

Having been a sole proprietor for five years with a very low operating budget, software license renewals and purchases became unbearable. Plus the students I taught in the evenings—the non-traditional ones—were opting to download Octave rather than buy the student version of MATLAB. Additionally, with the outrageous cost of textbooks, I broke and not only started self-publishing, but exploring open source tools as well.

This is a compilation of several years of exploratory research into those open-source tools. Based on ease of installation, ease of use, and graphical user interface (GUI), I landed on SCILAB, R, and LPSolve. There were other contenders including Octave and COIN-OR. I wanted to call this book: Operations Research using Tools that do not cost you an arm and a leg but perform as well as those that do.” Of course that is too long to fit on the cover...

This book is intended to be used as an upper-level undergraduate or graduate level textbook for Operations Research with a quantitative emphasis. It could be used in Management Science, but it does expect quite a bit of mathematical preparedness.

SCILAB, R and LPSolve are used extensively, where appropriate, and sometimes two or three of the tools are used to demonstrate the same application. It contains both examples and exercises using these tools. Moreover, I snuck a little Excel into the mix, since most people have it anyway.

The book is organized along traditional Operations Research curriculum beginning with a review of Linear Algebra and an introduction to SCILAB. This review is followed by several chapters of Mathematical Optimization, including modeling with Linear Programs, Graphical Solutions, Sensitivity Analysis, the Dual Problem, Integer Programming and Mixed Integer Linear programs (MILPs).

I skip the traditional Calculus review and the material that depends upon it as a prerequisite, and opt to cover an emerging field of analysis in Operations Research and Management Science: Analytics. Here we

review Probability and Statistics and introduce Descriptive and Predictive Modeling, including Cluster Analysis, Data Fitting, Ordinary Least Squares, Generalized Linear Models and Logistic Regression.

Next we delve into my forte, Simulation. We cover an introduction to Simulation, Queuing Theory, Simulation Models or various sorts, and close with Discrete Event Simulation.

There are many optional topics and whole chapters may be omitted with a little scrutiny. Thus, it could be adapted to a variety of course syllabi.

I do not promise perfection! There may be flaws and you use it at our own peril. However, many hours have gone into its preparation. To keep cost low, minimal outside editing was performed, so you may run into some typos, and my language may be a little relaxed in places.

All plots and graphs appearing in the text were generated using SCILAB, R, and Excel.

I truly hope you enjoy using the book.

# 1. What is Operations Research?

## 1.1. Holistic Analysis and Operations Research

I am not really sure what holistic analysis is, so I will define it. Our English word comes from the Greek ὅλος holos ("all, whole or entire"). Reductionism may be viewed as the complement of holism. Reductionism analyzes a complex system by subdividing or reduction to more fundamental parts. For businesses, knowledge and know-how, know-who, know-what and know-why are part of the whole business economics. Having a holistic view keeps us from missing the forest due to the trees.

**Operations Research (OR)**, or **operational research** in the U.K., is a discipline that deals with the application of advanced analytical methods to help make better decisions. The terms management science and analytics are sometimes used as synonyms for operations research. Yet, in my experience OR extends far beyond either. The figure shows a hierarchy of operations research activities, and I'll let you decide if they are also performed in analytics.

- Data Mining and Machine Learning
- Artificial Intelligence and Expert Systems
- Financial Engineering
- Games, Decision, and Strategic Planning
- Marketing Research
- Investment Science
- Experimental and Engineering Design
- Manufacturing and Production
- Logistics and Transportation
- Supply Chain Management
- Enterprise Resource Planning

## 1.2. History of Operations Research

Operational Research was born during the early years of WWII and matured rapidly. One of its primary function was the planning of Operation Overlord or the Normandy Invasion. It has its foundations in mathematics, computing and economic theories, on which basic tools in optimization and simulation are built. Today OR's are employed by airlines, train lines, logistic systems, delivery systems (e.g., FedEx), defense systems, military, oil companies, insurance companies, financial institutions, manufacturing, marketing and many more.

### 1.3. What do Operations Research Analysts do?

The Operations Research Analyst is a jack-of-all-trades, or at least that has been my experience. Some tend to specialize in a particular area, like mathematical optimization, but I think this is a mistake. One can find people who specialize in a particular methodology or discipline, but they would be challenged to find a good Operations Research analyst with a holistic view of the problem space.

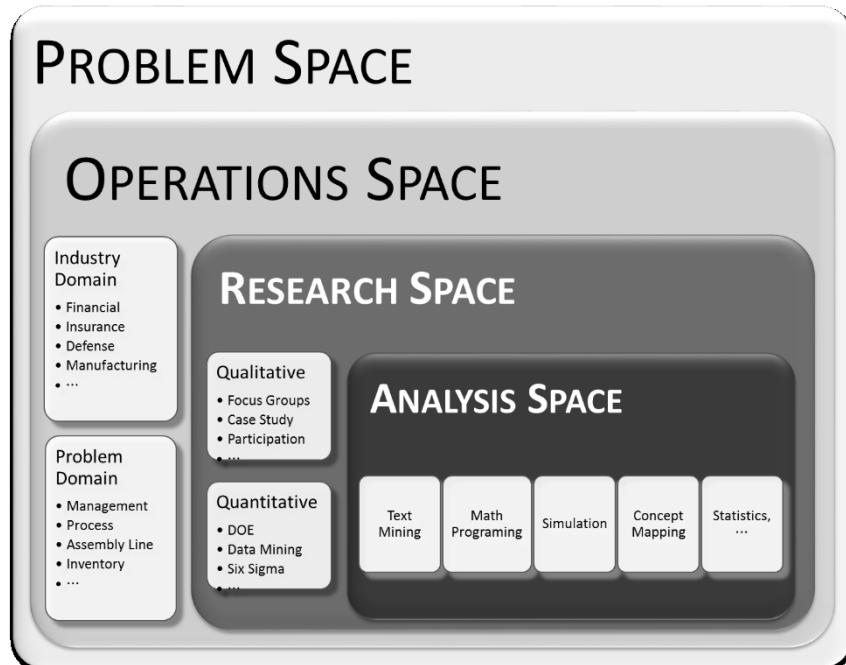
An OR's view of the problem space is really what defines them and describes what they do. Section 1.1 listed some of the activities that ORs engage in, but not without a holistic view of the problem space. Figure one depicts the entire problem space. Mathematically, we could look at it like this:

$$\begin{aligned} \{\{Analysis\ Space\} \subset Research\ Space\} &\subset Operations\ Space \\ &\subset Problem\ Space \end{aligned}$$

The OR Analyst must enter the problem space with the following in mind: (1) the potential operational domains, (2) the types of research that may be used, and (3) the types of analyses that may be appropriate. If one goes in having done nothing more than math programming for 10 years, that analysts is NOT an operations research analyst—they are just a math programmer.

Operations research analysts provide this holistic view, which then allows for the definition of the right problem within any domain, and application of the most appropriate research methodology, using the

most appropriate analyses. You cannot build a house with just a screwdriver, unless you are MacGyver<sup>1</sup>.



**Figure 1.** The Operations Research Problem Space

If we look at the historical context of OR, as we discussed in Section 1.2, we should be able to ascertain that anything short of a holistic point of view may have resulted in operational chaos much worse than missed dropped zones. The entire military operation—the most complex operation ever executed—could have easily failed.

---

<sup>1</sup> MacGyver is an American action-adventure television series created by Lee David Zlotoff. Henry Winkler and John Rich were the executive producers. The show follows secret agent Angus MacGyver, played by Richard Dean Anderson, who works as a troubleshooter for the fictional Phoenix Foundation in Los Angeles and as an agent for a fictional United States government agency, the Department of External Services (DXS). Resourceful and possessed of an encyclopedic knowledge of the physical sciences, he solves complex problems with everyday materials he finds at hand, along with his ever-present duct tape and Swiss Army knife.

## 1.4. Getting the Question Right

As an analysts and modeler I have a variety of customers with a variety of problems in business. We usually frame their “question” as a business case. The question of course is something they require an answer for it usually stems from a problem or a desire to have more Share of Wallet or other metric. What is interesting here is though they know they have a question that needs to be answered, they often do know how to state their question.

In "Making Analysis Relevant", Mr. Vince Roske reiterated a recipe provided by John D. (Dave) Robinson, MG USA, Ret, when he was the Director of Joint Staffs J-8. The recipe is provided below:

- What's the question?
- What's the “real” question?
- What do the final slides look like?
- What do I already know?
- How do I get the remaining information that I need?

So, this idea of what is the real question is not new and it is not unique to business.

An analyst, we have to draw that real question out through dialogue with our customers. Doing this is often as much an art as it is a science. If we do not do it, we stand a good chance delivering a well-built solution that happens to be the wrong one. I have done this at least twice in the last three years, knowing that getting the question right was paramount. In spite of my failures I have found some keys to help with formulating the business case.

1. **Tie it to a metric.** Even if currency is not at stake, you have to have a metric to focus on. Otherwise you can fail to see the forest for the trees. As your project rolls along you will forget what you are building to without this. IT can be Return on Investment, Share of Wallet, Key Performance Parameter, or something else, but you need a metric.
2. **Do not mention the solution.** Do not state you analytic solution

method in the business case. The customer may want a model and a model may be the right analytic solution for answering the question, but most customer do not understand what model do and do not. The business case should focus on the question/problem not the solution. If you allow it to, the customer will start building your model for you, without the expertise to do so. I have seen this happen.

3. **Use timing.** Guide the customer to include timing in their business case, i.e., achieve metric A with X months, detect behavior B at least Y months in advance, and so on. With a timing “device” you can help the customer realize the temporal nature of analytic solution. On one hand, the solutions do not work forever, but many of them believe otherwise. On the other hand, solutions apply only to events and behaviors in a specified time period. Behavior farther out from the phenomenon or event are quite different from behavior closer to it.
4. **Tie it to the business unit.** That might sound like a no-brainer, but I have seen business units responsible for web activity or call center activity want to tie the business case to a product. If you are trying to improve call center operations, you are trying to improve call center operations, period. Now, you are probably doing so with product sales as a downstream goal, but you cannot put the cart before the horse. Fix your call center problem and then ask a new question.

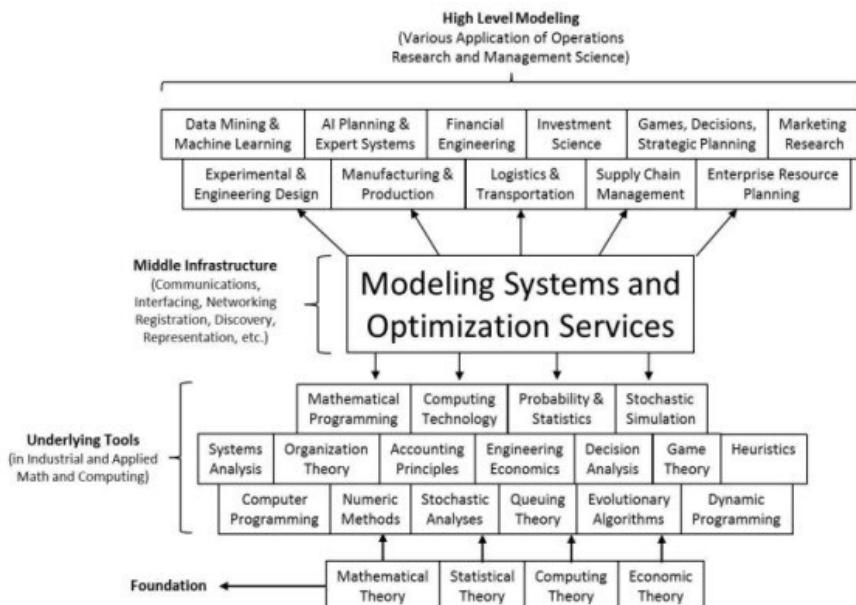
I am sure there is a lot more that can be said about this subject, but this is what I have learned over time and I am just one analyst. The fact remains that if we do not get the “real question”, then we may not provide the “right solution.”

Though there are other interesting pieces of Roske’s Recipe, I will not address here. Perhaps another article?

## 1.5. Modeling Systems and Optimization Services

Modeling Systems and Optimization Services is an interface part that bridges OR modeling with OR tools. When implemented smoothly, it is the part that is not noticed by modelers or users.

- Communications
- Interfacing
- Networking
- Discovery
- Representation



## 1.6. The Operations Research Toolkit

Underlying Tools is the level that is typically regarded as what uniquely defines Operations Research.

- Mathematical Programming
- Computing Technology
- Probability and Statistics
- Stochastic Simulation
- Systems Analysis
- Decision Analysis
- Game Theory
- Heuristics
- Computer Programming
- Numeric Methods
- Stochastic Analysis

- Organization Theory
- Accounting Principles
- Engineering Economics
- Queuing Theory
- Evolutionary Algorithms
- Dynamic Programming

## 1.7. What is *SCILAB* ?



*SCILAB* is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications. This is the software that we will use predominately through this text, although we will also use Excel and R for some applications.

*SCILAB* is released as open source under the CeCILL license (GPL compatible), and is available for download free of charge. *SCILAB* is available under GNU/Linux, Mac OS X and Windows XP/Vista/7/8 (see system requirements). *SCILAB* includes hundreds of mathematical functions. It has a high level programming language allowing access to advanced data structures, 2-D and 3-D graphical functions.

A large number of functionalities is included in *SCILAB*:

- Maths & Simulation. For usual engineering and science applications including mathematical operations and data analysis.
- 2-D & 3-D Visualization. Graphics functions to visualize, annotate and export data and many ways to create and customize various types of plots and charts.
- Optimization. Algorithms to solve constrained and unconstrained continuous and discrete optimization problems.
- Statistics. Tools to perform data analysis and modeling
- Control System Design & Analysis. Standard algorithms and tools for control system study

- Signal Processing. Visualize, analyze and filter signals in time and frequency domains.
- Application Development. Increase *SCILAB* native functionalities and manage data exchanges with external tools.
- Xcos - Hybrid dynamic systems modeler and simulator. Xcos is a graphical editor to design hybrid dynamical systems models. Models can be designed, loaded, saved, compiled and simulated. It is used for modeling mechanical systems, hydraulic circuits, control systems, etc.

## 1.8. What is R?



### 1.8.1. Introduction

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

### 1.8.2. The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and FORTRAN code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

## 1.9. What is LPSolve and what is it not?



The simple answer is, LPSolve is a Mixed Integer Linear Programming (MILP) solver.

LPSolve is a free (see LGPL for the GNU lesser general public license) linear (integer) programming solver based on the revised simplex method and the branch-and-bound method for the integers.

### 1.9.1. It contains full source, examples and manuals.

LPSolve solves pure linear, (mixed) integer/binary, semi-continuous and special ordered sets (SOS) models. Note the word linear. This means that equations must be of the first order, for example  $5x - 3y$ . However  $xy$  is not linear and cannot be handled by LPSolve. Both the objective function and the constraints have this restriction. Also see Ratios.

Via the Branch-and-bound algorithm, it can handle integer variables, semi-continuous variables and Special Ordered Sets (SOS).

LPSolve has no limit on model size and accepts standard both lp or mps input files, but even that can be extended. Note however that some models could give LPSolve a hard time and will even fail to solve. The larger the model the likely the chance for that. But even commercial solvers have problems with that. It can also be called as a library from different languages like C, VB, .NET, Delphi, Excel, Java, ...

It can also be called from AMPL, MATLAB, O-Matrix, SCILAB, Octave, and R via a driver program. LPSolve is written in ANSI C and can be compiled on many different platforms like Linux and WINDOWS.

LPSolve has its own community via the Yahoo group <http://groups.yahoo.com/group/LPSolve/>. There you can find the latest sources, executables for the common platforms, examples, manuals and a message board where people can share their thoughts on LPSolve.

Basically, LPSolve is a library, a set of routines, called the API that can be called from almost any programming language to solve MILP problems. There are several ways to pass the data to the library:

- Via the API
- Via input files
- Via an IDE

### 1.9.2. Using the LPSolve an IDE

Thanks to Henri Gourvest, there is now also an IDE program called LPSolve IDE that uses the API to provide a Windows application to solve models. See LPSolve IDE for its usage. With this program you don't have to know anything of API or computer programming languages. You can just provide your model to the program and it will solve the model and give you the result. LPSolve can be called from many programming language. Among them are C, C++, Pascal, Delphi, Java, VB, C#, VB.NET, Excel. But let this list not be a limitation. Any programming language capable of calling external libraries (DLLs under Windows, Shared libraries (.so) under Unix/Linux) can call LPSolve.

Here is a list of some key features of LPSolve:

- Mixed Integer Linear Programming (MILP) solver
- Basically no limit on model size
- It is free and with sources
- Supports Integer variables, Semi-continuous variables and Special Ordered Sets
- Can read model from MPS, LP or user written format
- Models can be built in-memory without the use of files
- Has a powerful API interface
- Easy callable from other programming languages
- Advanced pricing using Devex and Steepest Edge for both primal

and dual simplexes

- Provides different scaling methods to make the model more numerical stable
- Has presolve capabilities to tighten constraints/make the model smaller and faster to solve
- Has a base crashing routine to determine a starting point
- Allows restart after making changes to the model. Solve continues from the last found solution
- Possible to select desired combinations of primal and dual phases 1 and 2
- Possible to set several solver parameters like tolerances
- Alternative (and faster) inverse/re-factorization libraries are provided for. See Basis Factorization Packages
- Alternative model readers and writers possible via the XLI implementation. See External Language Interfaces
- Has the possibility to convert one model format to another format
- Provides post-optimal sensitivity analysis. See Sensitivity

## 2. What is Linear Algebra?

Most of mathematics came to me with difficulty—Linear Algebra made sense though. In this chapter, we study the topics in linear algebra that will be needed in the rest of the book. We begin by discussing the building blocks of linear algebra: matrices and vectors. Then we use our knowledge of matrices and vectors to develop a systematic procedure (the Gauss–Jordan method) for solving linear equations, which we then use to invert matrices. We close the chapter with an introduction to determinants.

The material covered in this chapter will be used in our study of linear and nonlinear programming.

**DEFINITION.** ■ A matrix is any rectangular array of numbers. ■

For example,

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \end{bmatrix}, [2 \quad 1]$$

Are all matrices.

If a matrix  $A$  has  $m$  rows and  $n$  columns, we call  $A$  an  $m \times n$  matrix. We refer to

$m \times n$  as the order of the matrix. A typical  $m \times n$  matrix  $A$  may be written as

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Definition. The number in the  $i$ th row and  $j$ th column of  $A$  is called the  $ij$ th element of  $A$  and is written  $a_{ij}$ .

For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

then  $a_{11} = 1$ ,  $a_{23} = 6$ , and  $a_{31} = 7$ .

Sometimes we will use the notation  $A = [a_{ij}]$  to indicate that  $A$  is the matrix whose  $ij$ th element is  $a_{ij}$ .

**DEFINITION.** ■ Two matrices  $A = [a_{ij}]$  and  $B = [b_{ij}]$  are equal if and only if  $A$  and  $B$  are of the same order and for all  $i$  and  $j$ ,  $a_{ij} = b_{ij}$ . ■

For example, if

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} x & y \\ w & z \end{bmatrix}$$

then  $A = B$  if and only if  $x = 1$ ,  $y = 2$ ,  $w = 3$ , and  $z = 4$ .

## 2.1. Vectors

Any matrix with only one column (that is, any  $m \times 1$  matrix) may be thought of as a column vector. The number of rows in a column vector is the dimension of the column vector. Thus,

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

may be thought of as a  $2 \times 1$  matrix or a two-dimensional column vector.  $R_m$  will denote the set of all  $m$ -dimensional column vectors.

In analogous fashion, we can think of any vector with only one row (a  $1 \times n$  matrix) as a row vector. The dimension of a row vector is the number of columns in the vector. Thus,  $[9 \ 2 \ 3]$  may be viewed as a  $1 \times 3$  matrix or a three-dimensional row vector. In this book, vectors appear in boldface type: for instance, vector  $\mathbf{v}$ . An  $m$ -dimensional vector (either row or column) in which all elements equal zero is called a zero vector (written  $\mathbf{0}$ ). Thus,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } [0 \ 0]$$

are two-dimensional zero vectors.

Any  $m$ -dimensional vector corresponds to a directed line segment in the  $m$ -dimensional plane. For example, in the two-dimensional plane, the vector

$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

corresponds to the line segment joining the point

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

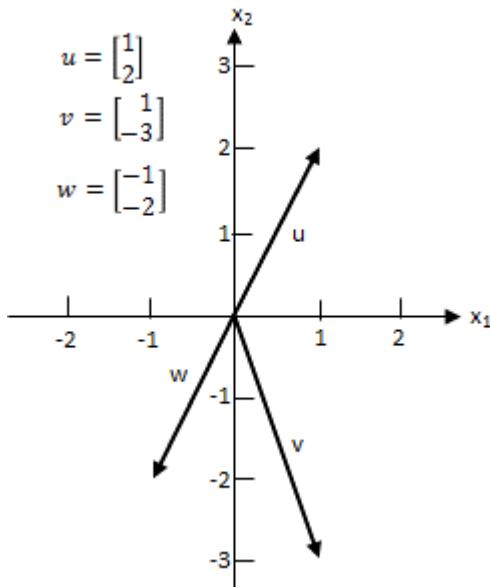
To the point

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

The directed line segments corresponding to

$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 1 \\ -3 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

are drawn in Figure 1.



**FIGURE 1.** Vectors Are Directed Line Segments

## 2.2. The Scalar Product of Two Vectors

An important result of multiplying two vectors is the scalar product. To define the scalar product of two vectors, suppose we have a row vector  $\mathbf{u} = [u_1 \ u_2 \ \cdots \ u_n]$  and a column vector

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

of the same dimension. The scalar product of  $\boldsymbol{u}$  and  $\boldsymbol{v}$  (written  $\boldsymbol{u} \cdot \boldsymbol{v}$ ) is the number  $u_1v_1 + u_2v_2 + \cdots + u_nv_n$ .

For the scalar product of two vectors to be defined, the first vector must be a row vector and the second vector must be a column vector. For example, if

$$\boldsymbol{u} = [1 \ 2 \ 3] \text{ and } \boldsymbol{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

then  $\boldsymbol{u} \cdot \boldsymbol{v} = 1(2) + 2(1) + 3(2) = 10$ . By these rules for computing a scalar product, if

$$\boldsymbol{u} = [1 \ 2] \text{ and } \boldsymbol{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

then  $\boldsymbol{u} \cdot \boldsymbol{v}$  is not defined. Also, if

$$\boldsymbol{u} = [1 \ 2 \ 3] \text{ and } \boldsymbol{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

then  $\boldsymbol{u} \cdot \boldsymbol{v}$  is not defined because the vectors are of two different dimensions.

Note that two vectors are perpendicular if and only if their scalar product equals 0. Thus, the vectors  $[1 \ -1]$  and  $[1 \ 1]$  are perpendicular.

We note that  $\boldsymbol{u} \cdot \boldsymbol{v} = |\boldsymbol{u}| |\boldsymbol{v}| \cos u$ , where  $|\boldsymbol{u}|$  is the length of the vector  $\boldsymbol{u}$  and  $u$  is the angle between the vectors  $\boldsymbol{u}$  and  $\boldsymbol{v}$ .

## 2.3. Matrix Operations

We now describe the arithmetic operations on matrices that are used later in this book.

### 2.3.1. The Scalar Multiple of a Matrix

Given any matrix  $A$  and any number  $c$  (a number is sometimes referred to as a scalar), the matrix  $cA$  is obtained from the matrix  $A$  by multiplying each element of  $A$  by  $c$ . For example, if

$$A = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix} \text{ then } 3A = \begin{bmatrix} 6 & 3 \\ -3 & 0 \end{bmatrix}$$

For  $c = -1$ , scalar multiplication of the matrix  $A$  is sometimes written as  $-A$ .

### 2.3.2. Addition of Two Matrices

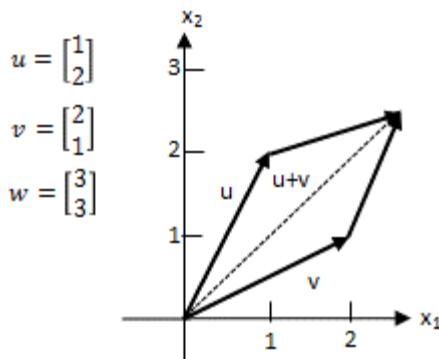
Let  $A = [a_{ij}]$  and  $B = [b_{ij}]$  be two matrices with the same order (say,  $m \times n$ ). Then the matrix  $C = A + B$  is defined to be the  $m \times n$  matrix whose ijth element is  $a_{ij} + b_{ij}$ . Thus, to obtain the sum of two matrices  $A$  and  $B$ , we add the corresponding elements of  $A$  and  $B$ . For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} -1 & -1 & -2 \\ 2 & 1 & -1 \end{bmatrix}$$

then

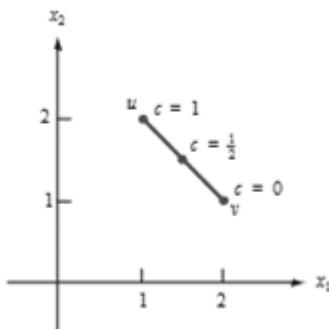
$$A + B = \begin{bmatrix} 1 - 1 & 2 - 1 & 3 - 2 \\ 0 + 2 & -1 + 1 & 1 - 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 0 & 0 \end{bmatrix}$$

This rule for matrix addition may be used to add vectors of the same dimension. For example, if  $\mathbf{u} = [1 \ 2]$  and  $\mathbf{v} = [2 \ 1]$ , then  $\mathbf{u} + \mathbf{v} = [1 + 2 \ 2 + 1] = [3 \ 3]$ . Vectors may be added geometrically by the parallelogram law (see Figure 2).

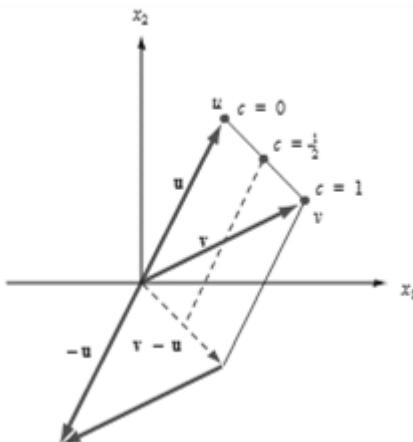


**FIGURE 2.** Addition of Vectors

We can use scalar multiplication and the addition of matrices to define the concept of a line segment. A glance at Figure 1 should convince you that any point  $u$  in the  $m$ -dimensional plane corresponds to the  $m$ -dimensional vector  $\mathbf{u}$  formed by joining the origin to the point  $u$ . For any two points  $u$  and  $v$  in the  $m$ -dimensional plane, the line segment joining  $u$  and  $v$  (called the line segment  $uv$ ) is the set of all points in the  $m$ -dimensional plane that correspond to the vectors  $c\mathbf{u} + (1 - c)\mathbf{v}$ , where  $0 < c < 1$  (Figure 3). For example, if  $u = (1, 2)$  and  $v = (2, 1)$ , then the line segment  $uv$  consists of the points corresponding to the vectors  $c[1 \ 2] + (1 - c)[2 \ 1] = [2 - c \ 1 + c]$ , where  $0 < c < 1$ . For  $c = 0$  and  $c = 1$ , we obtain the endpoints of the line segment  $uv$ ; for  $c = 1/2$ , we obtain the midpoint ( $0.5\mathbf{u} + 0.5\mathbf{v}$ ) of the line segment  $uv$ .



**FIGURE 3.** Line Segment Joining  $u = (1, 2)$  and  $v = (2, 1)$



**FIGURE 4.** Representation of Line Segment  $uv$

Using the parallelogram law, the line segment  $uv$  may also be viewed as the points corresponding to the vectors  $\mathbf{u} + c(\mathbf{v} - \mathbf{u})$ , where  $0 < c < 1$  (Figure 4). Observe that for  $c = 0$ , we obtain the vector  $\mathbf{u}$  (corresponding to point  $u$ ), and for  $c = 1$ , we obtain the vector  $\mathbf{v}$  (corresponding to point  $v$ ).

### 2.3.3. The Transpose of a Matrix

Given any  $m \times n$  matrix

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

the transpose of  $A$  (written  $A^T$ ) is the  $n \times m$  matrix

$$\begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}$$

Thus,  $A^T$  is obtained from  $A$  by letting row 1 of  $A$  be column 1 of  $A^T$ , letting row 2 of  $A$

be column 2 of  $A^T$ , and so on. For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ then } A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Observe that  $(A^T)^T = A$ . Let  $B = [1 \ 2]$ ; then

$$B^T = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ and } (B^T)^T = [1 \ 2] = B$$

As indicated by these two examples, for any matrix  $A$ ,  $(A^T)^T = A$ .

### 2.3.4. Matrix Multiplication

Given two matrices  $A$  and  $B$ , the matrix product of  $A$  and  $B$  (written  $AB$ ) is defined if and only if

$$\text{Number of columns in } A = \text{number of rows in } B \quad (2.1)$$

For the moment, assume that for some positive integer  $r$ ,  $A$  has  $r$  columns and  $B$  has  $r$  rows. Then for some  $m$  and  $n$ ,  $A$  is an  $m \times r$  matrix and  $B$  is an  $r \times n$  matrix.

**DEFINITION.** ■ The matrix product  $C = AB$  of  $A$  and  $B$  is the  $m \times n$  matrix  $C$  whose  $ij$ th element is determined as follows:  $ij$ th element of  $C = \text{scalar product of row } i \text{ of } A \times \text{column } j \text{ of } B$  ■

(2)

If Equation (2.1) is satisfied, then each row of  $A$  and each column of  $B$  will have the same number of elements. Also, if (2.1) is satisfied, then the scalar product in Equation (2.2) will be defined. The product matrix  $C = AB$  will have the same number of rows as  $A$  and the same number of columns as  $B$ .

**EXAMPLE 1.** Compute  $C = AB$  for

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} \text{ and } B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 1 & 2 \end{bmatrix}$$

**Solution.** Because  $A$  is a  $2 \times 3$  matrix and  $B$  is a  $3 \times 2$  matrix,  $AB$  is defined, and  $C$  will be a

$2 \times 2$  matrix. From Equation (2),

$$a_{11} = [1 \ 1 \ 2] \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = 1(1) + 1(2) + 2(1) = 5$$

$$a_{12} = [1 \ 1 \ 2] \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = 1(1) + 1(3) + 2(2) = 8$$

$$a_{21} = [2 \ 1 \ 3] \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = 2(1) + 1(2) + 3(1) = 7$$

$$a_{22} = [2 \ 1 \ 3] \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = 2(1) + 1(3) + 3(2) = 11$$

$$C = AB = \begin{bmatrix} 5 & 8 \\ 7 & 11 \end{bmatrix}$$

### EXAMPLE 2. Column Vector Times Row Vector

Find  $AB$  for

$$A = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ and } B = [3 \quad 4]$$

**Solution.** Because  $A$  has one column and  $B$  has one row,  $C = AB$  will exist. From Equation (2.2), we know that  $C$  is a  $2 \times 2$  matrix with

$$\begin{aligned} a_{11} &= 1(3) = 3 & a_{12} &= 1(4) = 4 \\ a_{21} &= 2(3) = 6 & a_{22} &= 2(4) = 8 \end{aligned}$$

Thus

$$C = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$$

### EXAMPLE 3. Row Vector Times Column Vector

Compute  $D = BA$  for the  $A$  and  $B$  of Example 2.

**Solution.** In this case,  $D$  will be a  $1 \times 1$  matrix (or a scalar). From Equation (2),

$$d_{11} = 3(1) + 4(2) = 11$$

Thus,  $D = [11]$ . In this example, matrix multiplication is equivalent to scalar multiplication of a row and column vector.

### EXAMPLE 4. Undefined Matrix Product

Show that  $AB$  is undefined if

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 2 \end{bmatrix}$$

**Solution.** This follows because  $A$  has two columns and  $B$  has three rows. Thus, Equation (2.1) is not satisfied.

**TABLE 1.** Gallons of Crude Oil Required to Produce 1 Gallon of Gasoline

| Crude Oil | Premium Unleaded | Regular<br>Unleaded | Regular Leaded |
|-----------|------------------|---------------------|----------------|
| 1         | $\frac{3}{4}$    | $\frac{2}{3}$       | $\frac{1}{4}$  |
| 2         | $\frac{1}{4}$    | $\frac{1}{3}$       | $\frac{3}{4}$  |

Many computations that commonly occur in operations research (and other branches of mathematics) can be concisely expressed by using matrix multiplication. To illustrate this, suppose an oil company manufactures three types of gasoline: premium unleaded, regular unleaded, and regular leaded. These gasolines are produced by mixing two types of crude oil: crude oil 1 and crude oil 2. The number of gallons of crude oil required to manufacture 1 gallon of gasoline is given in Table 1.

From this information, we can find the amount of each type of crude oil needed to manufacture a given amount of gasoline. For example, if the company wants to produce

10 gallons of premium unleaded, 6 gallons of regular unleaded, and 5 gallons of regular leaded, then the company's crude oil requirements would be

$$\text{Crude 1 required} = \left(\frac{3}{4}\right)(10) + \left(\frac{2}{3}\right)(6) + \left(\frac{1}{4}\right)5 = 12.75 \text{ gallons}$$

$$\text{Crude 2 required} = \left(\frac{1}{4}\right)(10) + \left(\frac{1}{3}\right)(6) + \left(\frac{3}{4}\right)5 = 8.25 \text{ gallons}$$

More generally, we define

$$p_U = \text{gallons of premium unleaded produced}$$

$$r_U = \text{gallons of regular unleaded produced}$$

$$r_L = \text{gallons of regular leaded produced}$$

$$c_1 = \text{gallons of crude 1 required}$$

$$c_2 = \text{gallons of crude 2 required}$$

Then the relationship between these variables may be expressed by

$$c_1 = \left(\frac{3}{4}\right)p_U + \left(\frac{2}{3}\right)r_U + \left(\frac{1}{4}\right)r_L$$

$$c_2 = \left(\frac{1}{4}\right)p_U + \left(\frac{1}{3}\right)r_U + \left(\frac{3}{4}\right)r_L$$

### 2.3.5. Properties of Matrix Multiplication

To close this section, we discuss some important properties of matrix multiplication. In what follows, we assume that all matrix products are defined.

1. Row  $i$  of  $AB = (\text{row } i \text{ of } A)B$ . To illustrate this property, let

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} \text{ and } B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 1 & 2 \end{bmatrix}$$

Then row 2 of the  $2 \times 2$  matrix  $AB$  is equal to

$$\begin{bmatrix} 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 1 & 2 \end{bmatrix} = [7 \quad 11]$$

This answer agrees with Example 1.

2. Column  $j$  of  $AB = A(\text{column } j \text{ of } B)$ . Thus, for  $A$  and  $B$  as given, the first column of  $AB$  is

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

Properties 1 and 2 are helpful when you need to compute only part of the matrix  $AB$ .

3. Matrix multiplication is associative. That is,  $A(BC) = (AB)C$ . To illustrate, let

$$A = [1 \quad 2], \quad B = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}, \quad C = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Then  $AB = [10 \quad 13]$  and  $(AB)C = 10(2) + 13(1) = [33]$ .

On the other hand,

$$BC = \begin{bmatrix} 7 \\ 13 \end{bmatrix}$$

so  $A(BC) = 1(7) + 2(13) = [33]$ . In this case,  $A(BC) = (AB)C$  does hold.

4. Matrix multiplication is distributive. That is,  $A(B + C) = AB + AC$  and  $(B + C)D = BD + CD$ .

## 2.4. Matrix Multiplication with Excel

Using the Excel MMULT function, it is easy to multiply matrices. To illustrate, let's use Excel to find the matrix product  $AB$  that we found in Example 1 (see Figure 5 and file Mmult.xls). We proceed as follows:

|    | A                            | B | C | D | E  | F |
|----|------------------------------|---|---|---|----|---|
| 1  | <b>Matrix Multiplication</b> |   |   |   |    |   |
| 2  |                              |   |   |   |    |   |
| 3  | A=                           |   |   | 1 | 1  | 2 |
| 4  |                              |   |   | 2 | 1  | 3 |
| 5  |                              |   |   |   |    |   |
| 6  | B=                           |   |   | 1 | 1  |   |
| 7  |                              |   |   | 2 | 3  |   |
| 8  |                              |   |   | 1 | 2  |   |
| 9  |                              |   |   |   |    |   |
| 10 | C=                           |   |   | 5 | 8  |   |
| 11 |                              |   |   | 7 | 11 |   |
| 12 |                              |   |   |   |    |   |

**Step 1.** Enter A and B in D2:F3 and D5:E7, respectively.

**Step 2.** Select the range (D9:E10) in which the product AB will be computed.

**Step 3.** In the upper left-hand corner (D9) of the selected range, type the formula

= MMULT(D2:F3,D5:E7)

Then hit **Control + Shift + Enter** (not just Enter), and the desired matrix product will be computed. Note that MMULT is an array function and not an ordinary spreadsheet function. This explains why we must preselect the range for  $AB$  and use Control Shift Enter.

## 2.5. Matrices and Systems of Linear Equations

Consider a system of linear equations given by

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_n \end{aligned} \tag{2.3}$$

In Equation (2.3),  $x_1, x_2, \dots, x_n$  are referred to as variables, or unknowns, and the  $a_{ij}$ 's and  $b_i$ 's are constants. A set of equations such as (2.3) is called a linear system of  $m$  equations in  $n$  variables.

**DEFINITION** ■ A solution to a linear system of  $m$  equations in  $n$  unknowns is a set of values for the unknowns that satisfies each of the system's  $m$  equations. ■

To understand linear programming, we need to know a great deal about the properties of solutions to linear equation systems. With this in mind, we will devote much effort to studying such systems.

We denote a possible solution to Equation (2.3) by an  $n$ -dimensional column vector  $\mathbf{x}$ , in which the  $i$ th element of  $\mathbf{x}$  is the value of  $x_i$ . The following example illustrates the concept of a solution to a linear system.

### EXAMPLE 5. Solution to Linear System

Show that

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

is a solution to the linear system

$$x_1 + 2x_2 = 5$$

$$2x_1 - x_2 = 0$$

and that

$$\mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Is not a solution to linear system (2.3).

**Solution.** To show that

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

is a solution to Equation (4), we substitute  $x_1 = 1$  and  $x_2 = 2$  in both equations and check that they are satisfied:  $1 + 2(2) = 5$  and  $2(1) - 2 = 0$ .

The vector

$$\mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

is not a solution to (4), because  $x_1 = 3$  and  $x_2 = 1$  fail to satisfy  $2x_1 - x_2 = 0$ .

Using matrices can greatly simplify the statement and solution of a system of linear equations. To show how matrices can be used to compactly represent Equation (2.3), let

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Then (2.3) may be written as

$$A\mathbf{x} = \mathbf{b} \quad (2.4)$$

Observe that both sides of Equation (2.4) will be  $m \times 1$  matrices (or  $m \times 1$  column vectors). For the matrix  $Ax$  to equal the matrix  $b$  (or for the vector  $Ax$  to equal the vector  $b$ ), their corresponding elements must be equal. The first element of  $Ax$  is the scalar product of row 1 of  $A$  with  $x$ . This may be written as

$$[a_{11} \quad a_{12} \quad \cdots \quad a_{1n}] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \quad (2.5)$$

This must equal the first element of  $b$  (which is  $b_1$ ). Thus, (2.5) implies that  $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$ . This is the first equation of (2.3). Similarly, (2.5) implies that the scalar product of row  $i$  of  $A$  with  $x$  must equal  $b_i$ , and this is just the  $i$ th equation of (2.3). Our discussion shows that (2.3) and (2.5) are two different ways of writing the same linear system. We call (2.5) the matrix representation of (2.3). For example, the **matrix representation** of (2.4) is

$$\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

Sometimes we abbreviate (2.5) by writing

$$Ab \quad (2.6)$$

If  $A$  is an  $m \times n$  matrix, it is assumed that the variables in (2.6) are  $x_1, x_2, \dots, x_n$ . Then

(2.6) is still another representation of (2.3). For instance, the matrix

$$\left[ \begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{array} \right]$$

represents the system of equations

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 2 \\ x_2 + 2x_3 &= 3 \\ x_1 + x_2 + x_3 &= 1 \end{aligned}$$

$$(2.7)$$

## 2.6. The Gauss–Jordan Method for Solving Systems of Linear Equations

We develop in this section an efficient method (the Gauss–Jordan method) for solving a system of linear equations. Using the Gauss–Jordan method, we show that any system of linear equations must satisfy one of the following three cases:

**Case 1** The system has no solution.

**Case 2** The system has a unique solution.

**Case 3** The system has an infinite number of solutions.

The Gauss–Jordan method is also important because many of the manipulations used in this method are used when solving linear programming problems by the simplex algorithm (see Chapter 4).

### 2.6.1. Elementary Row Operations

Before studying the Gauss–Jordan method, we need to define the concept of an **elementary row operation** (ERO). An ERO transforms a given matrix  $A$  into a new matrix  $A'$  via one of the following operations.

**Type 1 ERO.**  $A'$  is obtained by multiplying any row of  $A$  by a nonzero scalar. For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 5 & 6 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

then a Type 1 ERO that multiplies row 2 of  $A$  by 3 would yield

$$A' = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 9 & 15 & 18 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

**Type 2 ERO.** Begin by multiplying any row of  $A$  (say, row  $i$ ) by a nonzero scalar  $c$ . For some  $j \neq i$ , let row  $j$  of  $A' = c(\text{row } i \text{ of } A) + \text{row } j \text{ of } A$ , and let the other rows of  $A'$  be the same as the rows of  $A$ .

For example, we might multiply row 2 of  $A$  by 4 and replace row 3 of  $A$  by  $4(\text{row 2 of } A) + \text{row 3 of } A$ . Then row 3 of  $A'$  becomes

$$4[1 \ 3 \ 5 \ 6] + [0 \ 1 \ 2 \ 3] = [4 \ 13 \ 22 \ 27]$$

and

$$A' = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 5 & 6 \\ 4 & 13 & 22 & 27 \end{bmatrix}$$

**Type 3 ERO.** Interchange any two rows of  $A$ . For instance, if we interchange rows 1 and 3 of  $A$ , we obtain

$$A' = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 3 & 5 & 6 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

Type 1 and Type 2 EROs formalize the operations used to solve a linear equation system. To solve the system of equations

$$\begin{aligned} x_1 + x_2 &= 2 \\ 2x_1 + 4x_2 &= 7 \end{aligned} \tag{2.8}$$

we might proceed as follows. First replace the second equation in (2.7) by  $-2$ (first equation in (2.8)) + second equation in (2.8). This yields the following linear system:

$$\begin{aligned} x_1 + x_2 &= 2 \\ 2x_2 &= 3 \end{aligned} \tag{2.9}$$

Then multiply the second equation in (2.9) by  $\frac{1}{2}$ , yielding the system

$$\begin{aligned} x_1 + x_2 &= 2 \\ x_2 &= \frac{3}{2} \end{aligned} \tag{2.10}$$

Finally, replace the first equation in (2.10) by  $-1$ [second equation in (2.10)] + first equation in (2.10). This yields the system

$$x_1 = \frac{1}{2}$$

$$x_2 = \frac{3}{2}$$

(2.11)

System (2.11) has the unique solution  $x_1 = \frac{1}{2}$  and  $x_2 = \frac{3}{2}$ . The systems (2.8), (2.9), (2.10), and (2.11) are equivalent in that they have the same set of solutions. This means that  $x_1 = \frac{1}{2}$  and  $x_2 = \frac{3}{2}$  is also the unique solution to the original system, (2.8).

If we view (2.8) in the augmented matrix form  $(A|\mathbf{b})$ , we see that the steps used to solve (2.8) may be seen as Type 1 and Type 2 EROs applied to  $A\mathbf{b}$ . Begin with the augmented matrix version of (2.8):

$$\left[ \begin{array}{cc|c} 1 & 1 & 2 \\ 2 & 4 & 7 \end{array} \right] \quad (2.8')$$

Now perform a Type 2 ERO by replacing row 2 of (2.8') by  $-2(\text{row 1 of } (2.8')) + \text{row 2 of } (2.8')$ . The result is

$$\left[ \begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 2 & 3 \end{array} \right] \quad (2.9')$$

which corresponds to (2.9). Next, we multiply row 2 of (2.9') by  $\frac{1}{2}$  (a Type 1 ERO), resulting in

$$\left[ \begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 3/2 \end{array} \right] \quad (2.10')$$

which corresponds to (2.10). Finally, perform a Type 2 ERO by replacing row 1 of (2.10') by  $-1(\text{row 2 of } (2.10')) + \text{row 1 of } (2.10')$ . The result is

$$\left[ \begin{array}{cc|c} 1 & 0 & 1/2 \\ 0 & 1 & 3/2 \end{array} \right] \quad (2.11')$$

which corresponds to (2.11). Translating (2.11') back into a linear system, we obtain the system  $x_1 = \frac{1}{2}$  and  $x_2 = \frac{3}{2}$ , which is identical to (2.11).

### 2.6.2. Finding a Solution by the Gauss–Jordan Method

The discussion in the previous section indicates that if the matrix  $A'\mathbf{b}'$  is obtained from  $A\mathbf{b}$  via an ERO, the systems  $A\mathbf{x} = \mathbf{b}$  and  $A'\mathbf{x} = \mathbf{b}'$  are

equivalent. Thus, any sequence of EROs performed on the augmented matrix  $A\mathbf{b}$  corresponding to the system  $A\mathbf{x} = \mathbf{b}$  will yield an equivalent linear system.

The Gauss–Jordan method solves a linear equation system by utilizing EROs in a systematic fashion. We illustrate the method by finding the solution to the following linear system:

$$\begin{aligned} 2x_1 + 2x_2 + x_3 &= 9 \\ 2x_1 - x_2 + 2x_3 &= 6 \\ x_1 - x_2 + 2x_3 &= 5 \end{aligned} \tag{2.12}$$

The augmented matrix representation is

$$\left[ \begin{array}{ccc|c} 2 & 2 & 1 & 9 \\ 2 & -1 & 2 & 6 \\ 1 & -1 & 2 & 5 \end{array} \right] \tag{2.12'}$$

Suppose that by performing a sequence of EROs on (2.12') we could transform (2.12') into

$$\left[ \begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right] \tag{2.13'}$$

We note that the result obtained by performing an ERO on a system of equations can also be obtained by multiplying both sides of the matrix representation of the system of equations by a particular matrix. This explains why EROs do not change the set of solutions to a system of equations.

Matrix (2.13') corresponds to the following linear system:

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \\ x_3 &= 3 \end{aligned} \tag{2.13}$$

System (2.12) has the unique solution  $x_1 = 1, x_2 = 2, x_3 = 3$ . Because (2.13') was obtained from (2.12') by a sequence of EROs, we know that

(2.12) and (2.13) are equivalent linear systems. Thus,  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 3$  must also be the unique solution to (2.12). We now show how we can use EROs to transform a relatively complicated system such as (2.12) into a relatively simple system like (2.13). This is the essence of the Gauss–Jordan method.

We begin by using EROs to transform the first column of (2.12') into

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Then we use EROs to transform the second column of the resulting matrix into

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Finally, we use EROs to transform the third column of the resulting matrix into

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

As a final result, we will have obtained (2.13'). We now use the Gauss–Jordan method to solve (2.12). We begin by using a Type 1 ERO to change the element of (2.12') in the first row and first column into a 1. Then we add multiples of row 1 to row 2 and then to row 3 (these are Type 2 EROs). The purpose of these Type 2 EROs is to put zeroes in the rest of the first column. The following sequence of EROs will accomplish these goals.

**Step 1.** Multiply row 1 of (2.12') by  $\frac{1}{2}$ . This Type 1 ERO yields

$$A_1 | \mathbf{b}_1 = \left[ \begin{array}{ccc|c} 1 & 0 & \frac{1}{2} & \frac{9}{2} \\ 2 & -1 & 2 & 6 \\ 1 & -1 & 2 & 5 \end{array} \right]$$

**Step 2.** Replace row 2 of  $A_1|\mathbf{b}_1$  by  $-2(\text{row 1 of } A_1|\mathbf{b}_1) + \text{row 2 of } A_1|\mathbf{b}_1$ .  
The result of this Type 2 ERO is

$$A_2|\mathbf{b}_2 = \left[ \begin{array}{ccc|c} 1 & 0 & \frac{1}{2} & \frac{9}{2} \\ 0 & -3 & 1 & -3 \\ 1 & -1 & 2 & 5 \end{array} \right]$$

**Step 3.** Replace row 3 of  $A_2|\mathbf{b}_2$  by  $-1(\text{row 1 of } A_2|\mathbf{b}_2 + \text{row 3 of } A_2|\mathbf{b}_2)$ .  
The result of this

Type 2 ERO is

$$A_3|\mathbf{b}_3 = \left[ \begin{array}{ccc|c} 1 & 0 & \frac{1}{2} & \frac{9}{2} \\ 0 & -3 & 1 & -3 \\ 0 & -2 & \frac{3}{2} & \frac{1}{2} \end{array} \right]$$

The first column of (8') has now been transformed into

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

By our procedure, we have made sure that the variable  $x_1$  occurs in only a single equation and in that equation has a coefficient of 1. We now transform the second column of  $A_3|\mathbf{b}_3$  into

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

We begin by using a Type 1 ERO to create a 1 in row 2 and column 2 of  $A_3|\mathbf{b}_3$ . Then we use the resulting row 2 to perform the Type 2 EROs that are needed to put zeros in the rest of column 2. Steps 4–6 accomplish these goals.

**Step 4.** Multiply row 2 of  $A_3|\mathbf{b}_3$  by  $-\frac{1}{3}$ . The result of this Type 1 ERO is

$$A_4|\mathbf{b}_4 = \left[ \begin{array}{ccc|c} 1 & 1 & \frac{1}{2} & \frac{9}{2} \\ 0 & 1 & -\frac{1}{3} & 1 \\ 0 & -2 & \frac{3}{2} & \frac{1}{2} \end{array} \right]$$

**Step 5.** Replace row 1 of  $A_4|\mathbf{b}_4$  by  $-1(\text{row 2 of } A_4|\mathbf{b}_4) + \text{row 1 of } A_4|\mathbf{b}_4$ .  
The result of this Type 2 ERO is

$$A_5|\mathbf{b}_5 = \left[ \begin{array}{ccc|c} 1 & 0 & \frac{5}{6} & \frac{7}{2} \\ 0 & 1 & -\frac{1}{3} & 1 \\ 0 & -2 & \frac{3}{2} & \frac{1}{2} \end{array} \right]$$

**Step 6.** Replace row 3 of  $A_5|\mathbf{b}_5$  by  $2(\text{row 2 of } A_5|\mathbf{b}_5) + \text{row 3 of } A_5|\mathbf{b}_5$ .  
The result of this Type 2 ERO is

$$A_6|\mathbf{b}_6 = \left[ \begin{array}{ccc|c} 1 & 0 & \frac{5}{6} & \frac{7}{2} \\ 0 & 1 & -\frac{1}{3} & 1 \\ 0 & 0 & \frac{5}{6} & \frac{5}{2} \end{array} \right]$$

Column 2 has now been transformed into

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Observe that our transformation of column 2 did not change column 1.

To complete the Gauss–Jordan procedure, we must transform the third column of  $A_6|\mathbf{b}_6$  into

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

We first use a Type 1 ERO to create a 1 in the third row and third column of  $A_6|\mathbf{b}_6$ . Then we use Type 2 EROs to put zeros in the rest of column 3. Steps 7–9 accomplish these goals.

**Step 7.** Multiply row 3 of  $A_6|\mathbf{b}_6$  by  $\frac{6}{5}$ . The result of this Type 1 ERO is

$$A_7|\mathbf{b}_7 = \left[ \begin{array}{ccc|c} 1 & 0 & \frac{5}{6} & \frac{7}{2} \\ 0 & 1 & -\frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 1 & 3 \end{array} \right]$$

**Step 8.** Replace row 1 of  $A_7|\mathbf{b}_7$  by  $-\frac{5}{6}$  (row 3 of  $A_7|\mathbf{b}_7$ ) + row 1 of  $A_7|\mathbf{b}_7$ . The result of this Type 2 ERO is

$$A_8|\mathbf{b}_8 = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & -\frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 1 & 3 \end{array} \right]$$

**Step 9.** Replace row 2 of  $A_8|\mathbf{b}_8$  by  $\frac{1}{3}$  (row 3 of  $A_8|\mathbf{b}_8$ ) + row 2 of  $A_8|\mathbf{b}_8$ .

The result of this Type 2 ERO is

$$A_9|\mathbf{b}_9 = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right]$$

$A_9|\mathbf{b}_9$  represents the system of equations

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \\ x_3 &= 3 \end{aligned} \tag{2.13}$$

Thus, (2.13) has the unique solution  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 3$ . Because (2.13) was obtained from

(2.12) via EROs, the unique solution to (2.12) must also be  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 3$ .

The reader might be wondering why we defined Type 3 EROs (interchanging of rows). To see why a Type 3 ERO might be useful, suppose you want to solve

$$\begin{aligned} 2x_2 + x_3 &= 6 \\ x_1 + x_2 - x_3 &= 2 \\ 2x_1 + x_2 + x_3 &= 4 \end{aligned} \tag{2.14}$$

To solve (2.14) by the Gauss–Jordan method, first form the augmented matrix

$$A|\mathbf{b} = \left[ \begin{array}{ccc|c} 0 & 2 & 1 & 6 \\ 1 & 1 & -1 & 2 \\ 2 & 1 & 1 & 4 \end{array} \right]$$

The 0 in row 1 and column 1 means that a Type 1 ERO cannot be used to create a 1 in row 1 and column 1. If, however, we interchange rows 1 and 2 (a Type 3 ERO), we obtain

$$\left[ \begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 2 & 1 & 6 \\ 2 & 1 & 1 & 4 \end{array} \right]$$

Now we may proceed as usual with the Gauss–Jordan method.

### 2.6.3. Special Cases: No Solution or an Infinite Number of Solutions

Some linear systems have no solution, and some have an infinite number of solutions. The following two examples illustrate how the Gauss–Jordan method can be used to recognize these cases.

#### **EXAMPLE 6.** Linear System with No Solution

Find all solutions to the following linear system:

$$\begin{aligned} x_1 + 2x_2 &= 3 \\ 2x_1 + 4x_2 &= 4 \end{aligned} \tag{2.15}$$

**Solution.** We apply the Gauss–Jordan method to the matrix

$$Ab = \left[ \begin{array}{cc|c} 1 & 2 & 3 \\ 2 & 4 & 4 \end{array} \right]$$

We begin by replacing row 2 of  $Ab$  by  $-2(\text{row 1 of } Ab) + \text{row 2 of } Ab$ . The result of this Type 2 ERO is

$$\left[ \begin{array}{cc|c} 1 & 2 & 3 \\ 0 & 0 & -2 \end{array} \right] \tag{2.16}$$

We would now like to transform the second column of (16) into

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

but this is not possible. System (16) is equivalent to the following system of equations:

$$x_1 + 2x_2 = 3$$

$$0x_1 + 0x_2 = -2 \quad (2.16')$$

Whatever values we give to  $x_1$  and  $x_2$ , the second equation in (2.16') can never be satisfied. Thus, (2.16') has no solution. Because (2.16') was obtained from (2.15) by use of EROs, (2.15) also has no solution.

Example 6 illustrates the following idea: If you apply the Gauss–Jordan method to a linear system and obtain a row of the form  $[0 \ 0 \ \dots \ 0 \ c]$  ( $c \neq 0$ ), then the original linear system has no solution.

### EXAMPLE 7. Linear System with Infinite Number of Solutions

Apply the Gauss–Jordan method to the following linear system:

$$\begin{aligned} x_1 + x_2 &= 1 \\ x_2 + x_3 &= 3 \\ x_1 + 2x_2 + x_3 &= 4 \end{aligned} \quad (2.17)$$

**Solution.** The augmented matrix form of (17) is

$$A\mathbf{b} = \left[ \begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 3 \\ 1 & 2 & 1 & 4 \end{array} \right]$$

We begin by replacing row 3 (because the row 2, column 1 value is already 0) of  $A\mathbf{b}$  by  $-1(\text{row 1 of } A\mathbf{b}) + \text{row 3 of } A\mathbf{b}$ . The result of this Type 2 ERO is

$$A_1\mathbf{b}_1 = \left[ \begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 3 \\ 0 & 1 & 1 & 3 \end{array} \right] \quad (2.18)$$

Next we replace row 1 of  $A_1\mathbf{b}_1$  by  $-1(\text{row 2 of } A_1\mathbf{b}_1) + \text{row 1 of } A_1\mathbf{b}_1$ . The result of this Type 2 ERO is

$$A_2 \mathbf{b}_2 = \left[ \begin{array}{ccc|c} 1 & 0 & -1 & -2 \\ 0 & 1 & 1 & 3 \\ 0 & 1 & 1 & 3 \end{array} \right]$$

Now we replace row 3 of  $A_2 \mathbf{b}_2$  by  $-1(\text{row 2 of } A_2 \mathbf{b}_2) + \text{row 3 of } A_2 \mathbf{b}_2$ .  
The result of this Type 2 ERO is

$$A_3 \mathbf{b}_3 = \left[ \begin{array}{ccc|c} 1 & 0 & -1 & -2 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

We would now like to transform the third column of  $A_3 \mathbf{b}_3$  into

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

but this is not possible. The linear system corresponding to  $A_3 \mathbf{b}_3$  is

$$x_1 - x_3 = -2 \quad (2.18.1)$$

$$x_2 + x_3 = 3 \quad (2.18.2)$$

$$0x_1 + 0x_2 + 0x_3 = 0 \quad (2.18.3)$$

Suppose we assign an arbitrary value  $k$  to  $x_3$ . Then (2.18.1) will be satisfied if  $x_1 - k = -2$ , or  $x_1 = k - 2$ . Similarly, (2.18.2) will be satisfied if  $x_2 + k = 3$ , or  $x_2 = 3 - k$ . Of course, (2.18.3) will be satisfied for any values of  $x_1$ ,  $x_2$ , and  $x_3$ . Thus, for any number  $k$ ,  $x_1 = k - 2$ ,  $x_2 = k - 2$ ,  $x_3 = k$  is a solution to (2.18). Thus, (2.18) has an infinite number of solutions (one for each number  $k$ ). Because (2.18) was obtained from (2.17) via EROs, (2.17) also has an infinite number of solutions. A more formal characterization of linear systems that have an infinite number of solutions will be given after the following summary of the Gauss–Jordan method.

## 2.6.4. Summary of the Gauss–Jordan Method

**Step 1.** To solve  $A\mathbf{x} = \mathbf{b}$ , write down the augmented matrix  $A\mathbf{b}$ .

**Step 2.** At any stage, define a current row, current column, and current entry (the entry in the current row and column). Begin with row 1 as the current row, column 1 as the current column, and  $a_{11}$  as the current entry. (a) If  $a_{11}$  (the current entry) is nonzero, then use EROs to transform column 1 (the current column) to

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Then obtain the new current row, column, and entry by moving down one row and one column to the right, and go to step 3. (b) If  $a_{11}$  (the current entry) equals 0, then do a Type 3 ERO involving the current row and any row that contains a nonzero number in the current column. Use EROs to transform column 1 to

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Then obtain the new current row, column, and entry by moving down one row and one column to the right. Go to step 3. (c) If there are no nonzero numbers in the first column, then obtain a new current column and entry by moving one column to the right. Then go to step 3.

**Step 3.** (a) If the new current entry is nonzero, then use EROs to transform it to 1 and the rest of the current column's entries to 0. When finished, obtain the new current row, column, and entry. If this is impossible, then stop. Otherwise, repeat step 3. (b) If the current entry is 0, then do a Type 3 ERO with the current row and any row that contains a nonzero number in the current column. Then use EROs to transform that current entry to 1 and the rest of the current column's entries to 0. When finished, obtain the new current row, column, and entry. If this is impossible, then stop. Otherwise, repeat step 3. (c) If the current column has no nonzero numbers below the current row, then obtain the new current column and entry, and repeat step 3. If it is impossible, then stop.

This procedure may require “passing over” one or more columns without transforming them (see Problem 8).

**Step 4.** Write down the system of equations  $A'\mathbf{x} = \mathbf{b}'$  that corresponds to the matrix  $A'\mathbf{b}'$  obtained when step 3 is completed. Then  $A'\mathbf{x} = \mathbf{b}'$  will have the same set of solutions as  $A\mathbf{x} = \mathbf{b}$ .

## 2.6.5. Basic Variables and Solutions to Linear Equation Systems

To describe the set of solutions to  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ), we need to define the concepts of basic and nonbasic variables.

**DEFINITION** ■ After the Gauss–Jordan method has been applied to any linear system, a variable that appears with a coefficient of 1 in a single equation and a coefficient of 0 in all other equations is called a **basic variable (BV)**. ■

Any variable that is not a basic variable is called a **nonbasic variable (NBV)**. ■

Let BV be the set of basic variables for  $A'\mathbf{x} = \mathbf{b}'$  and NBV be the set of nonbasic variables for  $A'\mathbf{x} = \mathbf{b}'$ . The character of the solutions to  $A'\mathbf{x} = \mathbf{b}'$  depends on which of the following cases occurs.

**Case 1.**  $A'\mathbf{x} = \mathbf{b}'$  has at least one row of form  $[0 \ 0 \ \cdots \ 0 \ c]$  ( $c \neq 0$ ). Then  $A\mathbf{x} = \mathbf{b}$  has no solution (recall Example 6). As an example of Case 1, suppose that when the Gauss–Jordan method is applied to the system  $A\mathbf{x} = \mathbf{b}$ , the following matrix is obtained:

$$\left[ \begin{array}{cccc|c} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{array} \right]$$

In this case,  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ) has no solution.

**Case 2.** Suppose that Case 1 does not apply and NBV, the set of nonbasic variables, is empty. Then  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ) will have a unique solution. To illustrate this, we recall that in solving

$$2x_1 + 2x_2 + x_3 = 9$$

$$2x_1 - x_2 + 2x_3 = 6$$

$$2x_1 - x_2 + 2x_3 = 5$$

the Gauss–Jordan method yielded

$$A'\mathbf{b}' = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right]$$

In this case,  $BV = \{x_1, x_2, x_3\}$  and  $NBV$  is empty. Then the unique solution to  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ) is  $x_1 = 1, x_2 = 2, x_3 = 3$ .

**Case 3.** Suppose that Case 1 does not apply and  $NBV$  is nonempty. Then  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ) will have an infinite number of solutions. To obtain these, first assign each nonbasic variable an arbitrary value. Then solve for the value of each basic variable in terms of the nonbasic variables. For example, suppose

$$A'\mathbf{b}' = \left[ \begin{array}{ccccc|c} 1 & 0 & 0 & 1 & 1 & 3 \\ 0 & 1 & 0 & 2 & 0 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \quad (2.19)$$

Because Case 1 does not apply, and  $BV = \{x_1, x_2, x_3\}$  and  $NBV = \{x_4, x_5\}$ , we have an example of Case 3:  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ) will have an infinite number of solutions. To see what these solutions look like, write down  $A'\mathbf{x} = \mathbf{b}'$ :

$$x_1 + x_4 + x_5 = 3 \quad (2.19.1)$$

$$x_2 + x_4 = 2 \quad (2.19.2)$$

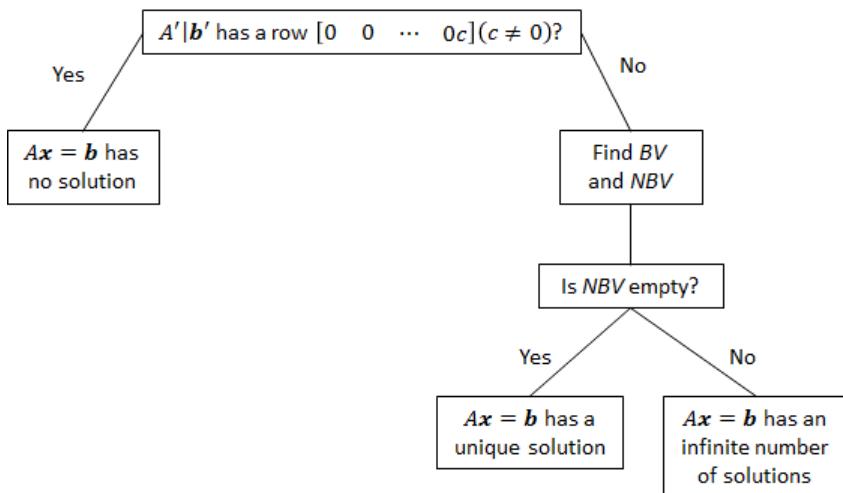
$$x_3 + x_5 = 1 \quad (2.19.3)$$

$$x_1 + x_2 + x_3 + x_4 + x_5 = 0 \quad (2.19.4)$$

Now assign the nonbasic variables ( $x_4$  and  $x_5$ ) arbitrary values  $c$  and  $k$ , with  $x_4 = c$  and  $x_5 = k$ . From (2.19.1), we find that  $x_1 = 3 - c - k$ . From (2.19.2), we find that  $x_2 = 2 - 2c$ . From (2.19.3), we find that  $x_3 = 1 - k$ . Because (2.19.4) holds for all values of the variables,  $x_1 = 3 - c - k, x_2 = 2 - 2c, x_3 = 1 - k, x_4 = c$ , and  $x_5 = k$  will, for any values of  $c$  and  $k$ , be a solution to  $A'\mathbf{x} = \mathbf{b}'$  (and  $A\mathbf{x} = \mathbf{b}$ ).

Our discussion of the Gauss–Jordan method is summarized in Figure 6. We have devoted so much time to the Gauss–Jordan method because, in our study of linear programming, examples of Case 3 (linear systems with an infinite number of solutions) will occur repeatedly. Because the end result of the Gauss–Jordan method must always be one of Cases 1–

3, we have shown that any linear system will have no solution, a unique solution, or an infinite number of solutions.



**FIGURE 6.** Description of Gauss–Jordan Method for Solving Linear Equations

## 2.7. The Inverse of a Matrix

To solve a single linear equation such as  $4x = 3$ , we simply multiply both sides of the equation by the multiplicative inverse of 4, which is  $4^{-1}$ , or  $\frac{1}{4}$ . This yields  $4^{-1}(4x) = (4^{-1})3$  or  $x = \frac{3}{4}$ . (Of course, this method fails to work for the equation  $0x = 3$ , because zero has no multiplicative inverse.) In this section, we develop a generalization of this technique that can be used to solve “square” (number of equations = number of unknowns) linear systems. We begin with some preliminary definitions.

**DEFINITION** ■ A square matrix is any matrix that has an equal number of rows and columns. ■ The diagonal elements of a square matrix are those elements  $a_{ij}$  such that  $i = j$ . ■ A square matrix for which all diagonal elements are equal to 1 and all nondiagonal elements are equal to 0 is called an identity matrix. ■

The  $m \times m$  identity matrix will be written as  $I_m$ . Thus,

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \dots$$

If the multiplications  $I_m A$  and  $A I_m$  are defined, it is easy to show that  $= I_m A = A I_m = A$ . Thus, just as the number 1 serves as the unit element for multiplication of real numbers,  $I_m$  serves as the unit element for multiplication of matrices.

Recall that  $\frac{1}{4}$  is the multiplicative inverse of 4. This is because  $4\left(\frac{1}{4}\right) = \left(\frac{1}{4}\right)4 = 1$ . This motivates the following definition of the inverse of a matrix.

**DEFINITION ■** For a given  $m \times m$  matrix  $A$ , the  $m \times m$  matrix  $B$  is the inverse of  $A$  if

$$BA = AB = I_m \quad (2.20)$$

(It can be shown that if  $BA = I_m$  or  $AB = I_m$ , then the other quantity will also equal  $I_m$ .) ■

Some square matrices do not have inverses. If there does exist an  $m \times m$  matrix  $B$  that satisfies Equation (2.20), then we write  $B = A^{-1}$ . For example, if

$$A = \begin{bmatrix} 2 & 0 & -1 \\ 3 & 1 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

The reader can verify that

$$\begin{bmatrix} 2 & 0 & -1 \\ 3 & 1 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ -5 & 1 & -7 \\ 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And

$$\begin{bmatrix} 1 & 0 & 1 \\ -5 & 1 & -7 \\ 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 & -1 \\ 3 & 1 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus,

$$A^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -5 & 1 & -7 \\ 1 & 0 & 2 \end{bmatrix}$$

To see why we are interested in the concept of a matrix inverse, suppose we want to solve a linear system  $Ax = \mathbf{b}$  that has  $m$  equations and  $m$  unknowns. Suppose that  $A^{-1}$  exists. Multiplying both sides of  $Ax = \mathbf{b}$  by  $A^{-1}$ , we see that any solution of  $Ax = \mathbf{b}$  must also satisfy  $A^{-1}(Ax) = A^{-1}\mathbf{b}$ . Using the associative law and the definition of a matrix inverse, we obtain

$$(A^{-1}A)x = A^{-1}\mathbf{b} \text{ or } I_m = A^{-1}\mathbf{b} \text{ or } x = A^{-1}\mathbf{b}$$

This shows that knowing  $A^{-1}$  enables us to find the unique solution to a square linear system. This is the analog of solving  $4x = 3$  by multiplying both sides of the equation by  $4^{-1}$ .

The Gauss–Jordan method may be used to find  $A^{-1}$  (or to show that  $A^{-1}$  does not exist). To illustrate how we can use the Gauss–Jordan method to invert a matrix, suppose we want to find  $A^{-1}$  for

$$A = \begin{bmatrix} 2 & 5 \\ 1 & 3 \end{bmatrix}$$

This requires that we find a matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = A^{-1}$$

that satisfies

$$\begin{bmatrix} 2 & 5 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.21)$$

From Equation (2.21), we obtain the following pair of simultaneous equations that must be satisfied by  $a$ ,  $b$ ,  $c$ , and  $d$ :

$$\begin{bmatrix} 2 & 5 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \begin{bmatrix} 2 & 5 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Thus, to find

$$\begin{bmatrix} a \\ c \end{bmatrix}$$

(the first column of  $A^{-1}$ ), we can apply the Gauss–Jordan method to the augmented matrix

$$\left[ \begin{array}{cc|c} 2 & 5 & 1 \\ 1 & 3 & 0 \end{array} \right]$$

Once EROs have transformed

to  $I_2$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

will have been transformed into the first column of  $A^{-1}$ . To determine

$$\begin{bmatrix} b \\ d \end{bmatrix}$$

(the second column of  $A^{-1}$ , we apply EROs to the augmented matrix

$$\left[ \begin{array}{cc|c} 2 & 5 & 0 \\ 1 & 3 & 1 \end{array} \right]$$

When

$$\left[ \begin{array}{cc} 2 & 5 \\ 1 & 3 \end{array} \right]$$

has been transformed into  $I_2$ ,

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

will have been transformed into the second column of  $A^{-1}$ . Thus, to find each column of

$A^{-1}$ , we must perform a sequence of EROs that transform

$$\left[ \begin{array}{cc} 2 & 5 \\ 1 & 3 \end{array} \right]$$

into  $I_2$ . This suggests that we can find  $A^{-1}$  by applying EROs to the  $2 \times 4$  matrix

$$A|I_2 = \left[ \begin{array}{cc|cc} 2 & 5 & 1 & 0 \\ 1 & 3 & 0 & 1 \end{array} \right]$$

When

$$\left[ \begin{array}{cc} 2 & 5 \\ 1 & 3 \end{array} \right]$$

Has been transformed into  $I_2$ ,

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

will have been transformed into the first column of  $A^{-1}$ , and

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

will have been transformed into the second column of  $A^{-1}$ . Thus, as  $A$  is transformed into

$I_2, I_2'$  is transformed into  $A^{-1}$ . The computations to determine  $A^{-1}$  follow.

**Step 1.** Multiply row 1 of  $A|I_2$  by 1. This yields

$$A'|I_2' = \left[ \begin{array}{cc|cc} 1 & 5/2 & 1/2 & 0 \\ 1 & 3 & 0 & 1 \end{array} \right]$$

**Step 2.** Replace row 2 of  $A'|I_2'$  by  $-1(\text{row 1 of } A'|I_2') + \text{row 2 of } A'|I_2'$ . This yields

$$A''|I_2'' = \left[ \begin{array}{cc|cc} 1 & 5/2 & 1/2 & 0 \\ 1 & 1/2 & -1/2 & 1 \end{array} \right]$$

**Step 3.** Multiply row 2 of  $A''|I_2''$  by 2. This yields

$$A'''|I_2''' = \left[ \begin{array}{cc|cc} 1 & 5/2 & 1/2 & 0 \\ 0 & 1 & -1 & 2 \end{array} \right]$$

**Step 4.** Replace row 1 of  $A'''|I_2'''$  by  $-5(\text{row 2 of } A'''|I_2''') + \text{row 1 of } A'''|I_2'''$ . This yields

$$\left[ \begin{array}{cc|cc} 1 & 0 & 3 & -5 \\ 0 & 1 & -1 & 2 \end{array} \right]$$

Because  $A$  has been transformed into  $I_2, I_2$  will have been transformed into  $A^{-1}$ . Hence,

$$A^{-1} = \begin{bmatrix} 3 & -5 \\ -1 & 2 \end{bmatrix}$$

The reader should verify that  $AA^{-1} = A^{-1}A = I_2$ .

## 2.8. Basic Linear Algebra in SCILAB

*SCILAB* is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications. Here, we'll make sure that we understand the basic algebraic operations that can be performed on matrices, and how we can use them to solve a set of linear equations.

### 2.8.1. A Note on Notation

The convention used in this lecture and in most linear algebra books is that an italics lower case letter ( $k$ ) denotes a scalar, a bold italics lower case letter ( $\mathbf{x}$ ) denotes a vector, and a capital letter ( $A$ ) denotes a matrix. Typically we name our *SCILAB* variables with a capital letter if they will be used as matrices, and lower case for scalars and vectors.

### 2.8.2. Vector Algebra

In *SCILAB*, a vector is simply a matrix with the size of one dimension equal to 1. We should distinguish between a row vector (a  $1 \times n$  matrix) and a column vector (a  $n \times 1$  matrix). Recall that we change a row vector  $x$  into a column vector using the transpose operator ( $x'$  in *SCILAB*). The same trick works for changing a column vector into a row vector.

```
-->x = [1 2 3 4 5];
-->y = [1 0 2 -1 4];
-->z = [1;3;-1;5;2];

-->x'
ans =
1.
2.
3.
4.
5.

-->z'
ans =
1.    3.    - 1.    5.    2.
```

We can add two vectors,  $x$  and  $y$ , together if they have the same dimensions. The resulting vector  $\mathbf{z} = \mathbf{x} + \mathbf{y}$  is simply an element by element addition of the components of  $x$  and  $y$ :  $z_i = x_i + y_i$ . From this

It follows that vector addition is both commutative and associative, just like regular addition. *SCILAB* also allows you to add a scalar  $k$  (a  $1 \times 1$  matrix) to a vector. The result of  $\mathbf{z} = \mathbf{x} + k$  is the element by element addition  $z_i = k x_i$ .

**x+y**

Vector multiplication can take a few different forms. First of all, if we multiply a scalar  $k$  times a vector  $\mathbf{x}$ , the result is a vector with the same dimension as  $\mathbf{x}$ :  $\mathbf{z} = k\mathbf{x}$  implies  $z_i = kx_i$ . There are two standard ways to multiply two vectors together: the inner product and the outer product.

**2\*x**

The inner product, sometimes called the dot product, is the result of multiplying a row vector times a column vector. The result is a scalar  $z = \sum_i x_i y_i$ . To take the inner product of two column vectors, use  $\mathbf{z} = \mathbf{x}\mathbf{y}^t$ . As we'll see, the orientation of the vectors matters because *SCILAB* treats vectors as matrices.

**x\*y'**

Unlike the inner product, the result of the outer product of two vectors is a matrix. In *SCILAB*, you get the outer product by multiplying a column vector times a row vector:  $Z = \mathbf{x}\mathbf{y}$ . The components of  $Z$  are  $Z_{ij} = x_i y_j$ . To take the outer product of two column vectors, use  $Z = \mathbf{x}^t \mathbf{y}$ .

**x' \* y**

Occasionally, what we really want to do is to multiply two vectors together element by element:  $z_i = x_i y_i$ . *SCILAB* provides the `*` command for this operation:  $\mathbf{z} = \mathbf{x} * \mathbf{y}$ .

To test our understanding, let's try some basic *SCILAB* commands:

```
x = 1:5
y = 6:10
x+y
x+5
5*x x*y' x'*y x.*y
```

How would you initialize the following matrix in *SCILAB* using outer products?

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 2 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

### 2.8.3. Matrix Algebra

The matrix operations are simply generalizations of the vector operations when the matrix has multiple rows *and* columns. You can think of a matrix as a set of row vectors or as a set of column vectors. Define the following matrices in *SCILAB*.

```
A = [1 2 2 -1 4; 1 -1 0 1 2; -1 2 1 3 -2; 2 4 1 -1 3; -1 -1  
0 1 1]  
B = testmatrix('magi',5)  
C = [1 2 2 -1 4; 1 -1 0 1 2; 2 -2 0 2 4; 2 4 1 -1 3; -1 -2  
-2 1 -4]  
I = eye(5,5)  
D = diag(1:5)  
E = rand(4,4)  
F = [1 2 3 4 5; 5 4 3 2 1; -1 0 1 -1 1; 2 4 6 4 2]
```

Matrix addition works element by element, just like vector addition. It is defined for any two matrices of the same size.  $C = A + B$  implies that  $C_{ij} = A_{ij} + B_{ij}$ . Once again, it is both commutative and associative. Scalar multiplication of matrices is also defined as it was with vectors. The result is a matrix:  $C = kA$  implies  $C_{ij} = kA_{ij}$ . Try these matrix additions.

```
A+B  
A+I
```

If you multiply a matrix times a column vector, then the result is another column vector - the column of inner products:  $\mathbf{b} = A\mathbf{x}$  implies  $b_i = \sum_i A_{ij}\mathbf{x}_j$ . Similarly, you can multiply a row vector times a matrix to get a row of inner products:  $\mathbf{b} = \mathbf{x}A$  implies  $b_i = \sum_i A_{ij}\mathbf{x}_j$ . Notice that in both cases, the definitions require that the first variable must have the same number of columns as the second variable has rows. Try multiplying these matrices and vectors

```
A*z  
z*A ##what do you get? why"  
x*A ##what do you get?
```

This idea generalizes to multiplying two matrices together. For the multiplication

$C = AB$ , the matrix  $C$  is simply a collection of inner products:  $C_{ik} = \sum_i A_{ij}B_{jk}$ .

In this case,  $A$  must have the same number of columns as  $B$  has rows. Like ordinary multiplication, matrix multiplication is associative and distributive, but unlike ordinary multiplication, *it is not commutative*. In general,  $AB = BA$ . Try these matrix multiplications.

```
A*B  
B*A  
A*B==B*A ##what do you get?  
F*A  
A*F ##what do you get?
```

Now we are in a position to better understand the matrix transpose. If  $= A^t$ , then

$B_{ij} = A_{ji}$ . Think of this as flipping the matrix along the diagonal. This explains why the transpose operator changes a row vector into a column vector and vice versa. The following identity holds for the definitions of multiplication and transpose:  $(AB)^t = B^t A^t$ . This helps us to understand the difference between  $x^t A$  and  $Ax$ . Notice that for column vector  $x$ ,  $(Ax)^t = x^t A^t$ . Try these transpose matrices.

```
A'  
I'  
D'
```

There are a few more matrix terms we should know. A square matrix is an  $n \times n$  matrix (it has the same number of rows and columns). A diagonal matrix  $A$  has non-zero elements only along the diagonal ( $A_{ii}$ ), and zeros everywhere else. You can initialize a diagonal matrix in SCILAB by passing a vector to the *diag* command. The identity matrix is a special diagonal matrix with all diagonal elements set to 1. You can initialize an identity matrix using the *eye* command. Try the following SCILAB commands:

```
diag(1:10)
```

## 2.8.4. Solving Linear Equations

Let's take a step back for a moment, and try to solve the following set of linear equations:

$$x_1 + 3x_2 = 4$$

$$2x_1 + 2x_2 = 9$$

With a little manipulation, we find that  $x_1 = 4.75$  and  $x_2 = -0.25$ . We could solve this set of equations because we had 2 equations and 2 unknowns. How should we solve a set of equations with 50 equations and 50 unknowns?

Let's rewrite the previous expression in matrix form:

$$\begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 9 \end{bmatrix}$$

Notice that we could use the same form,  $A\mathbf{x} = \mathbf{b}$ , for our set of 50 equations with 50 unknowns. As expected, *SCILAB* provides all of the tools that we need to solve this matrix formula, and it uses the idea of a matrix inverse.

The inverse of a square matrix  $A$ , which is  $A^{-1}$  in the textbooks and  $\text{inv}(A)$  in

*SCILAB*, has the property that  $A^{-1}A = AA^{-1} = I$ . Using this, let's manipulate our previous equation:

$$A\mathbf{x} = \mathbf{b}$$

$$A^{-1}A\mathbf{x} = A^{-1}\mathbf{b}$$

$$\mathbf{x} = A^{-1}\mathbf{b}$$

Now solve the original equations in *SCILAB* using  $\text{inv}(A) * \mathbf{b}$ . You should get the vector containing 4.75 and -0.25. Try these.

```
inv(A)*z  
inv(B)*z  
inv(C)*z ##what do you get?
```

Recall that we were given a set of points  $(x_i, y_i)$ , and we were asked to find the coefficients  $a$  and  $b$  to fit the following linear model:

$$y_i \approx a + bx_i$$

You can think of each point as an equation, and write the entire data set in matrix form:

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

If you call the left matrix  $A$  and the right side  $\mathbf{b}$ , then calling  $A \setminus \mathbf{b}$  will ask *SCILAB* to solve for the values of  $a$  and  $b$  that minimize the least-squared error of the model.

```
[L U P] = lu(C)          ##this is called LUP
decomposition
singular matrices
d = P*z
y = L\z
x = U\y
##LUP decomposition work with
```

There are a few things to remember about matrix inverses. First of all, they are only defined for square matrices. It works with the transpose and multiplication operations with the following identities:  $(A^t)^{-1} = (A^{-1})^t$  and  $(AB)^{-1} = B^{-1}A^{-1}$ . You should be able to verify these properties on your own using the ideas we've developed. But the most important thing to know about matrix inverses is that they don't always exist, even for square matrices. Using *SCILAB*, try taking the inverse of the following matrix:

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

Now try inserting this  $A$  into the system of equations at the beginning of this section, and solving it using good old fashioned algebra. Why doesn't the inverse exist?

One more thing or maybe two. The determinant of a square matrix can tell you if it is invertible. A singular matrix (non-invertible) will have a determinant equal to zero. Try these.

```
det(A)  
det(C)
```

A singular matrix will have a null space that is not empty, while a matrix of full rank has an empty null space. Rank of a matrix will tell you how many independent row or columns are in a matrix. A  $5 \times 5$  matrix  $A$  is of full rank if  $\text{Rank}(A) = 5$ . Try these.

```
rank(A)  
kernel(A)  
rank(C)  
kernel(C)      ##how many column vectors are in null(C)  
               ##matrix dimension (N) minus Rank = size of  
the  
               ##null space  
               ##dimension of C minus rank of C = 5-3 = 2
```

### 2.8.5. Gauss-Jordan

Recall our matrix

```
C = [1 2 2 -1 4; 1 -1 0 1 2; 2 -2 0 2 4; 2 4 1 -1 3; -1 -2 -2 1 -4]
```

```
C=[1 2 2 -1 4; 1 -1 0 1 2; 2 -2 0 2 4; 2 4 1 -1 3; -1 -2 -2  
1 -4]
```

$$C = \begin{bmatrix} 1 & 2 & 2 & -1 & 4 \\ 1 & -1 & 0 & 1 & 2 \\ 2 & -2 & 0 & 2 & 4 \\ 2 & 4 & 1 & -1 & 3 \\ -1 & -2 & -2 & 1 & -4 \end{bmatrix}$$

We will now try to get the matrix  $C$  row reduced to Gauss-Jordon form using ERO.

```
C([2 5],:) = C([5 2],:)      ##swaps rows 2 and 5  
C(2,:) = C(1,:)+C(2,:)      ##adds row 1 and row 2 replacing  
row 2  
C(3,:) = -  
1/2*C(3,:);                 ##multiplies -1/2 times  
                                ## row 3 replacing row 3  
C(3,:) = C(1,:)-C(3,:);    ##subtracts row 3 from row 1
```

```
## 1 replacing row 3
```

If you have followed these steps, the matrix looks like:

$$C = \begin{bmatrix} 1 & 2 & 2 & -1 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 2 \\ 2 & 4 & 1 & -1 & 3 \end{bmatrix}$$

```
C(4,:) = C(1,:)-C(4,:); ##subtract row 4 from row 1
C(4,:)
```

If you have followed this step you get:

$$C = \begin{bmatrix} 1 & 2 & 2 & -1 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 1 & -1 & 3 \end{bmatrix}$$

```
C(5,:) = 2*C(1,:)-C(5,:); ##subtract row 5 from 2 times row 1
C(5,:) = 1/6*C(5,:); ##multiple row five by 1/6
C(5,:)
```

## 2.8.6. Quadratic Optimization

In quadratic optimization we would like to solve the equation  $Ax = b$  even if  $A$  is not square. Let's separate the problem into a few cases where the matrix  $A$  is a  $m \times n$  matrix:

If  $m < n$ , then we have more unknowns than equations. In general, this system will have infinitely many solutions.

If  $m > n$ , then we have more equations than unknowns. In general, this system doesn't have any solution. What if we don't want to have SCILAB always return "no solution", but we actually want the closest solution in the least squares sense? This is equivalent to minimizing the following quantity:

$$E = \frac{1}{2} \sum_i \left( \sum_j A_{ij}x_j - b_i \right)^2$$

*SCILAB* provides the backslash operator to accomplish the least squares fit for a matrix equation:  $x = A \backslash b$ . Type *help slash* to appreciate the power of this command. You'll see that we could have used this command to solve the square matrix equations, too. Try this.

```
w = [1 2 3 4]
rank(F)
[L U] = lu(F)           ##this is called LU
decomposition
matrices
y = L\w'
x = U\y
##LU decomposition work with non-square
```

### 2.8.7. Indexing

You can work with parts of matrices and vectors by indexing into them. You use a vector of integers to tell *SCILAB* which elements of a vector or matrix to use. For example, we create a vector

```
--> x = [1.2, 5, 7.6, 3, 8]
x =
  1.2000  5.0000  7.6000  3.0000  8.0000
```

Now, to see the second element of  $x$ , type

```
--> x(2)
ans = 5
```

You can also view a list of elements as follows.

```
--> x([1, 3, 4])
ans =
  1.2000  7.6000  3.0000
```

This last command displays the 1st, 3rd and 4th elements of the vector  $x$ .

To select rows and columns from a matrix, we use the same principle.  
Let's define a matrix

```
--> A = [1, 2, 3; 4, 5, 6; 7, 8, 9]
A =
 1  2  3
 4  5  6
 7  8  9
```

and select the 1st and 3rd rows and 2nd and 3rd columns:

```
--> A([1, 3], [2, 3])
ans =
 2  3
 8  9
```

The colon operator (:) can be used to select all rows or columns from a matrix. So, to select all the elements from the 2nd row, type

```
--> A(2, :)
ans =
 4  5  6
```

You can also use : like this to select all matrix elements:

```
--> A(:, :)
ans =
 1  2  3
 4  5  6
 7  8  9
```

## 2.8.8. Ranges

We can also select a range of rows or columns from a matrix. We specify a range with

```
start:step:stop
```

You can actually type ranges at the *SCILAB* prompt to see what the results are. For example,

```
--> 1:3:10
ans =
 1  4  7  10
```

The first number displayed was start, the second was start + step, the third, start + (2 \* step). And the last number was less than or equal to stop.

Often, you simply want the step size to be 1. In this case, you can leave out the step parameter and type

```
--> 1:10
ans =
 1  2  3  4  5  6  7  8  9  10
```

As you can see, the result of a range command is simply a vector of integers. We can now use this to index into a vector or matrix. To select the 2\times 2 submatrix at the top left of A, use

```
--> A(1:2, 1:2)
ans =
 1  2
 4  5
```

Finally, there is a keyword called end that can be used when indexing into a matrix or vector. It refers to the last element in the row or column. For example, to see the last column in a matrix, you can use

```
--> A(:,end)
ans =
 3
 6
 9
```

## 2.8.9. Where to find more information

A good book on this material is *Introduction to Linear Algebra* by Gilbert Strang. All of the material we've covered here can be found in the first two chapters of that book.

## 2.8.10. SCILAB Functions and Operators Functions

$A/B$  computes  $X$  such that  $XB = A$ . This is called right division and is done without forming the inverse of  $B$ .

$A \setminus B$  computes  $X$  such that  $AX = B$ . This is called left division and is done without forming the inverse of  $A$ .

`balance(V)`, eigenvalue balancing

`cond(A)`, condition number

`det(A)`, computes the determinant of the matrix  $A$ .

`dot(x, y)`, dot product

`lambda=spec(A)` returns the eigenvalues of  $A$  in the vector `lambda`, and

`[V, lambda]=spec(A)` also returns the eigenvectors in `V` but `lambda` is a now matrix whose diagonals contain the eigenvalues. This relationship holds true (within round off errors)  $A = V\lambda V^{-1}$ .

`eye(n, n)`, returns the  $n \times n$  identity matrix. You can also use `eye(m, n)` to return  $m \times n$  rectangular identity matrices.

`inv(A)`, computes the inverse of non-singular matrix  $A$ . Note that calculating the inverse is often ‘not’ necessary. See the next two operators as examples. Note that in theory  $AA^{-1}$  should return the identity matrix, but in practice, there may be some round off errors so the result may not be exact.

`norm(A, p)` or `norm(A)` computes the  $p$ -norm of the matrix (or vector)  $A$ . The second argument is optional with default value  $p = 2$ .

`kernel(A)`, orthonormal basis of the null space

`orth(A)`, orthonormal basis of the range space

`pinv(A)`, pseudoinverse

`rank(A)`, computes the (numerical) rank of a matrix.

`rref(A)`, returns the reduced row echelon form of  $A$ .

`trace(A)`, computes the trace (sum of the diagonal elements) of  $A$ .

`logm(A)`, computes the matrix logarithm of a square matrix.

`linspace(a, b, n)`, returns a vector with  $n$  values, such that the first element equals  $a$ , the last element equals  $b$  and the difference between consecutive elements is constant. The last argument,  $n$ , is optional with default value 100.

`logspace(a, b, n)`, returns a vector with  $n$  values, such that the first element equals  $10^a$ , the last element equals  $10^b$  and the ratio between consecutive elements is constant. The last argument,  $n$  is optional with default value 50.

`matrix_type(A)`, returns full, singular, diagonal or rectangular, and so on.

`ones(m, n)`, returns an  $m \times n$  matrix filled with 1s. Similarly, `ones(n)` returns  $n \times n$  square matrix.

`rand(m, n)`, returns an  $m \times n$  matrix filled with random elements drawn uniformly from [0, 1). Similarly, `rand(n)` returns  $n \times n$  square matrix.

`randperm(n)`, returns a row vector containing a random permutation of the numbers 1, 2, ...,  $n$

`sqrtm(A)`, computes the matrix square root of a square matrix.

`tril(A)`, returns the lower triangular part of  $A$ .

`triu(A)`, returns the upper triangular part of  $A$ .

`zeros(m, n)`, returns an  $m \times n$  matrix filled with 0s. Similarly, `zeros(n)` returns  $n \times n$  square matrix.

### 2.8.11. Factorizations

`R=chol(A)`, computes the Cholesky factorization of the symmetric positive definite matrix  $A$ , i.e. the upper triangular matrix  $R$  such that  $R^T R = A$ .

`[L, U]=lu(A)`, computes the  $LU$  decomposition of  $A$ , i.e.  $L$  is lower triangular,  $U$  upper triangular and  $A = LU$ .

`[Q, R]=qr(A)`, computes the  $QR$  decomposition of  $A$ , i.e.  $Q$  is orthogonal,  $R$  is upper triangular and  $A = QR$ .

### 2.8.12. Matrix manipulations

`matrix (A,m,n)` or `matrix(A,[m,n])`, transforms the  $A$  matrix (or vector) into an  $n \times m$  matrix by stacking columnwise the entries of  $A$ .

`flipdim(A,dim,[,sb])`, Given  $x$ , a scalar / vector / matrix / hypermatrix of any type and two positive integers  $dim$  and  $sb$ , this function flips the  $x$  components by blocks of size  $sb$  along the dimension number  $dim$  of  $x$  ( $x$  and  $y$  have the same size).

`permute(x,dims)`, permutes the dimensions of an array according to `dims`.

`resize_matrix(mat,nbRows,nbCols)`, Creates a matrix of sizes `[nbRows, nbCols]` or `newSizes`, or an hypermatrix of sizes `newSizes`.

`repmat(A,m,n,...)`, or similarly `repmat(A,[m,n])`, returns a large matrix consisting of an  $m \times n$  tiling of copies of  $A$ . The size of the result is `[size(A,1)*M, size(A,2)*N]`.

`pertrans(A)`, returns the simultaneous permutation and transposition of  $X$ , i.e. the symmetric of  $X$  with reference to the second diagonal (utility function).

## 2.9. Review Problems

Gauss and Gauss-Jordan Elimination Methods

For problems 1 & 2 **SOLVE by HAND**:

- Problem 1 - Use *Algebraic* notation with **Elimination** or **Substitution** methods.
- Problem 2 - Use *Matrix* notation and **Gauss Elimination** method with back substitution.

Give *algebraic solutions*  $x=?$ ,  $y=?$ ,... or say *No Solution*, or give *parametric equations* for infinite solutions when necessary.

$$\begin{aligned} 1. \quad 6x - 2y &= 3 \\ &x + 3y = 4 \end{aligned}$$

$$\begin{aligned} 2. \quad x_1 &\quad - 3x_3 = -2 \\ 3x_1 + x_2 - 2x_3 &= 5 \\ 2x_1 + 2x_2 + x_3 &= 4 \end{aligned}$$

For problems 3 & 4 **SOLVE with SCILAB:**

Problems 3 & 4 - Use *elementary row operations* to apply the **Gauss-Jordan** process to solve these systems of linear equations. See the Linear Algebra subsections Section 1 and Section 2 of the Linear Algebra Notebook for examples of doing the row commands with **SCILAB**.

Give *algebraic solutions*:  $x = ?$ ,  $y = ?$ , ... or say: *No Solution*, or give *parametric equations* for infinite solutions when necessary.

$$3. \quad \begin{aligned} 6x - 8y &= 5 \\ -9x + 12y &= 7 \end{aligned}$$

$$4. \quad \begin{aligned} 3x - z &= -2 \\ 2x + y - 3z &= 5 \\ x + 2y + 2z &= 4 \end{aligned}$$

Apply the **Gauss-Jordan** method using the **SCILAB** function **rref( )** to solve each system. This will transform the matrices into reduced row echelon form. Add text comments to report your final solution for each problem. Express the solution in algebraic notation (i.e.  $x = ?$ ) or parametric form when necessary.

$$5. \quad \begin{aligned} 12x - 15y &= 9 \\ -2x + 3y &= 4 \\ 8x - 10y &= 6 \end{aligned}$$

$$6. \quad \begin{aligned} 6w + 4x + 3y - 5z &= 4 \\ 4w + 2x + 4 - 3z &= 2 \\ 7w + 5x + 2y - 8z &= 5 \\ w - 4x - 3z &= 18 \end{aligned}$$

$$7. \quad \begin{aligned} 1/12x_1 + 5/12x_2 + x_3 &= 14 \\ 1/3x_2 + 2/3x_3 + 60x_4 &= 3 \\ x_3 + 76x_4 + 6x_5 &= 8 \end{aligned}$$

Note:  $x_1$  through  $x_5$  is **five** variables!

You may discuss **SCILAB** techniques and check answers with other students, but all work handed in must be entirely your own.

HAND-INS Should Include: Hand-written solutions on attached paper with clearly marked *problem numbers* and *answers*. For software

solutions produce a computer printout with *Text Comments* inserted at the beginning of each problem to label the *problem number*, and a *Text Solution* inserted at the end of each problem on the printout to describe the solution. Express the text solutions with algebraic notation i.e.  $x_1 = ?$ ,  $x_2 = ?$ , ... or  $n$ -tuple notation:  $(x_1, \dots)$

### 3. What is a Model?

Recently I saw a question on Answer.com that was not answered, so I took the opportunity to provide a response. Afterward, I thought about people I have worked with/for over the years, very brilliant people in their own field, who really did not know the answer to this question.

#### 3.1. Introduction

First, not all models are mathematical. Modeling in general is to pretend that one deals with a real thing while really working with an imitation. In operations research the imitation is a computer model of the simulated reality. A flight simulator on a PC is also a computer model of some aspects of the flight: it shows on the screen the controls and what the “pilot” (the youngster who operates it) is supposed to see from the “cockpit” (his armchair). A statistical model predicting who has the propensity to buy a certain product, or a machine learning algorithm with the same objective are also models.

In order to understand what it means to model a phenomena or process, we must first understand the term “model” and understand its limitation. A model is a physical, mathematical, or otherwise logical “representation” of a system, entity, phenomenon, or process. “Representation” or “imitation” are key in understanding models. A model can also be thought of as an abstraction of the real world, or an approximation of it. If you think about the problem of modeling a human being, or just the mind of a human, you can immediately see the limitations of modeling. We can use the term “system” to encompass systems, entities, phenomenon, or process. Since a modeling is a representation, abstraction, or approximation of the “system” being modeled, we must understand that it is not an “exact” representation, i.e., we can't model every aspect of the system. First, we don't know everything we need to know in order to model the system. We may not be able to define a process of the system with mathematical precision, or with heuristic algorithms, and many of the processes may not appear logical. The data that we use for a statistical model, for instance, may not be perfect (probably isn't), and the model is only as good as the data upon which it is built. Second, even if we were to know everything about

the system, we may not have enough computing power to model every process, at least for complex systems, e.g., a human being, the earth's ecosystem, etc.; we may not have enough money to spend on it; and we may not have enough time to accomplish it.

Yet with their limitations, models are a good way to gain understanding of how a system operates. George E.P. Box said, "All models are wrong; some models are useful." They are wrong in that there is not a one-to-one mapping from the real system to the model; they are useful in that we can use them to understand the system, or at least certain aspects of the system. Even though we often use models to predict behavior, this is a dangerous process, and we must do it with caution.

There are many types of models that one can use when trying to represent a system. These can be divided into several classes, and for simplicity we will consider two classes: physical models and symbolic models. Physical models include mock-up models (e.g., a vehicle mock-up), scale models (e.g., a 1:48 scale aircraft model), Iconic model, natural model, or a fashion model (which represent what we all want to look like).

Symbolic models include narrative models, graphical models, tabular models, software models, and mathematical models. These models are not necessarily mutually exclusive. For example a tabular model might contain data derived from a mathematical model, or a mathematical model may be embedded in software code.

Thus, a mathematical model is a symbolic model, or a symbolic representation or imitation of a real system or phenomenon. An example of a model is the equation for a line (a liner model)  $y = mx + b$ . A more complicate mathematical model is a model of the waiting time for the drive through window at McDonalds during lunchtime (I will not confuse you with the mathematical formula). We do have models that cover all times during the day (at the drive through), but the solutions to most of these models cannot be derived, "mathematically." We use numerical approximations, heuristic models (or algorithms) and simulations to solve these.

## 3.2. Model Nomenclature

### 3.2.1. Static and Dynamic Models

A **static model** is one in which the decision variables do not involve sequences of decisions over multiple periods. A **dynamic model** is a model in which the decision variables *do* involve sequences of decisions over multiple periods. Basically, in a static model we solve a “one-shot” problem whose solutions prescribe optimal values of decision variables at all points in time. Example 1 is an example of a static model; the optimal solution will tell Daisy how to maximize yield at all points in time.

#### EXAMPLE 1. Pharmco Drugs

Pharmco Drugs produces Magcalia in huge batches by heating a chemical mixture in a pressurized container. Each time a batch is processed, a different amount of Magcalia is produced. The amount produced is the *process yield* (measured in pounds). Pharmco is interested in understanding the factors that influence the yield of the Magcalia production process.

**Solution.** Pharmco is first interested in determining the factors that influence the yield process. This would be referred to as a *descriptive model*, because it describes the behavior of the actual yield as a function of various factors. Pharmco might determine (using regression methods) that the following factors influence yield.

- Container volume in liter (V)
- Container pressure in milliliters (P)
- Container temperature in degrees Celsius (T)
- Chemical composition of the processed mixture

If we let A, B, and C be the percentage of mixture made up of chemicals A, B, and C, then Pharmco might find, for example, that

$$\begin{aligned} \text{Yield} = & 300 + 0.8V + 0.01P + 0.06T + 0.001TP - 0.01T^2 - 0.01P^2 \\ & + 11.7A + 9.4B + 16.4C + 19AB + 11.4AC - 9.6BC \end{aligned}$$

To determine this process, the yield relationship would have to be measured for many different combinations of the factors. Knowledge of this equation would enable Pharmco to describe the yield of the

production process once volume, pressure, temperature, and chemical composition were known.

### EXAMPLE 2. Cruiseco Inventory

For an example of a dynamic model, consider a company (call it Cruiseco) that must determine how to minimize the cost of meeting (on time) the demand for sailboats during the next year. Clearly Cruiseco's must determine how many sailboats it will produce during each of the next four quarters. Cruiseco's decisions involve decisions made over multiple periods, hence a model of Cruiseco's problem would be a dynamic model.

### 3.2.2. Linear and Nonlinear Models

Suppose that whenever decision variables appear in the objective function and in the constraints of an optimization model, the decision variables are always multiplied by constants and added together. Such a model is a **linear model**. If an optimization model is not linear, then it is a **nonlinear model**. In the constraints of Example 1, the decision variables are always multiplied by constants and added together. Thus, Example 1's constraints pass the test for a linear model. However, in the objective function for Example 1, the terms  $0.01T \cdot P$ ,  $-0.01T^2$ ,  $19A \cdot B$ ,  $11.4A \cdot C$ , and  $-9.6B \cdot C$  make the model nonlinear. In general, nonlinear models are much harder to solve than linear models. We will discuss linear models in Chapters 2 through 10. Nonlinear models will be discussed in Chapter 11.

### 3.2.3. Integer and Noninteger Models

If one or more decision variables must be integer, then we say that an optimization model is an **integer model**. If all the decision variables are free to assume fractional values, then the optimization model is a **noninteger model**. Clearly, volume, temperature, pressure, and percentage composition of our inputs may all assume fractional values. Thus, Example 1 is a noninteger model. If the decision variables in a model represent the number of workers starting work during each shift at a fast-food restaurant, then clearly we have an integer model. Integer models are much harder to solve than nonlinear models. They will be discussed in detail in Chapter 9.

### 3.2.4. Deterministic and Stochastic Models

Suppose that for any value of the decision variables, the value of the objective function and whether or not the constraints are satisfied is known with certainty. We then have a **deterministic model**. If this is not the case, then we have a **stochastic model**. All models in the first 12 chapters will be deterministic models. Stochastic models are covered in Chapter 15.

## 3.3. Model Functional Forms

We can construct different kinds of models with different functional forms. Some of these can be used for the same purpose (e.g. regression and neural networks) and others for a specific purpose, e.g. linear program. Some model types we will discuss include:

- Regression models
- Optimization models
- Network models
- Queuing models
- Simulation models

Some specific functional forms of model that we will discuss include

- Linear regression models
- Generalized linear models
- Linear programming models
- Integer programming models
- Dynamic programming models
- Discrete event simulation models

We begin with Linear Programming models. These models are prescriptive or optimization models. A prescriptive model “prescribes” behavior for an organization that will enable it to best meet its goal(s). The components of a prescriptive model include

- objective function(s)
- decision variables
- constraints

In short, an optimization model seeks to find values of the decision variables that optimize (maximize or minimize) an objective function among the set of all values for the decision variables that satisfy the given constraints.

### **EXAMPLE 1.** (continued) Maximizing Yield

Pharmco produces Magcalia in huge batches by heating a chemical mixture in a pressurized container. Each time a batch is processed, a different amount of Magcalia is produced. The amount produced is the *process yield* (measured in pounds). Daisy is interested in understanding the factors that influence the yield of the Magcalia production process. Describe a model-building process for this situation.

**Solution.** Art is first interested in determining the factors that influence the yield of the process.

#### The Objective Function

Naturally, Pharmco would like to maximize the yield of the process. In most models, there will be a function we wish to maximize or minimize. This function is called the model's *objective function*. Of course, to maximize the process yield we need to find the values of  $V$ ,  $P$ ,  $T$ ,  $A$ ,  $B$ , and  $C$  that make (1) as large as possible.

$$(1) \quad \text{yield} = 285 + .7V + .015P + .05T + .002T * P - .03T^2 - .001P^2 + 12.3A + 8.9B + 15.8C + 18A * B + 104A * C - 9.5B * C$$

In many situations, an organization may have more than one objective. For example, in assigning students to the two high schools in Bloomington, Indiana, the Monroe County School Board stated that the assignment of students involved the following objectives:

- Equalize the number of students at the two high schools.
- Minimize the average distance students travel to school.
- Have a diverse student body at both high schools.

#### The Decision Variables

The variables whose values are under our control and influence the performance of the system are called *decision variables*. In our example,

$V$ ,  $P$ ,  $T$ ,  $A$ ,  $B$ , and  $C$  are decision variables. Most of this book will be devoted to a discussion of how to determine the value of decision variables that maximize (sometimes minimize) an objective function.

## Constraints

In most situations, only certain values of decision variables are possible. For example, certain volume, pressure, and temperature combinations might be unsafe. Also,  $A$ ,  $B$ , and  $C$  must be nonnegative numbers that add to 1. Restrictions on the values of decision variables are called *constraints*. Suppose the following:

Volume must be between 1 and 5 liters.

- Pressure must be between 200 and 400 milliliters.
- Temperature must be between 100 and 200 degrees Celsius.
- Mixture must be made up entirely of  $A$ ,  $B$ , and  $C$ .

For the drug to properly perform, only half the mixture at most can be product  $A$ . These constraints can be expressed mathematically by the following constraints:

$$V \leq 5; V \geq 1; P \leq 400; P \geq 200; T \leq 200; T \geq 100; A \geq 0; B \geq 0; A + B + C = 1; A \leq 5$$

## The Complete Optimization Model

After letting  $z$  represent the value of the objective function, our entire optimization model may be written as follows:

$$\begin{aligned} \max z = & 285 + .7V + .015P + .05T + .002T^2 - .03T^2 - .001P^2 \\ & + 12.3A + 8.9B + 15.8C + 18A * B + 104A * C - 9.5B * C \end{aligned}$$

Subject to (s.t.)

$$\begin{aligned} V &\leq 5 \\ V &\geq 1 \\ P &\leq 400 \\ P &\geq 200 \\ T &\leq 200 \\ T &\geq 100 \\ A &\geq 0 \\ B &\geq 0 \end{aligned}$$

$$\begin{aligned}A + B + C &= 1 \\A &\leq 5\end{aligned}$$

Any specification of the decision variables that satisfies all of the model's constraints is said to be in the **feasible region**. For example,  $V = 2$ ,  $P = 300$ ,  $T = 50$ ,  $A = 0.4$ ,  $B = 0.3$ , and  $C = 0.1$  is in the feasible region. An **optimal solution** to an optimization model is any point in the feasible region that optimizes (in this case, *maximizes*) the objective function. Using the LPSolve we can determine that the optimal solution to this model is  $V = 5$ ,  $P = 200$ ,  $T = 100$ ,  $A = 0.294$ ,  $B = 0$ ,  $C = 0.706$ , and  $z = 183.38$ . Thus, a maximum yield of 183.38 pounds can be obtained with a 5-liter container, pressure of 200 milliliters, temperature of 100 degrees Celsius, and 29%  $A$  and 71%  $C$ . This means no other feasible combination of decision variables can obtain a yield exceeding 183.38 pounds.

### 3.4. Modeling in Operations Research

Years ago I learned the seven steps to model building as a budding Operations Research Analyst. As I grew from sapling to tree, I realized that something was missing, but I did not really know what link was absent until I plowed into analytics. Now I am prepared to tell you, from an Operations Research (OR) perspective what I believe is a more complete set of steps.

**Define the problem.** Defining the problem is necessary. Defining the "right" problem is absolutely critical, otherwise you're just wasting everyone's time. However, the customers you are working for may not know how to express the real problem or may not know what the problem really is. The Operations Research Analyst must ask the right questions and draw the right problem out form where it may be hiding.

**Define the Business case.** Once you identify the real problem, you have to help answer the question, "Is the problem worth solving (or at least worth the OR analyst's involvement)?". That may sound odd, but if the problem statement is not translated in to a business case, then the business does not need the problem solved badly enough to warrant the time and effort to solve it. We have to ask, "What is it worth to the

business?" Increased profit? Improved production? Savings of marketing dollars?...

**Define the Model Objective.** It takes a well-defined problem and solid business case to ensure that we build the right model or provide the right solution. In my earlier years, I often found myself halfway through a project before realizing I was not building the right model. A model objective that answers the call of the business case is essential. It will keep you on course.

**Determine the requirements.** We have a well-defined problem, a business case with a supporting model objective, so now let's go and build the model. Slow down! What are the requirements? Do you have the tools you need, the team you need, and the budget to support it? If data is to be used, do you have access to it, has it been processed, and is it in the right format? We have to determine the requirements or we may fail due to some oversight.

**Gather the Data.** If data is necessary, it will probably not be handed to you on a silver platter ready for consumption. It may be "dirty", in the wrong format, incomplete, or just not right. If you have fully identified the requirements, you will already know what has to be accomplished at this stage.

**Process the Data.** This is where you may spend an abundant amount of time. If you do not have clean, complete, and reliable data, you are doomed. You may have to remove inconsistencies, impute missing values, and so on. Then you have to analyze the data, perform data reduction, and integrate the data so that it is ready for use. Modeling with "bad" data results in a "bad" model!

**Build the Model.** So, you have done the hard part and it's time to have fun. Building the model is an iterative process. I doubt that anyone ever uses the first model they build. If they do, they probably should not. There is as much art as there is science in model development. Modeler judgment is as important as p-values, and Chi-square tests. If the model does not make sense to you—is not intuitive—then it is probably not the final model.

**Interpret the Model Results.** Here is where we often have the most difficulty as analysts. We understand the mathematical and statistical interpretation of results, but we have to go beyond that and translate it to the customer's language. They are not going to get the significance of the odds ratio of variable X. If you start thinking about that here, it will make life less painful later on.

**Validate the Model for Production.** This is purely a scientific task. The model must be documented every step of the way and then stand up to the scrutiny of stress testing, peer review, and so on. This is where you find out if you really did build the right model, and you built it right.

**Perform an Economic Analysis.** Here is where you answer the call of the business case. You must translate the results to dollars, the economic benefit of the solution, or another metric if savings or profit are not objectives. For instance, the military OR may be saving lives instead of dollars.

**Present the Results.** If you bothered to interpret the results and translate them to "customer speak", this part is easy. But, you are not just presenting results, you are selling the solution. If your work and your recommendations are not implemented, you just wasted a lot of time, effort and money.

**Follow-up.** Too often we fix the problem and go back to our lives or onto another problem, but following up with the customer to evaluate the effectiveness of the solution you provided is just as important as the solution that you provided. It is especially important if you want to ever see that customer again.

# 4. What is Linear Programming?

**Linear programming (LP)** is a tool for solving optimization problems. In 1947, George Dantzig developed an efficient method, the **simplex algorithm**, for solving linear programming problems (also called LP). Since the development of the simplex algorithm, LP has been used to solve optimization problems in industries as diverse as banking, education, forestry, petroleum, and trucking. In a survey of Fortune 500 firms, 85% of the respondents said they had used linear programming. As a measure of the importance of linear programming in operations research, approximately 70% of this book will be devoted to linear programming and related optimization techniques.

In Section 3.1, we begin our study of linear programming by describing the general characteristics shared by all linear programming problems. In Sections 3.2 and 3.3, we learn how to solve graphically those linear programming problems that involve only two variables. Solving these simple LPs will give us useful insights for solving more complex LPs. The remainder of the chapter explains how to formulate linear programming models of real-life situations.

## 4.1. What Is a Linear Programming Problem?

In this section, we introduce linear programming and define important terms that are used to describe linear programming (optimization) problems.

**Algorithm 3.1.** Optimization modeling.

**Step 1.** Find the **decision variables**, i.e. find out what are the variables whose values you can choose.

**Step 2.** Find the **objective function**, i.e. find out how your objective to be minimized or maximized depends on the decision variables.

**Step 3.** Find the **constraints**, i.e. find out the (in) equalities that the decision variables must satisfy. (Don't forget the possible sign constraints!)

**EXAMPLE 1.** Gepetto's Woodcarving, Inc.

Gepetto's manufactures two types of wooden toys: soldiers and trains. A soldier sells for \$27 and uses \$10 worth of raw materials. Each soldier that is manufactured increases Gepetto's variable labor and overhead costs by \$14. A train sells for \$21 and uses \$9 worth of raw materials. Each train built increases Gepetto's variable labor and overhead costs by \$10. The manufacture of wooden soldiers and trains requires two types of skilled labor: carpentry and finishing. A soldier requires 2 hours of finishing labor and 1 hour of carpentry labor. A train requires 1 hour of finishing and 1 hour of carpentry labor. Each week, Gepetto can obtain all the needed raw material but only 100 finishing hours and 80 carpentry hours. Demand for trains is unlimited, but at most 40 soldiers are bought each week. Gepetto wants to maximize weekly profit (revenues – costs). Formulate a mathematical model of Gepetto's situation that can be used to maximize Gepetto's weekly profit.

**Solution.** In developing the Gepetto model, we explore characteristics shared by all linear programming problems.

**Decision Variables.** We begin by defining the relevant decision variables. In any linear programming model, the decision variables should completely describe the decisions to be made (in this case, by Gepetto). Clearly, Gepetto must decide how many soldiers and trains should be manufactured each week. With this in mind, we define

$$\begin{aligned}x_1 &= \text{number of soldiers produced each week} \\x_2 &= \text{number of trains produced each week}\end{aligned}$$

**Objective Function.** In any linear programming problem, the decision maker wants to maximize (usually revenue or profit) or minimize (usually costs) some function of the decision variables. The function to be maximized or minimized is called the objective function. For the Gepetto problem, we note that fixed costs (such as rent and insurance) do not depend on the values of  $x_1$  and  $x_2$ . Thus, Gepetto can concentrate on maximizing (weekly revenues) – (raw material purchase costs) – (other variable costs).

Gepetto's weekly revenues and costs can be expressed in terms of the decision variables  $x_1$  and  $x_2$ . It would be foolish for Gepetto to

manufacture more soldiers than can be sold, so we assume that all toys produced will be sold. Then

$$\begin{aligned}\text{Weekly revenues} &= \text{weekly revenues from soldiers} \\ + \text{weekly revenues from trains} &= \left(\frac{\text{dollars}}{\text{soldiers}}\right) \left(\frac{\text{soldiers}}{\text{week}}\right) \\ &\quad + \left(\frac{\text{dollars}}{\text{train}}\right) \left(\frac{\text{trains}}{\text{week}}\right) = 27x_1 + 21x_2\end{aligned}$$

Also,

$$\text{Weekly raw material costs} = 10x_1 + 9x_2$$

$$\text{Other weekly variable costs} = 14x_1 + 10x_2$$

Then Gepetto wants to maximize

$$(27x_1 + 21x_2) - (10x_1 + 9x_2) - (14x_1 + 10x_2) = 3x_1 + 2x_2$$

Another way to see that Gepetto wants to maximize  $3x_1 + 2x_2$  is to note that

$$\begin{aligned}\text{Weekly revenues} &= \text{weekly contribution to profit from soldiers} \\ &\quad - \text{weekly nonfixed costs} \\ &\quad + \text{weekly contribution to profit from trains} \\ &= \left(\frac{\text{contribution to profit}}{\text{soldier}}\right) \left(\frac{\text{soldiers}}{\text{week}}\right) \\ &\quad + \left(\frac{\text{contribution to profit}}{\text{train}}\right) \left(\frac{\text{trains}}{\text{week}}\right)\end{aligned}$$

Also.

$$\frac{\text{contribution to profit}}{\text{soldier}} = 27 - 10 - 14 = 3$$

$$\frac{\text{contribution to profit}}{\text{train}} = 21 - 9 - 10 = 2$$

Then, as before, we obtain

$$\text{Weekly revenues} - \text{weekly nonfixed costs} = 3x_1 + 2x_2$$

Thus, Gepetto's objective is to choose  $x_1$  and  $x_2$  to maximize  $3x_1 + 2x_2$ . We use the variable  $z$  to denote the objective function value of any LP. Gepetto's objective function is

$$\text{Maximize } z = 3x_1 + 2x_2 \quad (4.1)$$

(In the future, we will abbreviate “maximize” by max and “minimize” by min.) The coefficient of a variable in the objective function is called the objective function coefficient of the variable. For example, the objective function coefficient for  $x_1$  is 3, and the objective function coefficient for  $x_2$  is 2. In this example (and in many other problems), the objective function coefficient for each variable is simply the contribution of the variable to the company’s profit.

**Constraints.** As  $x_1$  and  $x_2$  increase, Gepetto’s objective function grows larger. This means that if Gepetto were free to choose any values for  $x_1$  and  $x_2$ , the company could make an arbitrarily large profit by choosing  $x_1$  and  $x_2$  to be very large. Unfortunately, the values of  $x_1$  and  $x_2$  are limited by the following three restrictions (often called constraints):

**Constraint 1.** Each week, no more than 100 hours of finishing time may be used.

**Constraint 2.** Each week, no more than 80 hours of carpentry time may be used.

**Constraint 3.** Because of limited demand, at most 40 soldiers should be produced each week.

The amount of raw material available is assumed to be unlimited, so no restrictions have been placed on this.

The next step in formulating a mathematical model of the Gepetto problem is to express Constraints 1–3 in terms of the decision variables  $x_1$  and  $x_2$ . To express Constraint 1 in terms of  $x_1$  and  $x_2$ , note that

$$\begin{aligned} \frac{\text{Total finishing hrs.}}{\text{Week}} &= \left( \frac{\text{finishing hrs.}}{\text{soldiers}} \right) \left( \frac{\text{soldiers made}}{\text{week}} \right) \\ &\quad + \left( \frac{\text{finishing hrs.}}{\text{train}} \right) \left( \frac{\text{trains made}}{\text{week}} \right) = (2)x_1 + (1)x_2 \\ &= 2x_1 + x_2 \end{aligned}$$

Now constraint 1 may be expressed by

$$2x_1 + x_2 \leq 100 \quad (4.2)$$

Note that the units of each term in (2) are finishing hours per week. For a constraint to be reasonable, all terms in the constraint must have the same units. Otherwise one is adding apples and oranges, and the constraint won't have any meaning.

To express Constraint 2 in terms of  $x_1$  and  $x_2$ , note that

$$\begin{aligned}\frac{\text{Total carpentry hrs.}}{\text{Week}} &= \left(\frac{\text{carpentry hrs.}}{\text{soldiers}}\right)\left(\frac{\text{soldiers made}}{\text{week}}\right) \\ &\quad + \left(\frac{\text{carpentry hrs.}}{\text{train}}\right)\left(\frac{\text{trains made}}{\text{week}}\right) = (1)x_1 + (1)x_2 \\ &= x_1 + x_2\end{aligned}$$

Now constraint 2 may be expressed by

$$x_1 + x_2 \leq 80 \tag{4.3}$$

Again, note that the units of each term in (3) are the same (in this case, carpentry hours per week).

Finally, we express the fact that at most 40 soldiers per week can be sold by limiting the weekly production of soldiers to at most 40 soldiers. This yields the following constraint:

$$x_1 \leq 40$$

Thus (2)–(4) express Constraints 1–3 in terms of the decision variables; they are called the constraints for the Gepetto linear programming problem. The coefficients of the decision variables in the constraints are called **technological coefficients**. This is because the technological coefficients often reflect the technology used to produce different products. For example, the technological coefficient of  $x_2$  in (3) is 1, indicating that a soldier requires 1 carpentry hour. The number on the right-hand side of each constraint is called the constraint's **right-hand side** (or **rhs**). Often the rhs of a constraint represents the quantity of a resource that is available.

**Sign Restrictions.** To complete the formulation of a linear programming problem, the following question must be answered for each decision variable: Can the decision variable only assume nonnegative values, or

is the decision variable allowed to assume both positive and negative values?

If a decision variable  $x_i$  can only assume nonnegative values, then we add the **sign restriction**  $x_i \geq 0$ . If a variable  $x_i$  can assume both positive and negative (or zero) values, then we say that  $x_i$  is **unrestricted in sign** (often abbreviated **urs**). For the Gepetto problem, it is clear that  $x_1 > 0$  and  $x_2 \geq 0$ . In other problems, however, some variables may be urs. For example, if  $x_i$  represented a firm's cash balance, then  $x_i$  could be considered negative if the firm owed more money than it had on hand. In this case, it would be appropriate to classify  $x_i$  as urs.

Combining the sign restrictions  $x_1 \geq 0$  and  $x_2 \geq 0$  with the objective function (4.4) and Constraints (4.5)–(4.7) yields the following optimization model:

$$\max z = 3x_1 + 2x_2 \quad (\text{Objective function}) \quad (4.4)$$

subject to (s.t.)

$$2x_1 + x_2 \leq 100 \quad (\text{Finishing constraint}) \quad (4.5)$$

$$x_1 + x_2 \leq 80 \quad (\text{Carpentry constraint}) \quad (4.6)$$

$$x_1 \leq 40 \quad (\text{Constraint on demand for soldiers}) \quad (4.7)$$

$$x_1 \geq 0 \quad (\text{Sign restriction})^+ \quad (4.8)$$

$$x_2 \geq 0 \quad (\text{Sign restriction}) \quad (4.9)$$

"Subject to" (s.t.) means that the values of the decision variables  $x_1$  and  $x_2$  must satisfy all constraints and all sign restrictions.

Before formally defining a linear programming problem, we define the concepts of linear function and linear inequality.

**DEFINITION ■** A function  $f(x_1, x_2, \dots, x_n)$  of  $x_1, x_2, \dots, x_n$  is a **linear function** if and only if for some set of constants  $c_1, c_2, \dots, c_n$ ,  $f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$  ■

For example,  $f(x_1, x_2) = 2x_1 + x_2$  is a linear function of  $x_1$  and  $x_2$ , but  $f(x_1, x_2) = x_1^2x_2$  is not a linear function of  $x_1$  and  $x_2$ .

**DEFINITION ■** For any linear function  $f(x_1, x_2, \dots, x_n)$  and any number  $b$ , the inequalities  $f(x_1, x_2, \dots, x_n) \leq b$  and  $f(x_1, x_2, \dots, x_n) \geq b$  are **linear inequalities**. ■

Thus,  $2x_1 + 3x_2 \leq 3$  and  $2x_1 + x_2 \geq 3$  are linear inequalities, but  $x_1^2 x_2 \geq 3$  is not a linear inequality.

**DEFINITION ■** A **linear programming problem (LP)** is an optimization problem for which we do the following:

1. We attempt to maximize (or minimize) a linear function of the decision variables. The function that is to be maximized or minimized is called the **objective function**.
2. The values of the decision variables must satisfy a set of constraints. Each constraint must be a linear equation or linear inequality.
3. A sign restriction is associated with each variable. For any variable  $x_i$ , the sign restriction specifies that  $x_i$  must be either nonnegative ( $x_i \geq 0$ ) or unrestricted in sign (urs). ■

Because Gepetto's objective function is a linear function of  $x_1$  and  $x_2$ , and all of Gepetto's constraints are linear inequalities, the Gepetto problem is a linear programming problem. Note that the Gepetto problem is typical of a wide class of linear programming problems in which a decision maker's goal is to maximize profit subject to limited resources.

#### 4.1.1. The Proportionality and Additivity Assumptions

The fact that the objective function for an LP must be a linear function of the decision variables has two implications.

1. The contribution of the objective function from each decision variable is proportional to the value of the decision variable. For example, the contribution to the objective function from making four soldiers ( $4 \times 3 = \$12$ ) is exactly four times the contribution to the objective function from making one soldier ( $\$3$ ).
2. The contribution to the objective function for any variable is independent of the values of the other decision variables. For

example, no matter what the value of  $x_2$ , the manufacture of  $x_1$  soldiers will always contribute  $3x_1$  dollars to the objective function.

Analogously, the fact that each LP constraint must be a linear inequality or linear equation has two implications.

1. The contribution of each variable to the left-hand side of each constraint is proportional to the value of the variable. For example, it takes exactly three times as many finishing hours ( $2 \times 3 = 6$  finishing hours) to manufacture three soldiers as it takes to manufacture one soldier (2 finishing hours).
2. The contribution of a variable to the left-hand side of each constraint is independent of the values of the other variables. For example, no matter what the value of  $x_1$ , the manufacture of  $x_2$  trains uses  $x_2$  finishing hours and  $x_2$  carpentry hours.

The first implication given in each list is called the **Proportionality Assumption of Linear Programming**. Implication 2 of the first list implies that the value of the objective function is the sum of the contributions from individual variables, and implication 2 of the second list implies that the left-hand side of each constraint is the sum of the contributions from each variable. For this reason, the second implication in each list is called the Additivity Assumption of Linear Programming.

For an LP to be an appropriate representation of a real-life situation, the decision variables must satisfy both the Proportionality and Additivity Assumptions. Two other assumptions must also be satisfied before an LP can appropriately represent a real situation: the Divisibility and Certainty Assumptions.

#### 4.1.2. The Divisibility Assumption

The **Divisibility Assumption** requires that each decision variable be allowed to assume fractional values. For example, in the Gepetto problem, the Divisibility Assumption implies that it is acceptable to produce 1.5 soldiers or 1.63 trains. Because Gepetto can-not actually produce a fractional number of trains or soldiers, the Divisibility Assumption is not satisfied in the Gepetto problem. A linear programming problem in which some or all of the variables must be

nonnegative integers is called an **integer programming problem**. The solution of integer programming problems is discussed in Chapter 9.

In many situations where divisibility is not present, rounding off each variable in the optimal LP solution to an integer may yield a reasonable solution. Suppose the optimal solution to an LP stated that an auto company should manufacture 150,000.4 compact cars during the current year. In this case, you could tell the auto company to manufacture

150,000 or 150,001 compact cars and be fairly confident that this would reasonably approximate an optimal production plan. On the other hand, if the number of missile sites that the United States should use were a variable in an LP and the optimal LP solution said that 0.4 missile sites should be built, it would make a big difference whether we rounded the number of missile sites down to 0 or up to 1. In this situation, the integer programming methods of Chapter 9 would have to be used, because the number of missile sites is definitely not divisible.

#### 4.1.3. The Certainty Assumption

The **Certainty Assumption** is that each parameter (objective function coefficient, right-hand side, and technological coefficient) is known with certainty. If we were unsure of the exact amount of carpentry and finishing hours required to build a train, the Certainty Assumption would be violated.

#### 4.1.4. Feasible Region and Optimal Solution

Two of the most basic concepts associated with a linear programming problem are feasible region and optimal solution. For defining these concepts, we use the term point to mean a specification of the value for each decision variable.

**DEFINITION ■** The **feasible region** for an LP is the set of all points that satisfies all the LP's constraints and sign restrictions. ■

For example, in the Gepetto problem, the point ( $x_1 = 40, x_2 = 20$ ) is in the feasible region. Note that  $x_1 = 40$  and  $x_2 = 20$  satisfy the constraints (2)–(4) and the sign restrictions (5)–(6):

**Constraint (2)**,  $2x_1 + x_2 \leq 100$ , is satisfied, because  $2(40) + 20 \leq 100$ .

**Constraint (3)**,  $x_1 + x_2 \leq 80$ , is satisfied, because  $40 + 20 \leq 80$ .

**Constraint (4)**,  $x_1 \leq 40$ , is satisfied, because  $40 \leq 40$ .

**Restriction (5)**,  $x_1 \geq 0$ , is satisfied, because  $40 \geq 0$ .

**Restriction (6)**,  $x_2 \geq 0$ , is satisfied, because  $20 \geq 0$ .

On the other hand, the point  $(x_1 = 15, x_2 = 70)$  is not in the feasible region, because even though  $x_1 = 15$  and  $x_2 = 70$  satisfy (2), (4), (5), and (6), they fail to satisfy (3):  $15 + 70$  is not less than or equal to 80. Any point that is not in an LP's feasible region is said to be an infeasible point. As another example of an infeasible point, consider  $(x_1 = 40, x_2 = -20)$ . Although this point satisfies all the constraints and the sign restriction (5), it is infeasible because it fails to satisfy the sign restriction (6),  $x_2 \geq 0$ . The feasible region for the Gepetto problem is the set of possible production plans that Gepetto must consider in searching for the optimal production plan.

**DEFINITION ■** For a maximization problem, an optimal solution to an LP is a point in the feasible region with the largest objective function value. Similarly, for a minimization problem, an optimal solution is a point in the feasible region with the smallest objective function value. ■

Most LPs have only one optimal solution. However, some LPs have no optimal solution, and some LPs have an infinite number of solutions (these situations are discussed in Section 4.3). In Section 4.2, we show that the unique optimal solution to the Gepetto problem is  $(x_1 = 20, x_2 = 60)$ . This solution yields an objective function value of

$$z = 3x_1 + 2x_2 = 3(20) + 2(60) = \$180$$

When we say that  $(x_1 = 20, x_2 = 60)$  is the optimal solution to the Gepetto problem, we are saying that no point in the feasible region has an objective function value that exceeds 180. Gepetto can maximize profit by building 20 soldiers and 60 trains each week. If Gepetto were to produce 20 soldiers and 60 trains each week, the weekly profit would be \$180 less weekly fixed costs. For example, if Gepetto's only fixed cost

were rent of \$100 per week, then weekly profit would be  $180 - 100 = \$80$  per week.

#### 4.1.5. Problems

1. Deereco manufactures two types of trucks: 1 and 2. Each truck must go through the painting shop and assembly shop. If the painting shop were completely devoted to painting Type 1 trucks, then 800 per day could be painted; if the painting shop were completely devoted to painting Type 2 trucks, then 700 per day could be painted. If the assembly shop were completely devoted to assembling truck 1 engines, then 1,500 per day could be assembled; if the assembly shop were completely devoted to assembling truck 2 engines, then 1,200 per day could be assembled. Each Type 1 truck contributes \$300 to profit; each Type 2 truck contributes \$500. Formulate an LP that will maximize Deereco's profit.
2. Farmer McDonald must determine how many acres of corn and wheat to plant this year. An acre of wheat yields 25 bushels of wheat and requires 10 hours of labor per week. An acre of corn yields 10 bushels of corn and requires 4 hours of labor per week. All wheat can be sold at \$4 a bushel, and all corn can be sold at \$3 a bushel. Seven acres of land and 40 hours per week of labor are available. Government regulations require that at least 30 bushels of corn be produced during the current year. Let  $x_1$  = number of acres of corn planted, and  $x_2$  = number of acres of wheat planted. Using these decision variables, formulate an LP whose solution will tell Farmer McDonald how to maximize the total revenue from wheat and corn.

#### 4.2. The Graphical Solution of Two-Variable Linear Programming Problems

Any LP with only two variables can be solved graphically. We always label the variables  $x_1$  and  $x_2$  and the coordinate axes the  $x_1$  and  $x_2$  axes. Suppose we want to graph the set of points that satisfies

$$2x_1 + 3x_2 \leq 6 \quad (4.10)$$

The same set of points  $(x_1, x_2)$  satisfies

$$3x_2 \leq 6 - 2x_1$$

This last inequality may be rewritten as

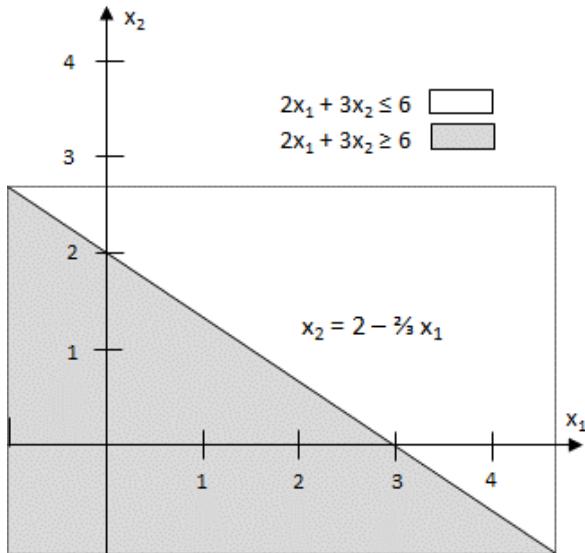
$$x_2 \leq \frac{1}{3}(6 - 2x_1) = 2 - \frac{2}{3}x_1 \quad (4.11)$$

Because moving downward on the graph decreases  $x_2$  (see Figure 1), the set of points that satisfies (4.10) and (4.11) lies on or below the line  $x_2 = 2 - \frac{2}{3}x_1$ . This set of points is indicated by darker shading in Figure 1.

Note, however, that  $x_2 = 2 - \frac{2}{3}x_1$ ,  $3x_2 = 6 - 2x_1$ , and  $2x_1 + 3x_2 = 6$  are all the same line. This means that the set of points satisfying (4.10) lies on or below the line  $2x_1 + 3x_2 = 6$ . Similarly, the set of points satisfying  $2x_1 + 3x_2 \geq 6$  lies on or above the line  $2x_1 + 3x_2 = 6$ . (These points are shown by lighter shading in Figure 1.)

Consider a linear inequality constraint of the form  $f(x_1, x_2) \geq b$  or  $f(x_1, x_2) \leq b$ . In general, it can be shown that in two dimensions, the set of points that satisfies a linear inequality includes the points on the line  $f(x_1, x_2) = b$ , defining the inequality plus all points on one side of the line.

There is an easy way to determine the side of the line for which an inequality such as  $f(x_1, x_2) \leq b$  or  $f(x_1, x_2) \geq b$  is satisfied. Just choose any point  $P$  that does not satisfy the line  $f(x_1, x_2) = b$ . Determine whether  $P$  satisfies the inequality. If it does, then all points on the same side as  $P$  of  $f(x_1, x_2) = b$  will satisfy the inequality. If  $P$  does not satisfy the inequality, then all points on the other side of  $f(x_1, x_2) = b$ , which does not contain  $P$ , will satisfy the inequality. For example, to determine whether  $2x_1 + 3x_2 \geq 6$  is satisfied by points above or below the line  $2x_1 + 3x_2 = 6$ , we note that  $(0, 0)$  does not satisfy  $2x_1 + 3x_2 \geq 6$ . Because  $(0, 0)$  is below the line  $2x_1 + 3x_2 = 6$ , the set of points satisfying  $2x_1 + 3x_2 \geq 6$  includes the line  $2x_1 + 3x_2 = 6$  and the points above the line  $2x_1 + 3x_2 = 6$ . This agrees with Figure 1.



**FIGURE 1.** Graphing a Linear Inequality (made with SCILAB)

#### 4.2.1. Finding the Feasible Solution

We illustrate how to solve two-variable LPs graphically by solving the Gepetto problem. To begin, we graphically determine the feasible region for Gepetto's problem. The feasible region for the Gepetto problem is the set of all points  $(x_1, x_2)$  satisfying

$$2x_1 + x_2 \leq 100 \quad (\text{Finishing constraint}) \quad (4.12)$$

$$x_1 + x_2 \leq 80 \quad (\text{Carpentry constraint}) \quad (4.13)$$

$$x_1 \leq 40 \quad (\text{Constraint on demand for soldiers}) \quad (4.44)$$

$$x_1 \geq 0 \quad (\text{Sign restriction}) \dagger \quad (4.15)$$

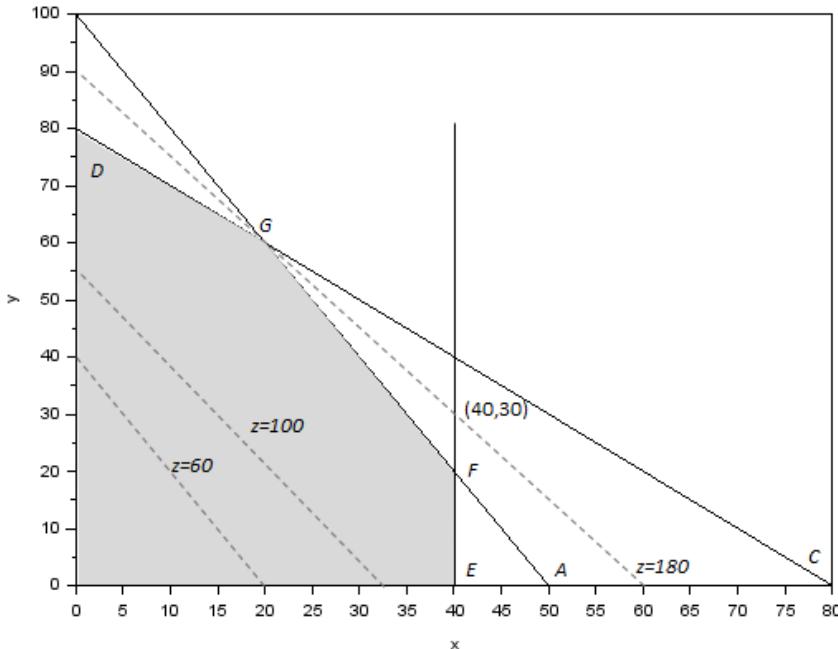
$$x_2 \geq 0 \quad (\text{Sign restriction}) \quad (4.16)$$

For a point  $(x_1, x_2)$  to be in the feasible region,  $(x_1, x_2)$  must satisfy all the inequalities (4.12)–(4.16). Note that the only points satisfying (4.15) and (4.16) lie in the first quadrant of the  $x_1 - x_2$  plane. This is indicated in Figure 2 by the arrows pointing to the right from the  $x_2$  axis and upward from the  $x_1$  axis. Thus, any point that is outside the first quadrant cannot be in the feasible region. This means that the feasible

region will be the set of points in the first quadrant that satisfies (4.12)–(4.14).

Our method for determining the set of points that satisfies a linear inequality will also identify those that meet (4.12)–(4.14). From Figure 2, we see that (4.12) is satisfied by all points below or on the line  $AB$  ( $AB$  is the line  $2x_1 + x_2 = 100$ ). Inequality (4.13) is satisfied by all points on or below the line  $CD$  ( $CD$  is the line  $x_1 + x_2 = 80$ ). Finally, (4.14) is satisfied by all points on or to the left of line  $EF$  ( $EF$  is the line  $x_1 = 40$ ). The side of a line that satisfies an inequality is indicated by the direction of the arrows in Figure 2.

From Figure 2, we see that the set of points in the first quadrant that satisfies (4.12), (4.13), and (4.14) is bounded by the five-sided polygon  $DGFEH$ . Any point on this polygon or in its interior is in the feasible region. Any other point fails to satisfy at least one of the inequalities (4.12)–(4.16). For example, the point  $(40, 30)$  lies outside  $DGFEH$  because it is above the line segment  $AB$ . Thus  $(40, 30)$  is infeasible, because it fails to satisfy (4.12).



**FIGURE 2.** Graphical Solution of Gepetto Problem (made with SCILAB)

An easy way to find the feasible region is to determine the set of infeasible points. Note that all points above line  $AB$  in Figure 2 are infeasible, because they fail to satisfy (4.12). Similarly, all points above  $CD$  are infeasible, because they fail to satisfy (4.13). Also, all points to the right of the vertical line  $EF$  are infeasible, because they fail to satisfy (4.14). After these points are eliminated from consideration, we are left with the feasible region ( $DGFEH$ ).

#### 4.2.2. Finding the Optimal Solution

Having identified the feasible region for the Gepetto problem, we now search for the optimal solution, which will be the point in the feasible region with the largest value of  $z = 3x_1 + 2x_2$ . To find the optimal solution, we need to graph a line on which all points have the same  $z$ -value. In a max problem, such a line is called an **isoprofit line** (in a min problem, an **isocost line**). To draw an isoprofit line, choose any point in the feasible region and calculate its  $z$ -value. Let us choose  $(20, 0)$ . For  $(20, 0)$ ,  $z = 3(20) + 2(0) = 60$ . Thus,  $(20, 0)$  lies on the isoprofit line  $z = 3x_1 + 2x_2 = 60$ . Rewriting  $3x_1 + 2x_2 = 60$  as  $x_2 = 30 - \frac{3}{2}x_1$ , we see that the isoprofit line  $x_1 + 2x_2 = 60$  has a slope of  $-\frac{3}{2}$ . Because all isoprofit lines are of the form  $3x_1 + 2x_2 = \text{constant}$ , all isoprofit lines have the same slope. This means that once we have drawn one isoprofit line, we can find all other isoprofit lines by moving parallel to the isoprofit line we have drawn.

It is now clear how to find the optimal solution to a two-variable LP. After you have drawn a single isoprofit line, generate other isoprofit lines by moving parallel to the drawn isoprofit line in a direction that increases  $z$  (for a max problem). After a point, the isoprofit lines will no longer intersect the feasible region. The last isoprofit line intersecting (touching) the feasible region defines the largest  $z$ -value of any point in the feasible region and indicates the optimal solution to the LP. In our problem, the objective function  $z = 3x_1 + 2x_2$  will increase if we move in a direction for which both  $x_1$  and  $x_2$  increase. Thus, we construct additional isoprofit lines by moving parallel to  $3x_1 + 2x_2 = 60$  in a northeast direction (upward and to the right). From Figure 2, we see that the isoprofit line passing through point  $G$  is the last isoprofit line to intersect the feasible region. Thus,  $G$  is the point in the feasible region

with the largest  $z$ -value and is therefore the optimal solution to the Gepetto problem. Note that point  $G$  is where the lines  $2x_1 + x_2 = 100$  and  $x_1 + x_2 \leq 80$  intersect. Solving these two equations simultaneously, we find that  $(x_1 = 20, x_2 = 60)$  is the optimal solution to the Gepetto problem. The optimal value of  $z$  may be found by substituting these values of  $x_1$  and  $x_2$  into the objective function. Thus, the optimal value of  $z$  is  $z = 3(20) + 2(60) = 180$ .

#### 4.2.3. Binding and Nonbinding Constraints

Once the optimal solution to an LP has been found, it is useful (see Chapters 5 and 6) to classify each constraint as being a binding constraint or a nonbinding constraint.

**DEFINITION** ■ A constraint is **binding** if the left-hand side and the right-hand side of the constraint are equal when the optimal values of the decision variables are substituted into the constraint. ■

Thus, (4.12) and (4.13) are binding constraints.

**DEFINITION** ■ A constraint is **nonbinding** if the left-hand side and the right-hand side of the constraint are unequal when the optimal values of the decision variables are substituted into the constraint. ■

Because  $x_1 = 20$  is less than 40, (4.14) is a nonbinding constraint.

#### 4.2.4. Convex Sets, Extreme Points, and LP

The feasible region for the Gepetto problem is an example of a convex set.

**DEFINITION** ■ A set of points  $S$  is a **convex set** if the line segment joining any pair of points in  $S$  is wholly contained in  $S$ . ■

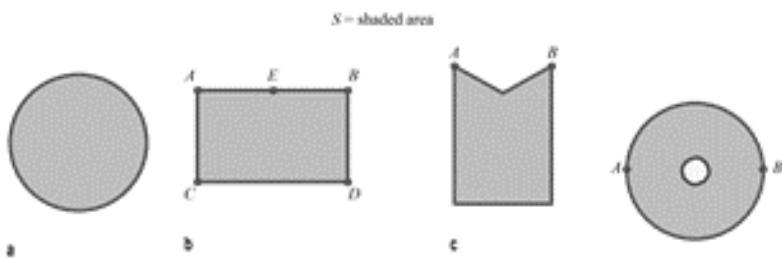
Figure 3 gives four illustrations of this definition. In Figures 3a and 3b, each line segment joining two points in  $S$  contains only points in  $S$ . Thus, in both these figures,  $S$  is convex. In Figures 3c and 3d,  $S$  is not convex. In each figure, points  $A$  and  $B$  are in  $S$ , but there are points on the line segment  $AB$  that are not contained in  $S$ . In our study of linear programming, a certain type of point in a convex set (called an *extreme point*) is of great interest.

**DEFINITION ■** For any convex set  $S$ , a point  $P$  in  $S$  is an **extreme point** if each line segment that lies completely in  $S$  and contains the point  $P$  has  $P$  as an endpoint of the line segment. ■

For example, in Figure 3a, each point on the circumference of the circle is an extreme point of the circle. In Figure 3b, points  $A$ ,  $B$ ,  $C$ , and  $D$  are extreme points of  $S$ . Although point  $E$  is on the boundary of  $S$  in Figure 3b,  $E$  is not an extreme point of  $S$ . This is because  $E$  lies on the line segment  $AB$  ( $AB$  lies completely in  $S$ ), and  $E$  is not an endpoint of the line segment  $AB$ . Extreme points are sometimes called corner points, because if the set  $S$  is a polygon, the extreme points of  $S$  will be the vertices, or corners, of the polygon.

The feasible region for the Gepetto problem is a convex set. This is no accident: It can be shown that the feasible region for any LP will be a convex set. From Figure 2, we see that the extreme points of the feasible region are simply points  $D, F, E, G$ , and  $H$ . It can be shown that the feasible region for any LP has only a finite number of extreme points. Also note that the optimal solution to the Gepetto problem (point  $G$ ) is an extreme point of the feasible region. It can be shown that *any LP that has an optimal solution has an extreme point that is optimal*. This result is very important, because it reduces the set of points that yield an optimal solution from the entire feasible region (which generally contains an *infinite* number of points) to the set of extreme points (a finite set).

For the Gepetto problem, it is easy to see why the optimal solution must be an extreme point of the feasible region. We note that  $z$  increases as we move isoprofit lines in a northeast direction, so the largest  $z$ -value in the feasible region must occur at some point  $P$  that has no points in the feasible region northeast of  $P$ . This means that the optimal solution must lie somewhere on the boundary of the feasible region  $DGFEH$ . The LP must have an extreme point that is optimal, because for any line segment on the boundary of the feasible region, the largest  $z$ -value on that line segment must be assumed at one of the endpoints of the line segment.



**FIGURE 3.** Convex and Nonconvex Sets

To see this, look at the line segment  $FG$  in Figure 2.  $FG$  is part of the line  $2x_1 + x_2 = 100$  and has a slope of  $-2$ . If we move along  $FG$  and decrease  $x_1$  by 1, then  $x_2$  will increase by 2, and the value of  $z$  changes as follows:  $3x_1$  goes down by  $3(1) = 3$ , and  $2x_2$  goes up by  $2(2) = 4$ . Thus, in total,  $z$  increases by  $4 - 3 = 1$ . This means that moving along  $FG$  in a direction of decreasing  $x_1$  increases  $z$ . Thus, the value of  $z$  at point  $G$  must exceed the value of  $z$  at any other point on the line segment  $FG$ .

A similar argument shows that for any objective function, the maximum value of  $z$  on a given line segment must occur at an endpoint of the line segment. Therefore, for any LP, the largest  $z$ -value in the feasible region must be attained at an endpoint of one of the line segments forming the boundary of the feasible region. In short, one of the extreme points of the feasible region must be optimal. (To test your understanding, show that if Gepetto's objective function were  $z = 6x_1 + x_2$ , point  $F$  would be optimal, whereas if Gepetto's objective function were  $z = x_1 + 6x_2$ , point  $D$  would be optimal.) Our proof that an LP always has an optimal extreme point depended heavily on the fact that both the objective function and the constraints were linear functions.

#### 4.2.5. The Graphical Solution of Minimization Problems

##### EXAMPLE 2. Phil Short Auto

Phil Short Auto manufactures luxury cars and trucks. The company believes that its most likely customers are high-income women and men. To reach these groups, Phil Short Auto has embarked on an ambitious TV advertising campaign and has decided to purchase 1-minute

commercial spots on two types of programs: comedy shows and football games. Each comedy commercial is seen by 7 million high-income women and 2 million high-income men. Each football commercial is seen by 2 million high-income women and 12 million high-income men. A 1-minute comedy ad costs \$50,000, and a 1-minute football ad costs \$100,000. Phil Short would like the commercials to be seen by at least 28 million high-income women and 24 million high-income men. Use linear programming to determine how Phil Short Auto can meet its advertising requirements at minimum cost.

**Solution.** Phil Short must decide how many comedy and football ads should be purchased, so the decision variables are

$$x_1 = \text{number of 1 - minute comedy ads purchased}$$

$$x_2 = \text{number of 1 - minute football ads purchased}$$

Then Phil Short wants to minimize total advertising cost (in thousands of dollars).

$$\begin{aligned} \text{Total advertising cost} &= \text{cost of comedy ads} + \text{cost of football ads} \\ &= \left( \frac{\text{cost}}{\text{comedy ad}} \right) (\text{total comedy ads}) + \left( \frac{\text{cost}}{\text{football ad}} \right) (\text{total football ads}) \\ &= 50x_1 + 100x_2 \end{aligned}$$

Thus, Phil Short's objective function is

$$\min z = 50x_1 + 100x_2 \quad (4.17)$$

Phil Short faces the following constraints:

**Constraint 1.** Commercials must reach at least 28 million high-income women.

**Constraint 2.** Commercials must reach at least 24 million high-income men.

To express Constraints 1 and 2 in terms of  $x_1$  and  $x_2$ , let HIW stand for high-income women viewers and HIM stand for high-income men viewers (in millions).

$$\begin{aligned}
HIW &= \left( \frac{HIW}{comedy ad} \right) \left( \frac{total}{comedy ads} \right) \\
&\quad + \left( \frac{HIW}{football ad} \right) \left( \frac{total}{football ads} \right) \\
&= 7x_1 + 2x_2
\end{aligned}$$

$$\begin{aligned}
HIM &= \left( \frac{HIM}{comedy ad} \right) \left( \frac{total}{comedy ads} \right) \\
&\quad + \left( \frac{HIM}{football ad} \right) \left( \frac{total}{football ads} \right) \\
&= 2x_1 + 12x_2
\end{aligned}$$

Constraint 1 may now be expressed as

$$7x_1 + 2x_2 \geq 28 \quad (4.18)$$

and Constraint 2 may be expressed as

$$2x_1 + 12x_2 \geq 24 \quad (4.19)$$

The sign restrictions  $x_1 \geq 0$  and  $x_2 \geq 0$  are necessary, so the Phil Short LP is given by:

$$\min z = 50x_1 + 100x_2 \quad (4.17)$$

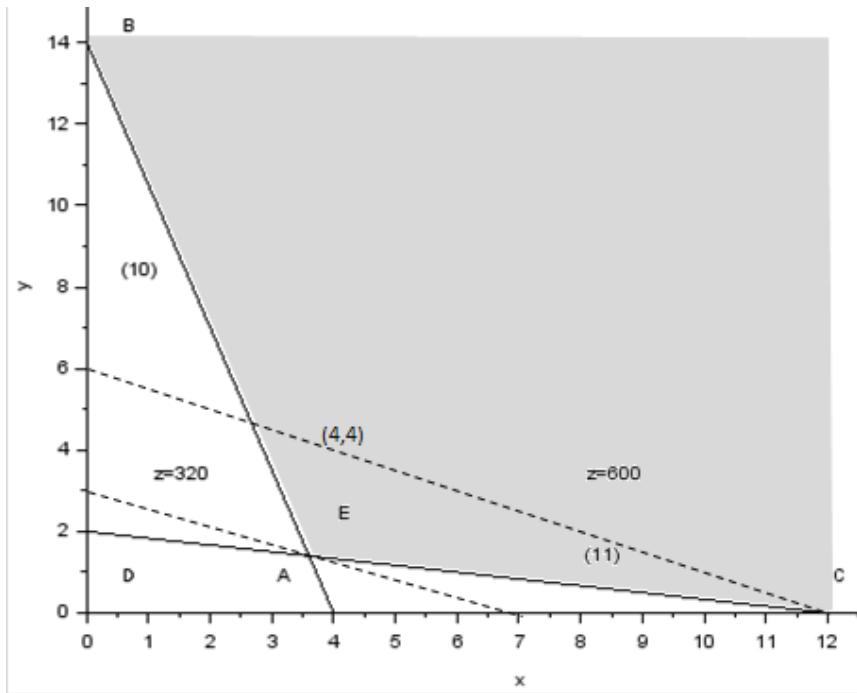
s.t.

$$7x_1 + 2x_2 \geq 28 \quad (HIW)$$

$$2x_1 + 12x_2 \geq 24 \quad (HIM)$$

$$x_1 \geq 0 \text{ and } x_2 \geq 0$$

This problem is typical of a wide range of LP applications in which a decision maker wants to minimize the cost of meeting a certain set of requirements. To solve this LP graphically, we begin by graphing the feasible region (Figure 4). Note that (4.18) is satisfied by points on or above the line  $AB$  ( $AB$  is part of the line  $7x_1 + 2x_2 \geq 28$ ) and that (4.19) is satisfied by the points on or above the line  $CD$  ( $CD$  is part of the line  $2x_1 + 12x_2 \geq 24$ ). From Figure 4, we see that the only first-quadrant points satisfying both (4.18) and (4.19) are the points in the shaded region bounded by the  $x_1$  axis, CEB, and the  $x_2$  axis.



**FIGURE 4.** Graphical Solution of Phil Short Problem (made with *SCILAB*)

Like the Gepetto problem, the Phil Short problem has a convex feasible region, but the feasible region for Phil Short, unlike Gepetto's, contains points for which the value of at least one variable can assume arbitrarily large values. Such a feasible region is called an **un-bounded feasible region**.

Because Phil Short wants to minimize total advertising cost, the optimal solution to the problem is the point in the feasible region with the smallest  $z$ -value. To find the optimal solution, we need to draw an isocost line that intersects the feasible region. An isocost line is any line on which all points have the same  $z$ -value (or same cost). We arbitrarily choose the isocost line passing through the point  $(x_1 = 4, x_2 = 4)$ . For this point,  $z = 50(4) + 100(4) = 600$ , and we graph the isocost line  $z = 50x_1 + 100x_2 = 600$ .

We consider lines parallel to the isocost line  $50(4) + 100(4) = 600$  in the direction of decreasing  $z$  (southwest). The last point in the feasible region that intersects an isocost line will be the point in the feasible

region having the smallest  $z$ -value. From Figure 4, we see that point  $E$  has the smallest  $z$ -value of any point in the feasible region; this is the optimal solution to the Phil Short problem. Note that point  $E$  is where the lines  $7x_1 + 2x_2 \geq 28$  and  $2x_1 + 12x_2 \geq 24$  intersect. Simultaneously solving these equations yields the optimal solution ( $x_1 = 3.6, x_2 = 1.4$ ). The optimal  $z$ -value can then be found by substituting these values of  $x_1$  and  $x_2$  into the objective function. Thus, the optimal  $z$ -value is  $z = 50(3.6) + 100(1.4) = 320 = \$320,000$ . Because at point  $E$  both the HIW and HIM constraints are satisfied with equality, both constraints are binding.

#### 4.2.6. Problems

1. Messy Chemical manufactures three chemicals: A, B and C. These chemicals are produced via two production processes: 1 and 2. Running process 1 for an hour costs \$4 and yields 3 units of A, 1 of B, and 1 of C. Running process 2 for an hour costs \$1 and produces 1 unit of A and 1 of B. To meet customer demands, at least 10 units of A, 5 of B, and 3 of C must be produced daily. Graphically determine a daily production plan that minimizes the cost of meeting Messy Chemical's daily demands.
2. Farmer Mary Jane owns 45 acres of land. She is going to plant each with wheat or corn. Each acre planted with wheat yields \$200 profit; each with corn yields \$300 profit. The labor and fertilizer used for each acre are given in Table 1. One hundred workers and 120 tons of fertilizer are available. Use linear programming to determine how Mary Jane can maximize profits from her land.

**TABLE 1**

|            | Wheat     | Corn      |
|------------|-----------|-----------|
| Labor      | 3 workers | 2 workers |
| Fertilizer | 2 tons    | 4 tons    |

#### 4.3. Special Cases

The Gepetto and Phil Short problems each had a unique optimal solution. In this section, we encounter three types of LPs that do not have unique optimal solutions.

1. Some LPs have an infinite number of optimal solutions (alternative or multiple optimal solutions).
2. Some LPs have no feasible solutions (infeasible LPs).
3. Some LPs are unbounded: There are points in the feasible region with arbitrarily large (in a max problem) z-values.

#### 4.3.1. Alternative or Multiple Optimal Solutions

##### EXAMPLE 3. Alternative Optimal Solutions

An auto company manufactures cars and trucks. Each vehicle must be processed in the paint shop and body assembly shop. If the paint shop were only painting trucks, then 40 per day could be painted. If the paint shop were only painting cars, then 60 per day could be painted. If the body shop were only producing cars, then it could process 50 per day. If the body shop were only producing trucks, then it could process 50 per day. Each truck contributes \$300 to profit, and each car contributes \$200 to profit. Use linear programming to determine a daily production schedule that will maximize the company's profits.

**Solution.** The company must decide how many cars and trucks should be produced daily. This leads us to define the following decision variables:

$$x_1 = \text{number of trucks produced daily}$$

$$x_2 = \text{number of cars produced daily}$$

The company's daily profit (in hundreds of dollars) is  $3x_1 + 2x_2$ , so the company's objective function may be written as

$$\max z = 3x_1 + 2x_2 \quad (4.19)$$

The company's two constraints are the following:

**Constraint 1.** The fraction of the day during which the paint shop is busy is less than or equal to 1.

**Constraint 2.** The fraction of the day during which the body shop is busy is less than or equal to 1.

We have

$$\text{Fraction of day paint shop works on trucks} = \left( \frac{\text{fraction of day}}{\text{truck}} \right) \left( \frac{\text{trucks}}{\text{day}} \right)$$

$$= \frac{1}{40} x_2$$

$$\text{Fraction of day paint shop works on cars} = \frac{1}{60} x_2$$

$$\text{Fraction of day body shop works on trucks} = \frac{1}{50} x_1$$

$$\text{Fraction of day body shop works on cars} = \frac{1}{50} x_2$$

Thus, Constraint 1 may be expressed by

$$\frac{1}{40} x_1 + \frac{1}{60} x_2 < 1 \quad (\text{Paint shop constraint}) \tag{4.20}$$

and Constraint 2 may be expressed by

$$\frac{1}{50} x_1 + \frac{1}{50} x_2 < 1 \quad (\text{Body shop constraint}) \tag{4.21}$$

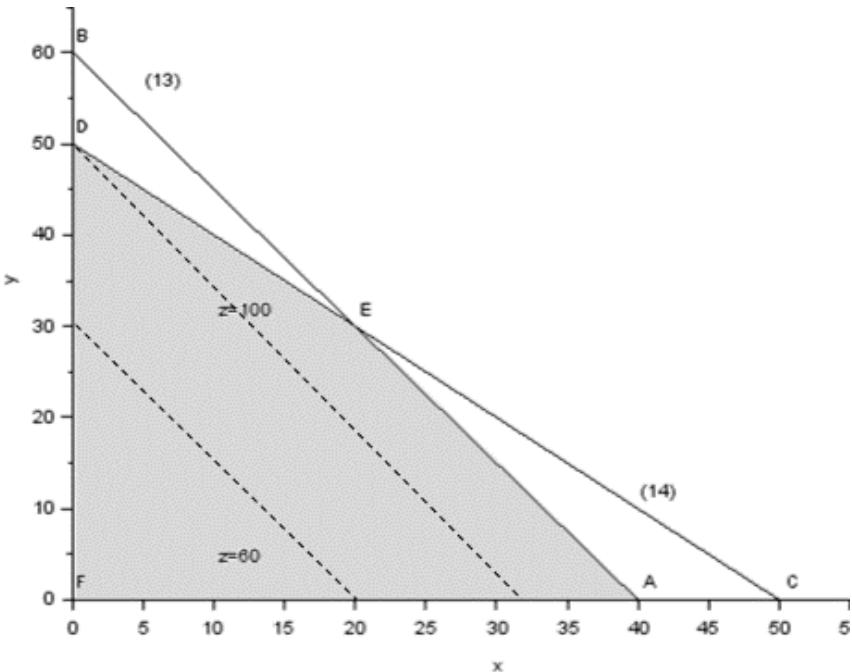
Because  $x_1 > 0$  and  $x_2 > 0$  must hold, the relevant LP is

$$\begin{aligned} \max z &= 3x_1 + 2x_2 \\ &\quad (4.19) \end{aligned}$$

s.t. (4.20) and (4.21)

$$\begin{aligned} \frac{1}{40} x_1 + \frac{1}{60} x_2 &< 1 \\ \frac{1}{50} x_1 + \frac{1}{50} x_2 &< 1 \\ x_1, x_2 &> 0 \end{aligned}$$

The feasible region for this LP is the shaded region in Figure 5 bounded by  $AEDF$ .



**FIGURE 5.** Graphical Solution of Example 3 (made with SCILAB)

For our isoprofit line, we choose the line passing through the point  $(20, 0)$ . Because  $(20, 0)$  has a  $z$ -value of  $3(20) + 2(0) = 60$ , this yields the isoprofit line  $z = 3x_1 + 2x_2 = 60$ . Examining lines parallel to this isoprofit line in the direction of increasing  $z$  (northeast), we find that the last “point” in the feasible region to intersect an isoprofit line is the entire line segment  $AE$ . This means that any point on the line segment  $AE$  is optimal. We can use any point on  $AE$  to determine the optimal  $z$ -value. For example, point  $A$ ,  $(40, 0)$ , gives  $z = 3(40) = 120$ .

In summary, the auto company’s LP has an infinite number of optimal solutions, or multiple or alternative optimal solutions. This is indicated by the fact that as an isoprofit line leaves the feasible region, it will intersect an entire line segment corresponding to the binding constraint (in this case,  $AE$ ).

From our current example, it seems reasonable (and can be shown to be true) that if two points ( $A$  and  $E$  here) are optimal, then any point on the line segment joining these two points will also be optimal.

If an alternative optimum occurs, then the decision maker can use a secondary criterion to choose between optimal solutions. The auto company's managers might prefer point  $A$  because it would simplify their business (and still allow them to maximize profits) by allowing them to produce only one type of product (trucks).

The technique of **goal programming** (see Section 5.7) is often used to choose among alternative optimal solutions.

#### 4.3.2. Infeasible LP

It is possible for an LP's feasible region to be empty (contain no points), resulting in an infeasible LP. Because the optimal solution to an LP is the best point in the feasible region, an infeasible LP has no optimal solution.

##### EXAMPLE 4. Infeasible LP

Suppose that auto dealers require that the auto company in Example 3 produce at least 30 trucks and 20 cars. Find the optimal solution to the new LP.

**Solution.** After adding the constraints  $x_1 > 30$  and  $x_2 > 20$  to the LP of Example 3, we obtain the following LP:

$$\max z = 3x_1 + 2x_2$$

$$\text{s.t.} \quad \frac{1}{40}x_1 + \frac{1}{60}x_2 \leq 1 \quad (4.22)$$

$$\frac{1}{50}x_1 + \frac{1}{60}x_2 \leq 1 \quad (4.23)$$

$$x_1 + 50x_2 \geq 30 \quad (4.23)$$

$$x_2 \geq 20 \quad (4.25)$$

$$x_1, x_2 \geq 0$$

The graph of the feasible region for this LP is Figure 6.

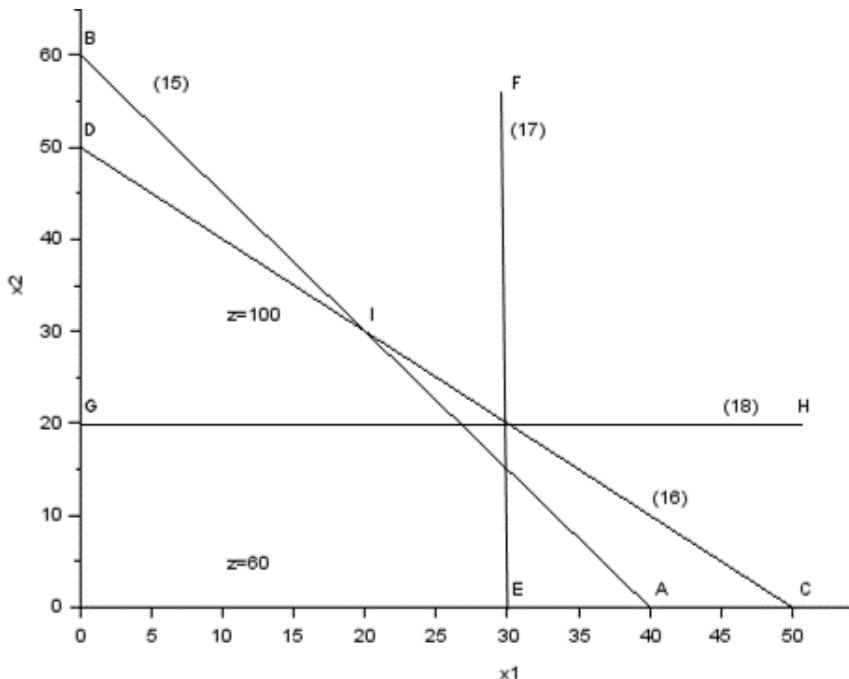
Constraint (4.22) is satisfied by all points on or below AB (AB is  $\frac{1}{40}x_1 + \frac{1}{60}x_2 \leq 1$ ).

Constraint (4.24) is satisfied by all points on or below CD (CD is  $\frac{1}{50}x_1 + \frac{1}{60}x_2 \leq 1$ ).

Constraint (4.24) is satisfied by all points on or to the right of EF (EF is  $x_1 = 30$ ).

Constraint (4.25) is satisfied by all points on or above GH (GH is  $x_2 = 20$ ).

From Figure 6 it is clear that no point satisfies all of (4.22)–(4.25). This means that Example 4 has an empty feasible region and is an infeasible LP.



**FIGURE 6.** An Empty Feasible Region (Infeasible LP) (made with SCILAB)

In Example 4, the LP is infeasible because producing 30 trucks and 20 cars requires more paint shop time than is available.

### 4.3.3. Unbounded LP

Our next special LP is an unbounded LP. For a max problem, an unbounded LP occurs if it is possible to find points in the feasible region with arbitrarily large  $z$ -values, which corresponds to a decision maker earning arbitrarily large revenues or profits. This would indicate that an unbounded optimal solution should not occur in a correctly formulated LP. Thus, if the reader ever solves an LP on the computer and finds that the LP is unbounded, then an error has probably been made in formulating the LP or in inputting the LP into the computer.

For a minimization problem, an LP is unbounded if there are points in the feasible region with arbitrarily small  $z$ -values. When graphically solving an LP, we can spot an unbounded LP as follows: A max problem is unbounded if, when we move parallel to our original isoprofit line in the direction of increasing  $z$ , we never entirely leave the feasible region. A minimization problem is unbounded if we never leave the feasible region when moving in the direction of decreasing  $z$ .

#### EXAMPLE 5. Unbounded LP

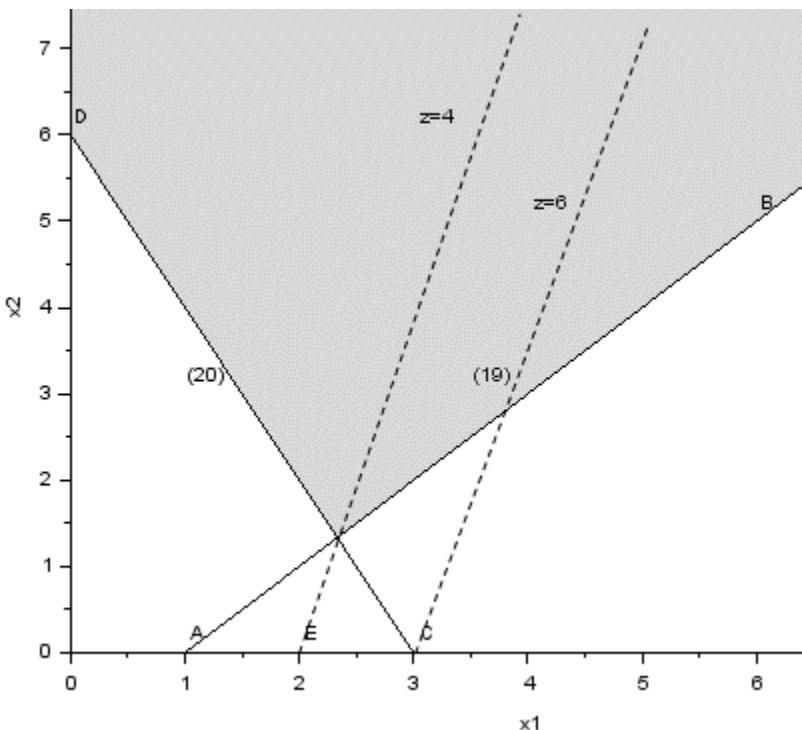
Graphically solve the following LP:

$$\max z = 2x_1 - x_2 \quad (4.26)$$

$$\text{s.t.} \quad (4.27)$$

$$\begin{aligned} x_1 - x_2 &\leq 1 \\ 2x_1 + x_2 &\geq 6 \\ x_1, x_2 &> 0 \end{aligned}$$

**Solution.** From Figure 7, we see that (4.26) is satisfied by all points on or above  $AB$  ( $AB$  is the line  $x_1 - x_2 = 1$ ). Also, (4.27) is satisfied by all points on or above  $CD$  ( $CD$  is  $2x_1 + x_2 = 6$ ).



**FIGURE 7.** An Unbounded LP (made with *SCILAB*)

Thus, the feasible region for Example 5 is the (shaded) unbounded region in Figure 7, which is bounded only by the  $x_2$  axis, line segment  $DE$ , and the part of line  $AB$  beginning at  $E$ . To find the optimal solution, we draw the isoprofit line passing through  $(2, 0)$ . This isoprofit line has  $z = 2x_1 - x_2 = 2(2) - 0 = 4$ . The direction of increasing  $z$  is to the southeast (this makes  $x_1$  larger and  $x_2$  smaller). Moving parallel to  $z = 2x_1 - x_2$  in a southeast direction, we see that any isoprofit line we draw will intersect the feasible region. (This is because any isoprofit line is steeper than the line  $x_1 - x_2 = 1$ .)

Thus, there are points in the feasible region that have arbitrarily large  $z$ -values. For example, if we wanted to find a point in the feasible region that had  $z \geq 1,000,000$ , we could choose any point in the feasible region that is southeast of the isoprofit line  $z = 1,000,000$ .

From the discussion in the last two sections, we see that every LP with two variables must fall into one of the following four cases:

**Case 1.** The LP has a unique optimal solution.

**Case 2.** The LP has alternative or multiple optimal solutions: Two or more extreme points are optimal, and the LP will have an infinite number of optimal solutions.

**Case 3.** The LP is infeasible: The feasible region contains no points.

**Case 4.** The LP is unbounded: There are points in the feasible region with arbitrarily large z-values (max problem) or arbitrarily small z-values (min problem).

In Chapter 4, we show that every LP (not just LPs with two variables) must fall into one of Cases 1–4.

In the rest of this chapter, we lead the reader through the formulation of several more complicated linear programming models. The most important step in formulating an LP model is the proper choice of decision variables. If the decision variables have been properly chosen, the objective function and constraints should follow without much difficulty. Trouble in determining an LP's objective function and constraints is usually the result of an incorrect choice of decision variables.

#### 4.3.4. Problems

Identify which of Cases 1–4 apply to each of the following LPs:

$$1. \max z = x_1 + x_2$$

s.t.

$$\begin{aligned}x_1 + x_2 &\leq 4 \\x_1 - x_2 &\geq 5 \\x_1, x_2 &\geq 0\end{aligned}$$

$$2. \max z = 4x_1 + x_2$$

s.t.

$$\begin{aligned}8x_1 + 2x_2 &\leq 16 \\5x_1 + 2x_2 &\leq 12 \\x_1, x_2 &\geq 0\end{aligned}$$

$$3. \max z = -x_1 + 3x_2$$

s.t.

$$\begin{aligned}x_1 - 2x_2 &\leq 4 \\x_1 + 2x_2 &\geq 4 \\x_1, x_2 &\geq 0\end{aligned}$$

$$4. \max z = 3x_1 + x_2$$

s.t.

$$\begin{aligned}2x_1 + 3x_2 &\leq 6 \\x_1 + 3x_2 &\leq 9 \\x_1, x_2 &\geq 0\end{aligned}$$

5. Money manager Will Milkem deals with French currency (the franc) and American currency (the dollar). At 12 midnight, he can buy francs by paying 0.25 dollars per franc and dollars by paying 3 francs per dollar. Let  $x_1$  = number of dollars bought (by paying francs) and  $x_2$  = number of francs bought (by paying dollars). Assume that both types of transactions take place simultaneously, and the only constraint is that at 12:01 A.M. Will must have a nonnegative number of francs and dollars.

- Formulate an LP that enables Will to maximize the number of dollars he has after all transactions are completed.
- Graphically solve the LP and comment on the answer.

#### 4.4. A Work-Scheduling Problem

Many applications of linear programming involve determining the minimum-cost method for satisfying workforce requirements. The following example illustrates the basic features common to many of these applications.

##### EXAMPLE 6. USPS Problem

A USPS office requires different numbers of full-time employees on different days of the week. The number of full-time employees required on each day is given in Table 2. Union rules state that each full-time

employee must work five consecutive days and then receive two days off. For example, an employee who works Monday to Friday must be off on Saturday and Sunday. The USPS office wants to meet its daily requirements using only full-time employees. Formulate an LP that the USPS office can use to minimize the number of full-time employees who must be hired.

**Solution.** Before giving the correct formulation of this problem, let's begin by discussing an incorrect solution. Many students begin by defining  $x_i$  to be the number of employees working on day  $i$  (day 1 = Monday, day 2 = Tuesday, and so on). Then they reason that (number of full-time employees) = (number of employees working on Monday) + (number of employees working on Tuesday) + ⋯ + (number of employees working on Sunday). This reasoning leads to the following objective function:

$$\min z = x_1 + x_2 + \dots + x_6 + x_7$$

To ensure that the post office has enough full-time employees working on each day, they add the constraints  $x_i \geq$  (number of employees required on day  $i$ ). For example, for Monday add the constraint  $x_1 \geq 17$ . Adding the sign restrictions  $x_i \geq 0$  ( $i = 1, 2, \dots, 7$ ) yields the following LP:

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

s.t.

$$\begin{aligned} x_1 &\geq 17 \\ x_2 &\geq 13 \\ x_3 &\geq 15 \\ x_4 &\geq 19 \\ x_5 &\geq 14 \\ x_6 &\geq 16 \\ x_7 &\geq 1 \\ x_i &\geq 0 \quad (i = 1, 2, \dots, 7) \end{aligned}$$

There are at least two flaws in this formulation. First, the objective function is not the number of full-time USPS office employees. The current objective function counts each employee five times, not once.

For example, each employee who starts work on Monday works Monday to Friday and is included in  $x_1, x_2, x_3, x_4$ , and  $x_5$ . Second, the variables  $x_1, x_2, \dots, x_7$  are interrelated, and the interrelation between the variables is not captured by the current set of constraints. For example, some of the people who are working on Monday (the  $x_1$  people) will be working on Tuesday. This means that  $x_1$  and  $x_2$  are interrelated, but our constraints do not indicate that the value of  $x_1$  has any effect on the value of  $x_2$ .

**TABLE 2.** Requirements for Post Office

| Day           | Number of Full-time Employees Required |
|---------------|--|
| 1 = Monday    | 17                                     |
| 2 = Tuesday   | 13                                     |
| 3 = Wednesday | 15                                     |
| 4 = Thursday  | 19                                     |
| 5 = Friday    | 14                                     |
| 6 = Saturday  | 16                                     |
| 7 = Sunday    | 11                                     |

$x_i$  = number of employees beginning work on day  $i$

The key to correctly formulating this problem is to realize that the USPS office's primary decision is not how many people are working each day but rather how many people begin work on each day of the week. With this in mind, we define

For example,  $x_1$  is the number of people beginning work on Monday (these people work Monday to Friday). With the variables properly defined, it is easy to determine the correct objective function and constraints. To determine the objective function, note that (number of full-time employees) = (number of employees who start work on Monday) + (number of employees who start work on Tuesday) +  $\dots$  + (number of employees who start work on Sunday). Because each employee begins work on exactly one day of the week, this expression

does not double-count employees. Thus, when we correctly define the variables, the objective function is

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

The USPS office must ensure that enough employees are working on each day of the week. For example, at least 17 employees must be working on Monday. Who is working on Monday? Everybody except the employees who begin work on Tuesday or on Wednesday (they get, respectively, Sunday and Monday, and Monday and Tuesday off). This means that the number of employees working on Monday is  $x_1 + x_4 + x_5 + x_6 + x_7$ . To ensure that at least 17 employees are working on Monday, we require that the constraint

$$x_1 + x_4 + x_5 + x_6 + x_7 \geq 17$$

be satisfied. Adding similar constraints for the other six days of the week and the sign restrictions  $x_i \geq 0$  ( $i = 1, 2, \dots, 7$ ) yields the following formulation of the USPS office's problem:

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

s.t.

$$\begin{aligned} x_1 &+ x_4 + x_5 + x_6 + x_7 \geq 17 && (\text{Mon constraint}) \\ x_1 + x_2 &+ x_5 + x_6 + x_7 \geq 13 && (\text{Tue constraint}) \\ x_1 + x_2 + x_3 &+ x_6 + x_7 \geq 15 && (\text{Wed constraint}) \\ x_1 + x_2 + x_3 + x_4 &+ x_6 + x_7 \geq 19 && (\text{Thu constraint}) \\ x_1 + x_2 + x_3 + x_4 + x_5 &\geq 14 && (\text{Fri constraint}) \\ x_2 + x_3 + x_4 + x_5 + x_6 &\geq 16 && (\text{Sat constraint}) \\ x_3 + x_4 + x_5 + x_6 + x_7 &\geq 11 && (\text{Sun constraint}) \\ x_i &\geq 0 (i = 1, 2, \dots, 7) && (\text{Sign restrictions}) \end{aligned}$$

The optimal solution to this LP is  $z = \frac{67}{3}$ ,  $x_1 = \frac{4}{3}$ ,  $x_2 = \frac{10}{3}$ ,  $x_3 = 2$ ,  $x_4 = \frac{22}{3}$ ,  $x_5 = 0$ ,  $x_6 = \frac{10}{3}$ ,  $x_7 = 5$ . Because we are only allowing full-time employees, however, the variables must be integers, and the Divisibility Assumption is not satisfied. To find a reasonable answer in which all variables are integers, we could try to round the fractional variables up, yielding the feasible solution  $z = 25$ ,  $x_1 = 2$ ,  $x_2 = 4$ ,  $x_3 = 2$ ,  $x_4 = 8$ ,  $x_5 = 0$ ,  $x_6 = 4$ ,  $x_7 = 5$ . It turns out, however, that integer

programming can be used to show that an optimal solution to the post office problem is  $z = 23, x_1 = 4, x_2 = 4, x_3 = 2, x_4 = 6, x_5 = 0, x_6 = 4, x_7 = 3$ . Notice that there is no way that the optimal linear programming solution could have been rounded to obtain the optimal all-integer solution.

Baker (1974) has developed an efficient technique (that does not use linear programming) to determine the minimum number of employees required when each worker receives two consecutive days off.

#### 4.4.1. Problems

1. In the post office example, suppose that each full-time employee works 8 hours per day. Thus, Monday's requirement of 17 workers may be viewed as a requirement of  $8(17) = 136$  hours. The post office may meet its daily labor requirements by using both full-time and part-time employees. During each week, a full-time employee works 8 hours a day for five consecutive days, and a part-time employee works 4 hours a day for five consecutive days. A full-time employee costs the post office \$15 per hour, whereas a part-time employee (with reduced fringe benefits) costs the post office only \$10 per hour. Union requirements limit part-time labor to 25% of weekly labor requirements. Formulate an LP to minimize the post office's weekly labor costs.
2. Suppose that the post office can force employees to work one day of overtime each week. For example, an employee whose regular shift is Monday to Friday can also be required to work on Saturday. Each employee is paid \$50 a day for each of the first five days worked during a week and \$62 for the overtime day (if any). Formulate an LP whose solution will enable the post office to minimize the cost of meeting its weekly work requirements.

#### 4.5. A Capital Budgeting Problem

In this section, we discuss how linear programming can be used to determine optimal financial decisions. This section considers a simple capital budgeting model.

We first explain briefly the concept of net present value (NPV), which can be used to compare the desirability of different investments. Time 0 is the present.

Suppose investment 1 requires a cash outlay of \$10,000 at time 0 and a cash outlay of \$14,000 two years from now and yields a cash flow of \$24,000 one year from now. Investment 2 requires a \$6,000 cash outlay at time 0 and a \$1,000 cash outlay two years from now and yields a cash flow of \$8,000 one year from now. Which investment would you prefer?

Investment 1 has a net cash flow of

$$-10,000 + 24,000 - 14,000 = \$0$$

and investment 2 has a net cash flow of

$$-6,000 + 8,000 - 1,000 = \$1,000$$

On the basis of net cash flow, investment 2 is superior to investment 1. When we compare investments on the basis of net cash flow, we are assuming that a dollar received at any point in time has the same value. This is not true! Suppose that there exists an investment (such as a money market fund) for which \$1 invested at a given time will yield (with certainty)  $(1 + r)$  one year later. We call  $r$  the annual interest rate. Because \$1 now can be transformed into  $(1 + r)$  one year from now, we may write

$$\$1 \text{ now} = \$1(1 + r) \text{ one year from now}$$

Applying this reasoning to the  $\$1(1 + r)$  obtained one year from now shows that

$$\begin{aligned}\$1 \text{ now} &= \$1(1 + r) \text{ one year from now} \\ &= \$1(1 + r)^2 \text{ two years from now}\end{aligned}$$

and

$$\$1 \text{ now} = \$1(1 + r)^k \text{ } k \text{ years from now}$$

Dividing both sides of this equality by  $(1 + r)^k$  shows that

$$\$1 \text{ received } k \text{ years from now} = \$1(1 + r)^{-k} \text{ now}$$

In other words, a dollar received  $k$  years from now is equivalent to receiving  $\$(1 + r)^{-k}$  now.

We can use this idea to express all cash flows in terms of time 0 dollars (this process is called discounting cash flows to time 0). Using discounting, we can determine the total value (in time 0 dollars) of the cash flows for any investment. The total value (in time 0 dollars) of the cash flows for any investment is called the net present value, or NPV, of the investment. The NPV of an investment is the amount by which the investment will increase the firm's value (as expressed in time 0 dollars).

Assuming  $r = 0.20$ , we can compute the NPV for investments 1 and 2.

$$\begin{aligned} \text{NPV of investment 1} &= -10,000 + \frac{24,000}{1+0.20} - \frac{14,000}{(1+0.20)^2} \\ &= \$277.78 \end{aligned}$$

This means that if a firm invested in investment 1, then the value of the firm (in time 0 dollars) would increase by \$277.78. For investment 2,

$$\begin{aligned} \text{NPV of investment 2} &= -6,000 + \frac{8,000}{1 + 0.20} - \frac{1,000}{(1 + 0.20)^2} \\ &= -\$27.78 \end{aligned}$$

If a firm invested in investment 2, then the value of the firm (in time 0 dollars) would be reduced by \$27.78.

Thus, the NPV concept says that investment 1 is superior to investment 2. This conclusion is contrary to the one reached by comparing the net cash flows of the two investments. Note that the comparison between investments often depends on the value of  $r$ . For example, the reader is asked to show in Problem 1 at the end of this section that for  $r = 0.02$ , investment 2 has a higher NPV than investment 1. Of course, our analysis assumes that the future cash flows of an investment are known with certainty.

#### 4.5.1. Computing NPV with Excel

If we receive a cash flow of  $c_t$  in  $t$  years from now ( $t = 1, 2, \dots, T$ ) and we discount cash flows at a rate  $r$ , then the NPV of our cash flows is given by

$$\sum_{t=1}^{t=T} \frac{c_t}{(1-r)^t}$$

The basic idea is that \$1 today equals  $(1 + r)$  a year from now, so

$$\frac{1}{1+r} \text{ today} = \$1 \text{ a year from now}$$

The Excel function =NPV makes this computation easy. The syntax is

=NPV ( $r$ , range of cash flows)

The formula assumes that cash flows occur at the end of the year.

Projects with  $NPV > 0$  add value to the company, while projects with negative  $NPV$  reduce the company's value.

We illustrate the computation of NPV in the file NPV.xls.

### **EXAMPLE 7. Computing NPV**

For a discount rate of 15%, consider a project with the cash flows shown in Figure 8.

- Compute project NPV if cash flows are at the end of the year.
- Compute project NPV if cash flows are at the beginning of the year.
- Compute project NPV if cash flows are at the middle of the year.

### **Solution**

- We enter in cell C7 the formula

=NPV (C1,C4:I4)

and obtain \$375.06.

- Because all cash flows are received a year earlier, we multiply each cash flow's value by  $(1 + 1.15)$ , so the answer is obtained in C8 with formula

= (1 + C1) • C7

NPV is now larger: \$431.32.

We checked this in cell D8 with the formula

$$= C4 + NPV(C1,D4:I4)$$

- c. Because all cash flows are received six months earlier, we multiply each cash flow's value by  $Y1 - .15$ . NPV is now computed in C9 with the formula

$$= (1.15)^{0.5} \bullet C7$$

Now NPV is \$402.21.

| A                 | B             | C        | D        | E   | F    | G    | H   |
|-------------------|---------------|----------|----------|-----|------|------|-----|
| 1 dr              | 0.15          |          |          |     |      |      |     |
| 2                 |               |          |          |     |      |      |     |
| 3 Time            |               | 1        | 2        | 3   | 4    | 5    | 6   |
| 4                 |               | -400     | 200      | 600 | -900 | 1000 | 250 |
| 5                 |               |          |          |     |      |      |     |
| 6                 |               |          |          |     |      |      |     |
| 7 end of year     | end of yr.    | \$375.06 |          |     |      |      |     |
| 8 beginning of yr | beg. Of yr.   | \$431.32 | \$431.32 |     |      |      |     |
| 9 middle of year  | middle of yr. | \$402.21 |          |     |      |      |     |
| 10                |               |          |          |     |      |      |     |

With this background information, we are ready to explain how linear programming can be applied to problems in which limited investment funds must be allocated to investment projects. Such problems are called capital budgeting problems.

#### EXAMPLE 8. Project Selection

Sunshine Oil Company is considering five different investment opportunities. The cash out-flows and net present values (in millions of dollars) are given in Table 3. Sunshine Oil has \$40 million available for investment now (time 0); it estimates that one year from now (time 1) \$20 million will be available for investment. Sunshine Oil may purchase any fraction of each investment. In this case, the cash outflows and NPV are adjusted accordingly. For example, if Sunshine Oil purchases one-fifth of investment 3, then a cash outflow of  $\frac{1}{5}(5) = \$1$  million would be required at time 0, and a cash outflow of  $\frac{1}{5}(5) = \$1$  million would

be required at time 1. The one-fifth share of investment 3 would yield an NPV of  $\frac{1}{5}(16) = \$3.2$  million. Sunshine Oil wants to maximize the NPV that can be obtained by investing in investments 1–5. Formulate an LP that will help achieve this goal. Assume that any funds left over at time 0 cannot be used at time 1.

**Solution.** Sunshine Oil must determine what fraction of each investment to purchase. We define

$$x_i = \text{fraction of investment } i \text{ purchased by Sunshine Oil, } (i = 1, 2, 3, 4, 5)$$

Star's goal is to maximize the NPV earned from investments. Now, (total NPV) = (NPV earned from investment 1) + (NPV earned from investment 2) + ⋯ + (NPV earned from investment 5). Note that

NPV from investment 1 = NPV from investment 1(fraction of investment 1 purchased)  
 $13x_1$

Applying analogous reasoning to investments 2–5 shows that Sunshine Oil wants to maximize

$$z = 13x_1 + 16x_2 + 16x_3 + 14x_4 + 39x_5 \quad (4.28)$$

Sunshine Oil's constraints may be expressed as follows:

**Constraint 1.** Star cannot invest more than \$40 million at time 0.

**Constraint 2.** Star cannot invest more than \$20 million at time 1.

**Constraint 3.** Star cannot purchase more than 100% of investment  $i$  ( $i = 1, 2, 3, 4, 5$ ).

To express Constraint 1 mathematically, note that (dollars invested at time 0) = (dollars invested in investment 1 at time 0) + (dollars invested in investment 2 at time 0) + ⋯ + (dollars invested in investment 5 at time 0). Also, in millions of dollars,

$$\begin{aligned} & \text{Dollars invested in investment 1} \\ &= (\text{dollars required for investment 1 at time 0}) (\text{fraction of investment 1 purchased}) \\ &= 11x_1 \end{aligned}$$

**TABLE 3.** Cash Flows and Net Present Value for Investments in Capital Budgeting

|                     | Investment (\$) |    |    |    |    |
|---------------------|-----------------|----|----|----|----|
|                     | 1               | 2  | 3  | 4  | 5  |
| Time 0 cash outflow | 11              | 53 | 5  | 5  | 29 |
| Time 1 cash outflow | 3               | 6  | 5  | 1  | 34 |
| NPV                 | 13              | 16 | 16 | 14 | 39 |

Similarly, for investments 2–5,

$$\text{Dollars invested at time } 0 = 11x_1 + 53x_2 + 5x_3 + 5x_4 + 29x_5$$

Then Constraint 1 reduces to

$$11x_1 + 53x_2 + 5x_3 + 5x_4 + 29x_5 < 40 \quad (\text{Time 0 constraint}) \quad (4.29)$$

Constraint 2 reduces to

$$3x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 < 20 \quad (\text{Time 1 constraint}) \quad (4.30)$$

Constraints 3–7 may be represented by

$$x_i < 1 \quad (i = 1, 2, 3, 4, 5) \quad (4.31-4.37)$$

Combining (4.31)–(4.37) with the sign restrictions  $x_i > 0$  ( $i = 1, 2, 3, 4, 5$ ) yields the following LP:

$$\max z = 13x_1 + 16x_2 + 16x_3 + 14x_4 + 39x_5$$

s.t.

$$11x_1 + 53x_2 + 5x_3 + 5x_4 + 29x_5 \leq 40 \quad (\text{Time 0 constraint})$$

$$3x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 \leq 20 \quad (\text{Time 1 constraint})$$

$$x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 \leq 1$$

$$3x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 \leq 1$$

$$3x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 \leq 1$$

$$3x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 \leq 1$$

$$3x_1 + 6x_2 + 5x_3 + x_4 + 34x_5 \leq 1$$

$$x_i \geq 0 (i = 1, 2, & 3, 4, 5)$$

The optimal solution to this LP is  $x_1 = x_3 = x_4 = 1$ ,  $x_2 = 0.201$ ,  $x_5 = 0.288$ ,  $z = 57.449$ .

Sunshine Oil should purchase 100% of investments 1, 3, and 4; 20.1% of investment 2; and 28.8% of investment 5. A total NPV of \$57,449,000 will be obtained from these investments.

It is often impossible to purchase only a fraction of an investment without sacrificing the investment's favorable cash flows. Suppose it costs \$12 million to drill an oil well just deep enough to locate a \$30-million gusher. If there were a sole investor in this project who invested \$6 million to undertake half of the project, then he or she would lose the entire investment and receive no positive cash flows. Because, in this example, reducing the money invested by 50% reduces the return by more than 50%, this situation would violate the Proportionality Assumption.

In many capital budgeting problems, it is unreasonable to allow the  $x_i$  to be fractions: Each  $x_i$  should be restricted to 0 (not investing at all in investment  $i$ ) or 1 (purchasing all of investment  $i$ ). Thus, many capital budgeting problems violate the Divisibility Assumption.

#### 4.5.2. Problems

1. Show that if  $r = 0.02$ , investment 2 has a larger NPV than investment 1.
2. Two investments with varying cash flows (in thousands of dollars) are available, as shown in Table 4. At time 0, \$10,000 is available for investment, and at time 1, \$7,000 is available. Assuming that  $r = 0.10$ , set up an LP whose solution maximizes the NPV obtained from these investments. Graphically find the optimal solution to the LP.

**TABLE 4**

| Investment | Cash Flow (in \$ Thousands) at Time |    |   |   |
|------------|-------------------------------------|----|---|---|
|            | 0                                   | 1  | 2 | 3 |
| 1          | -6                                  | -5 | 7 | 9 |
| 2          | -8                                  | -3 | 9 | 7 |

(Assume that any fraction of an investment may be purchased.)

3. A company has nine projects under consideration. The NPV added by each project and the capital required by each project during the next two years is given in Table 5. All figures are in millions. For example, Project 1 will add \$14 million in NPV and require expenditures of \$12 million during year 1 and \$3 million during year 2. Fifty million is available for projects during year 1 and \$20 million is available during year 2. Assuming we may undertake a fraction of each project, how can we maximize NPV?

TABLE 5

|                | Project |    |    |    |    |    |    |    |    |
|----------------|---------|----|----|----|----|----|----|----|----|
|                | 1       | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| Year 1 Outflow | 12      | 54 | 6  | 6  | 30 | 6  | 48 | 36 | 18 |
| Year 2 Outflow | 3       | 7  | 6  | 2  | 35 | 6  | 4  | 3  | 3  |
| NPV            | 14      | 17 | 17 | 15 | 40 | 12 | 14 | 10 | 12 |

## 4.6. Short-Term Financial Planning

LP models can often be used to aid a firm in short- or long-term financial planning (also see Section 3.11). Here we consider a simple example that illustrates how linear programming can be used to aid a corporation's short-term financial planning.

### EXAMPLE 9. Short-Term Financial Planning

Circuit Town is a small electronics company that manufactures tape recorders and radios. The per-unit labor costs, raw material costs, and selling price of each product are given in Table 6. On December 1, 2002, Circuit Town has available raw material that is sufficient to manufacture 100 tape recorders and 100 radios. On the same date, the company's balance sheet is as shown in Table 7, and Circuit Town's asset-liability ratio (called the current ratio) is  $20,000/10,000 = 2$ .

Circuit Town must determine how many tape recorders and radios should be produced during December. Demand is large enough to ensure that all goods produced will be sold. All sales are on credit, however, and payment for goods produced in December will not be received until February 1, 2003. During December, Circuit Town will collect \$2,000 in accounts receivable, and Circuit Town must pay off \$1,000 of the outstanding loan and a monthly rent of \$1,000. On January 1, 2003, Circuit Town will receive a shipment of raw material worth \$2,000, which will be paid for on February 1, 2003. Circuit Town's management has decided that the cash balance on January 1, 2003, must be at least \$4,000. Also, Circuit Town's bank requires that the current ratio at the beginning of January be at least 2. To maximize the contribution to profit from December production, (revenues to be received) - (variable production costs), what should Circuit Town produce during December?

**TABLE 6.** Cost Information for Circuit Town

|                   | Tape Recorder | Radio |
|-------------------|---------------|-------|
| Selling price     | \$100         | \$90  |
| Labor cost        | \$ 50         | \$35  |
| Raw material cost | \$ 30         | \$40  |

**TABLE 7.** Balance Sheet for Circuit Town

|                        | Assets   | Liabilities |
|------------------------|----------|-------------|
| Cash                   | \$10,000 |             |
| Accounts receivable§   | \$ 3,000 |             |
| Inventory outstanding¶ | \$ 7,000 |             |
| Bank loan              |          | \$10,000    |

§Accounts receivable is money owed to Circuit Town by customers who have previously purchased Circuit Town products.

¶Value of December 1, 2002, inventory =  $30(100) + 40(100) = \$7,000$ .

**Solution.** Circuit Town must determine how many tape recorders and radios should be produced during December. Thus, we define

$x_1$  = number of tape recorders produced during December

$x_2$  = number of radios produced during December

To express Circuit Town's objective function, note that

$$\frac{\text{Contribution to profit}}{\text{Tape recorder}} = 100 - 50 - 30 = \$20$$

$$\frac{\text{Contribution to profit}}{\text{Radio}} = 90 - 35 - 40 = \$15$$

As in the Gepetto example, this leads to the objective function

$$\max z = 20x_1 + 15x_2 \quad (4.33)$$

Circuit Town faces the following constraints:

**Constraint 1.** Because of limited availability of raw material, at most 100 tape recorders can be produced during December.

**Constraint 2.** Because of limited availability of raw material, at most 100 radios can be produced during December.

**Constraint 3.** Cash on hand on January 1, 2002, must be at least \$4,000.

**Constraint 4.**  $(\text{January 1 assets})/(\text{January 1 liabilities}) > 2$  must hold.

Constraint 1 is described by

$$x_1 \leq 100 \quad (4.34)$$

Constraint 2 is described by

$$x_2 \leq 100 \quad (4.35)$$

To express Constraint 3, note that

$$\begin{aligned} \text{January 1 cash on hand} &= \text{December 1 cash on hand} \\ &+ \text{accounts receivable collected during December} \\ &- \text{portion of loan repaid during December} \\ &- \text{December rent} - \text{December labor costs} \end{aligned}$$

$$\begin{aligned}
&= 10,000 + 2,000 - 1,000 - 1,000 - 50x_1 - 35x_2 \\
&= 10,000 - 50x_1 - 35x_2
\end{aligned}$$

Now Constraint 3 may be written as

$$10,000 - 50x_1 - 35x_2 \geq 4,000 \quad (4.36')$$

Most computer codes require each LP constraint to be expressed in a form in which all variables are on the left-hand side and the constant is on the right-hand side. Thus, for computer solution, we should write (4.36') as

$$50x_1 + 35x_2 \leq 6,000 \quad (4.36)$$

To express Constraint 4, we need to determine Circuit Town's January 1 cash position, accounts receivable, inventory position, and liabilities in terms of  $x_1$  and  $x_2$ . We have already shown that

Then

$$\begin{aligned}
\text{January 1 cash position} &= 10,000 - 50x_1 - 35x_2 \\
\text{January 1 accounts receivable} &= \text{December 1 accounts receivable} \\
&\quad + \text{accounts receivable from December sales} \\
&\quad - \text{accounts receivable collected during December} \\
&= 3,000 + 100x_1 + 90x_2 - 2000 \\
&= 1,000 + 100x_1 + 90x_2
\end{aligned}$$

It now follows that

$$\begin{aligned}
\text{Value of January 1 inventory} &= \text{value of December 1 inventory} \\
&\quad - \text{value of inventory used in December} \\
&\quad + \text{value of inventory received on January 1} \\
&= 7,000 - (30x_1 + 40x_2) + 2,000 \\
&= 9,000 - 30x_1 - 40x_2
\end{aligned}$$

We can now compute the January 1 asset position:

$$\begin{aligned}
\text{January 1 asset position} &= \text{January 1 cash position} \\
&\quad + \text{January 1 accounts receivable} \\
&\quad + \text{January 1 inventory position} \\
&= (10,000 - 50x_1 - 35x_2) \\
&\quad + (1,000 + 100x_1 + 90x_2) + (9,000 - 30x_1 - 40x_2)
\end{aligned}$$

$$= 20,000 + 20x_1 + 15x_2$$

Finally,

$$\begin{aligned}\text{January 1 liabilities} &= \text{December 1 liabilities} - \text{December loan payment} \\ &+ \text{amount due on January 1 inventory shipment} \\ &= 10,000 - 1,000 + 2,000 \\ &= \$11,000\end{aligned}$$

Constraint 4 may now be written as

$$\frac{20,000 + 20x_1 + 15x_2}{11,000} \geq 2$$

Multiplying both sides of this inequality by 11,000 yields

$$20,000 + 20x_1 + 15x_2 > 22,000$$

Putting this in a form appropriate for computer input, we obtain

$$20x_1 + 15x_2 > 2,000 \quad (4.37)$$

Combining (4.33)–(4.37) with the sign restrictions  $x_1 \geq 0$  and  $x_2 \geq 0$  yields the following LP:

$$\max z = 20x_1 + 15x_2$$

s.t.

$$20x_1 + 15x_2 \leq 100 \quad (\text{Tape recorder constraint})$$

$$20x_1 + 15x_2 \leq 100 \quad (\text{Radio constraint})$$

$$50x_1 + 35x_2 \leq 6,000 \quad (\text{Cash position constraint})$$

$$20x_1 + 15x_2 \geq 2,000 \quad (\text{Current ratio constraint})$$

$$x_1, x_2 \geq 0 \quad (\text{Sign restrictions})$$

When solved graphically (or by computer), the following optimal solution is obtained:

$z = 2,500$ ,  $x_1 = 50$ ,  $x_2 = 100$ . Thus, Circuit Town can maximize the contribution of December's production to profits by manufacturing 50 tape recorders and 100 radios. This will contribute  $20(50) + 15(100) = \$2,500$  to profits.

## 4.7. Multiperiod Financial Models

The following example illustrates how linear programming can be used to model multiperiod cash management problems. The key is to determine the relations of cash on hand during different periods.

### EXAMPLE 10. Incompeto Multiperiod Investment

Incompeto Investment Corporation must determine investment strategy for the firm during the next three years. Currently (time 0), \$100,000 is available for investment. Investments A, B, C, D, and E are available. The cash flow associated with investing \$1 in each investment is given in Table 8.

For example, \$1 invested in investment B requires a \$1 cash outflow at time 1 and returns 50¢ at time 2 and \$1 at time 3. To ensure that the company's portfolio is diversified, Incompeto requires that at most \$75,000 be placed in any single investment. In addition to investments A–E, Incompeto can earn interest at 8% per year by keeping uninvested cash in money market funds. Returns from investments may be immediately reinvested. For example, the positive cash flow received from investment C at time 1 may immediately be reinvested in investment B. Incompeto cannot borrow funds, so the cash available for investment at any time is limited to cash on hand. Formulate an LP that will maximize cash on hand at time 3.

TABLE 8

|   |  | Cash Flow (\$) at Time* |       |       |      |
|---|--|-------------------------|-------|-------|------|
|   |  | 0                       | 1     | 2     | 3    |
|   |  | -1                      | +0.50 | +1    | 0    |
| A |  | -1                      | +0.50 | +1    | 0    |
| B |  | 0                       | -1    | +0.50 | +1   |
| C |  | -1                      | +1.2  | 0     | 0    |
| D |  | -1                      | 0     | 0     | +1.9 |
| E |  | 0                       | 0     | -1    | +1.5 |

**Solution.** Incompeto must decide how much money should be placed in each investment (including money market funds). Thus, we define the following decision variables:

$A$  = dollars invested in investment A

$B$  = dollars invested in investment B

$C$  = dollars invested in investment C

$D$  = dollars invested in investment D

$E$  = dollars invested in investment E

$S_t$  = dollars invested in money market funds at time  $t$  ( $t = 0, 1, 2$ )

Incompeto wants to maximize cash on hand at time 3. At time 3, Incompeto's cash on hand will be the sum of all cash inflows at time 3. From the description of investments  $A-E$  and the fact that from time 2 to time 3,  $S_2$  will increase to  $1.08S_2$ ,

$$\text{Time 3 cash on hand} = B + 1.9D + 1.5E + 1.08S_2$$

Thus, Incompeto's objective function is

$$\max z = B + 1.9D + 1.5E + 1.08S_2 \quad (4.8)$$

In multiperiod financial models, the following type of constraint is usually used to relate decision variables from different periods:

$$\begin{aligned} & \text{Cash available at time } t \\ &= \text{cash invested at time } t \\ &+ \text{uninvested cash at time } t \text{ that is carried over to time } t + 1 \end{aligned}$$

If we classify money market funds as investments, we see that

$$\text{Cash available at time } t = \text{cash invested at time } t \quad (4.39)$$

Because investments  $A, C, D$ , and  $S_0$  are available at time 0, and \$100,000 is available at time 0, (63) for time 0 becomes

$$100,000 = A + C + D + S_0 \quad (4.40)$$

At time 1,  $0.5A + 1.2C + 1.08S_0$  is available for investment, and investments  $B$  and  $S_1$  are available. Then for  $t = 1$ , (63) becomes

$$0.5A + 1.2C + 1.08S_0 = B + S_1 \quad (4.41)$$

At time 2,  $A + 0.5B + 1.08S_1$  is available for investment, and investments  $E$  and  $S_2$  are available. Thus, for  $t = 2$ , (63) reduces to

$$A + 0.5B + 1.08S_1 = E + S_2 \quad (4.42)$$

Let's not forget that at most \$75,000 can be placed in any of investments  $A$ – $E$ . To take care of this, we add the constraints

$$A \leq 75,000 \quad (4.43)$$

$$B \leq 75,000 \quad (4.44)$$

$$C \leq 75,000 \quad (4.45)$$

$$D \leq 75,000 \quad (4.46)$$

$$E \leq 75,000 \quad (4.47)$$

Combining (4.38) and (4.40)–(4.47) with the sign restrictions (all variables  $\geq 0$ ) yields the following LP:

$$\max z = B + 1.9D + 1.5E + 1.08S_2$$

s.t.

$$A + C + D + S_0 = 100,000$$

$$0.5A + 1.2C + 1.08S_0 = B + S_1$$

$$A + 0.5B + 1.08S_1 = E + S_2$$

$$A \leq 75,000$$

$$B \leq 75,000$$

$$C \leq 75,000$$

$$D \leq 75,000$$

$$E \leq 75,000$$

$$A, B, C, D, E, S_0, S_1, S_2 \geq 0$$

We find the optimal solution to be  $z = 218,500$ ,  $A = 60,000$ ,  $B = 30,000$ ,  $D = 40,000$ ,  $E = 75,000$ ,  $C = S_0 = S_1 = S_2 = 0$ . Thus, Incompeto should not invest in money market funds. At time 0, Incompeto should invest \$60,000 in  $A$  and \$40,000 in  $D$ . Then, at time 1, the \$30,000 cash inflow from  $A$  should be invested in  $B$ . Finally, at time 2, the \$60,000 cash inflow from  $A$  and the \$15,000 cash inflow from  $B$  should be invested in  $E$ . At time 3, Incompeto's \$100,000 will have grown to \$218,500.

You might wonder how our formulation ensures that Incompeto never invests more money at any time than the firm has available. This is ensured by the fact that each variable  $S_i$  must be nonnegative. For example,  $S_0 \geq 0$  is equivalent to  $100,000 - A - C - D \geq 0$ , which ensures that at most \$100,000 will be invested at time 0.

#### 4.7.1. Problems

1. A consultant to Incompeto claims that Incompeto's cash on hand at time 3 is the sum of the cash inflows from all investments, not just those investments yielding a cash inflow at time 3. Thus, the consultant claims that Incompeto's objective function should be

$$\max z = 1.5A + 1.5B + 1.2C + 1.9D + 1.5E + 1.08S_0 + 1.08S_1 \\ + 1.08S_2$$

Explain why the consultant is incorrect.

2. A small toy store, Toys Inc. projects the monthly cash flows (in thousands of dollars) in Table 9 during the year 2014. A negative cash flow means that cash outflows exceed cash inflows to the business. To pay its bills, Toys Inc. will need to borrow money early in the year. Money can be borrowed in two ways:
  - a. Taking out a long-term one-year loan in January. Interest of 1% is charged each month, and the loan must be paid back at the end of December.
  - b. Each month money can be borrowed from a short- term bank line of credit. Here, a monthly interest rate of 1.5% is charged. All short-term loans must be paid off at the end of December.

**TABLE 9**

| Month    | Cash Flow | Month     | Cash Flow |
|----------|-----------|-----------|-----------|
| January  | -12       | July      | -7        |
| February | -10       | August    | -2        |
| March    | -8        | September | 15        |

|       |     |          |    |
|-------|-----|----------|----|
| April | -10 | October  | 12 |
| May   | -4  | November | -7 |
| June  | 5   | December | 45 |

At the end of each month, excess cash earns 0.4% interest. Formulate an LP whose solution will help Toys Inc. maximize its cash position at the beginning of January, 2014.

3. Consider Problem 5 with the following modification: Each month Toys Inc. can delay payments on some or all of the cash owed for the current month. This is called “stretching payments.” Payments may be stretched for only one month, and a 1% penalty is charged on the amount stretched. Thus, if it stretches payments on \$10,000 cash owed in January, then it must pay  $10,000(1.01) = \$10,100$  in February. With this modification, formulate an LP that would help Toys Inc. maximize its cash on hand at the beginning of January 1, 2014.

## 4.8. Multiperiod Work Scheduling

In Section 3.5, we saw that linear programming could be used to schedule employees in a static environment where demand did not change over time. The following example (a modified version of a problem from Wagner [1975]) shows how LP can be used to schedule employee training when a firm faces demand that changes over time.

### EXAMPLE 11. Multiperiod Work Scheduling

PEAR is a chain of computer service stores. The number of hours of skilled repair time that PEAR requires during the next five months is as follows:

Month 1 (January): 6,000 hours

Month 2 (February): 7,000 hours

Month 3 (March): 8,000 hours

Month 4 (April): 9,500 hours

Month 5 (May): 11,000 hours

At the beginning of January, 50 skilled technicians work for PEAR. Each skilled technician can work up to 160 hours per month. To meet future demands, new technicians must be trained. It takes one month to train a new technician. During the month of training, a trainee must be supervised for 50 hours by an experienced technician. Each experienced technician is paid \$2,000 a month (even if he or she does not work the full 160 hours). During the month of training, a trainee is paid \$1,000 a month. At the end of each month, 5% of PEAR's experienced technicians quit to join Plum Computers. Formulate an LP whose solution will enable PEAR to minimize the labor cost incurred in meeting the service requirements for the next five months.

**Solution.** PEAR must determine the number of technicians who should be trained during month  $t$  ( $t = 1, 2, 3, 4, 5$ ). Thus, we define

$$x_t = \text{number of technicians trained during month } t \quad (t = 1, 2, 3, 4, 5)$$

PEAR wants to minimize total labor cost during the next five months. Note that

Total labor cost = cost of paying trainees + cost of paying experienced technicians

To express the cost of paying experienced technicians, we need to define, for  $t = 1, 2, 3, 4, 5$ ,

$$y_t = \text{number of experienced technicians at the beginning of month } t$$

Then

*Total labor cost*

$$\begin{aligned} &= (1,000x_1 + 1,000x_2 + 1,000x_3 + 1,000x_4 \\ &\quad + 1,000x_5) \\ &\quad + (2,000y_1 + 2,000y_2 + 2,000y_3 + 2,000y_4 \\ &\quad + 2,000y_5) \end{aligned}$$

Thus, PEAR's objective function is

$$\begin{aligned} \min z &= 1,000x_1 + 1,000x_2 + 1,000x_3 + 1,000x_4 + 1,000x_5 \\ &\quad + 2,000y_1 + 2,000y_2 + 2,000y_3 + 2,000y_4 + 2,000y_5 \end{aligned}$$

What constraints does PEAR face? Note that we are given  $y_1 = 50$ , and that for  $t = 1, 2, 3, 4, 5$ , PEAR must ensure that

$$\begin{aligned} & \text{Number of available technician hours during month } t \geq \\ & \text{Number of technician hours required during month } t \end{aligned} \quad (72)$$

Because each trainee requires 50 hours of experienced technician time, and each skilled technician is available for 160 hours per month,

$$\text{Number of available technician hours during month } t = 160y_t - 50x_t$$

Now (72) yields the following five constraints:

$$\begin{aligned} 160y_1 - 50x_1 &> 6,000 && (\text{month 1 constraint}) \\ 160y_2 - 50x_2 &> 7,000 && (\text{month 2 constraint}) \\ 160y_3 - 50x_3 &> 8,000 && (\text{month 3 constraint}) \\ 160y_4 - 50x_4 &> 9,500 && (\text{month 4 constraint}) \\ 160y_5 - 50x_5 &> 11,000 && (\text{month 5 constraint}) \end{aligned}$$

As in the other multiperiod formulations, we need constraints that relate variables from different periods. In the PEAR problem, it is important to realize that the number of skilled technicians available at the beginning of any month is determined by the number of skilled technicians available during the previous month and the number of technicians trained during the previous month:

$$\begin{aligned} & \text{Experienced technicians available at beginning of month } t \\ &= \text{Experienced technicians available at beginning of month } (t-1) \\ &+ \text{technicians trained during month } (t-1) \\ &- \text{experienced technicians who quit during month } (t-1) \end{aligned} \quad (4.48)$$

For example, for February, (73) yields

$$y_2 = y_1 + x_1 - 0.05y_1 \quad \text{or} \quad y_2 = 0.95y_1 + x_1$$

Similarly, for March, (73) yields

$$y_3 = 0.95y_2 + x_2$$

and for April, and for May,

$$y_4 = 0.95y_3 + x_3$$

$$y_5 = 0.95y_4 + x_4$$

Adding the sign restrictions  $x_t > 0$  and  $y_t > 0$  ( $t = 1, 2, 3, 4, 5$ ), we obtain the follow- ing LP:

$$\begin{aligned} \min z = & 1,000x_1 + 1,000x_2 + 1,000x_3 + 1,000x_4 + 1,000x_5 \\ & + 2,000y_1 + 2,000y_2 + 2,000y_3 + 2,000y_4 + 2,000y_5 \end{aligned}$$

s.t.

$$\begin{aligned} 160y_1 - 50x_1 &> 6,000 & y_1 &= 50 \\ 160y_2 - 50x_2 &> 7,000 & 0.95y_1 + x_1 &= y_2 \\ 160y_3 - 50x_3 &> 8,000 & 0.95y_2 + x_2 &= y_3 \\ 160y_4 - 50x_4 &> 9,500 & 0.95y_3 + x_3 &= y_4 \\ 160y_5 - 50x_5 &> 11,000 & 0.95y_4 + x_4 &= y_5 \\ x_t, y_t &> 0, (t = 1, 2, 3, 4, 5) & & \end{aligned}$$

The optimal solution is  $z = 593,777$ ;  $x_1 = 0$ ;  $x_2 = 8.45$ ;  $x_3 = 11.45$ ;  $x_4 = 9.52$ ;  $x_5 = 0$ ;  $y_1 = 50$ ;  $y_2 = 47.5$ ;  $y_3 = 53.58$ ;  $y_4 = 62.34$ ; and  $y_5 = 68.75$ .

In reality, the  $y_t$ 's must be integers, so our solution is difficult to interpret. The problem with our formulation is that assuming that exactly 5% of the employees quit each month can cause the number of employees to change from an integer during one month to a fraction during the next month. We might want to assume that the number of employees quitting each month is the integer closest to 5% of the total workforce, but then we do not have a linear programming problem!

#### 4.8.1. Problems

1. If  $y_1 = 38$ , then what would be the optimal solution to PEAR's problem?
2. An insurance company believes that it will require the following numbers of personal computers during the next six months: January, 9; February, 5; March, 7; April, 9; May, 10; June, 5. Computers can be rented for a period of one, two, or three months at the following unit rates: one-month rate, \$200; two-month rate, \$350; three-month rate, \$450. Formulate an LP that can be used to minimize the cost of renting the required computers. You may assume that if a

machine is rented for a period of time extending beyond June, the cost of the rental should be prorated. For example, if a computer is rented for three months at the beginning of May, then a rental fee of  $\frac{2}{3}(450) = \$300$ , not \$450, should be assessed in the objective function.

3. The IRS has determined that during each of the next 12 months it will need the number of supercomputers given in Table 10. To meet these requirements, the IRS rents supercomputers for a period of one, two, or three months. It costs \$100 to rent a supercomputer for one month, \$180 for two months, and \$250 for three months. At the beginning of month 1, the IRS has no supercomputers. Determine the rental plan that meets the next 12 months' requirements at minimum cost. Note: You may assume that fractional rentals are okay, so if your solution says to rent 140.6 computers for one month we can round this up or down (to 141 or 140) without having much effect on the total cost.

TABLE 10

| Month | Computer Requirements |
|-------|-----------------------|
| 1     | 800                   |
| 2     | 1,000                 |
| 3     | 600                   |
| 4     | 500                   |
| 5     | 1,200                 |
| 6     | 400                   |
| 7     | 800                   |
| 8     | 600                   |
| 9     | 400                   |
| 10    | 500                   |
| 11    | 800                   |
| 12    | 600                   |

## 4.9. SUMMARY

### 4.9.1. Linear Programming Definitions

A **linear programming problem (LP)** consists of three parts:

1. A linear function (the **objective function**) of decision variables (say,  $x_1, x_2, \dots, x_n$ ) that is to be maximized or minimized.
2. A set of **constraints** (each of which must be a linear equality or linear inequality) that restrict the values that may be assumed by the decision variables.
3. The **sign restrictions**, which specify for each decision variable  $x_j$  either (1) variable  $x_j$  must be nonnegative— $x_j > 0$ ; or (2) variable  $x_j$  may be positive, zero, or negative— $x_j$  is **unrestricted in sign (urs)**.

The coefficient of a variable in the objective function is the variable's **objective function coefficient**. The coefficient of a variable in a constraint is a **technological coefficient**. The right-hand side of each constraint is called a **right-hand side (rhs)**.

A point is simply a specification of the values of each decision variable. The **feasible region** of an LP consists of all points satisfying the LP's constraints and sign restrictions. Any point in the feasible region that has the largest z-value of all points in the feasible region (for a max problem) is an **optimal solution** to the LP. An LP may have no optimal solution, one optimal solution, or an infinite number of optimal solutions.

A constraint in an LP is **binding** if the left-hand side and the right-hand side are equal when the values of the variables in the optimal solution are substituted into the constraint.

### 4.9.2. Graphical Solution of Linear Programming Problems

The feasible region for any LP is a **convex set**. If an LP has an optimal solution, there is an extreme (or corner) point of the feasible region that is an optimal solution to the LP.

We may graphically solve an LP (max problem) with two decision variables as follows:

**Step 1.** Graph the feasible region.

**Step 2.** Draw an isoprofit line.

**Step 3.** Move parallel to the isoprofit line in the direction of increasing  $z$ . The last point in the feasible region that contacts an isoprofit line is an optimal solution to the LP.

#### 4.9.3. LP Solutions: Four Cases

When an LP is solved, one of the following four cases will occur:

**Case 1.** The LP has a unique solution.

**Case 2.** The LP has more than one (actually an infinite number of ) optimal solutions. This is the case of alternative optimal solutions. Graphically, we recognize this case when the isoprofit line last hits an entire line segment before leaving the feasible region.

**Case 3.** The LP is infeasible (it has no feasible solution). This means that the feasible region contains no points.

**Case 4.** The LP is unbounded. This means (in a max problem) that there are points in the feasible region with arbitrarily large  $z$ -values. Graphically, we recognize this case by the fact that when we move parallel to an isoprofit line in the direction of increasing  $z$ , we never lose contact with the LP's feasible region.

#### 4.9.4. Formulating LPs

The most important step in formulating most LPs is to determine the decision variables correctly.

In any constraint, the terms must have the same units. For example, one term cannot have the units “pounds of raw material” while another term has the units “ounces of raw material.”

#### 4.9.5. Review Problems

1. Golden Breweries produces beer and ale. Beer sells for \$5 per barrel,

and ale sells for \$2 per barrel. Producing a barrel of beer requires 5 lb of corn and 2 lb of hops. Producing a barrel of ale requires 2 lb of corn and 1 lb of hops. Sixty pounds of corn and 25 lb of hops are available. Formulate an LP that can be used to maximize revenue. Solve the LP graphically.

2. I now have \$100. The following investments are available during the next three years:

- Investment A.Every dollar invested now yields \$0.10 a year from now and \$1.30 three years from now.
- Investment B.Every dollar invested now yields \$0.20 a year from now and \$1.10 two years from now.
- Investment C.Every dollar invested a year from now yields \$1.50 three years from now.

During each year, uninvested cash can be placed in money market funds, which yield 6% interest per year. At most \$50 may be placed in each of investments A, B, and C. Formulate an LP to maximize my cash on hand three years from now.

3. Incompeto has the following investments available:

- Investment AFor each dollar invested at time 0, we receive \$0.10 at time 1 and \$1.30 at time 2. (Time 0 = now; time 1 = one year from now; and so on.)
- Investment B.For each dollar invested at time 1, we receive \$1.60 at time 2.
- Investment C.For each dollar invested at time 2, we receive \$1.20 at time 3.

At any time, leftover cash may be invested in T-bills, which pay 10% per year. At time 0, we have \$100. At most, \$50 can be invested in each of investments A, B, and C. Formulate an LP that can be used to maximize Incompeto's cash on hand at time 3.

4. CarMin has a \$150,000 advertising budget. To increase automobile sales, the firm is considering advertising in newspapers and on television. The more CarMin uses a particular medium, the less effective is each additional ad. Table 11 shows the number of new

customers reached by each ad. Each newspaper ad costs \$1,000, and each television ad costs \$10,000. At most, 30 newspaper ads and 15 television ads can be placed. How can CarMin maximize the number of new customers created by advertising?

**TABLE 11**

|            | Number of Ads | New Customers |
|------------|---------------|---------------|
| Newspaper  | 1–10          | 900           |
| 11–20      | 600           |               |
| 21–30      | 300           |               |
| Television | 1–5           | 10,000        |
| 6–10       | 5,000         |               |
| 11–15      | 2,000         |               |

5. For a telephone survey, a marketing research group needs to contact at least 150 wives, 120 husbands, 100 single adult males, and 110 single adult females. It costs \$2 to make a daytime call and (because of higher labor costs) \$5 to make an evening call. Table 12 lists the results. Because of limited staff, at most half of all phone calls can be evening calls. Formulate an LP to minimize the cost of completing the survey.

**TABLE 12**

| Person Responding | Percent of Daytime Calls | Percent of Evening Calls |
|-------------------|--------------------------|--------------------------|
| Wife              | 30                       | 30                       |
| Husband           | 10                       | 30                       |
| Single male       | 10                       | 15                       |
| Single female     | 10                       | 20                       |
| None              | 40                       | 5                        |

Currently we own 100 shares each of stocks 1 through

6. The original price we paid for these stocks, today's price, and the expected price in one year for each stock is shown in Table 13. We need money today and are going to sell some of our stocks. The tax rate on capital gains is 30%. If we sell 50 shares of stock 1, then we must pay tax of  $0.3 \cdot 50(30 - 20) = \$150$ . We must also pay transaction costs of 1% on each transaction. Thus, our sale of 50 shares of stock 1 would incur transaction costs of  $0.01 \cdot 50 \cdot 30 = \$15$ . After taxes and transaction costs, we must be left with \$30,000 from our stock sales. Our goal is to maximize the expected (before-tax) value in one year of our remaining stock. What stocks should we sell? Assume it is all right to sell a fractional share of stock.

**TABLE 13**

| Stock | Shares<br>Owned | Price (\$) |         |             |
|-------|-----------------|------------|---------|-------------|
|       |                 | Purchase   | Current | In One Year |
| 1     | 100             | 20         | 30      | 36          |
| 2     | 100             | 25         | 34      | 39          |
| 3     | 100             | 30         | 43      | 42          |
| 4     | 100             | 35         | 47      | 45          |
| 5     | 100             | 40         | 49      | 51          |
| 6     | 100             | 45         | 53      | 55          |
| 7     | 100             | 50         | 60      | 63          |
| 8     | 100             | 55         | 62      | 64          |
| 9     | 100             | 60         | 64      | 66          |
| 10    | 100             | 65         | 66      | 70          |

Tax rate (%) 0.3

Transaction 0.01  
cost (%)

7. The Gotham City Police Department employs 30 police officers. Each officer works 5 days per week. The crime rate fluctuates with the day of the week, so the number of police officers required each day

- depends on which day of the week it is: Saturday, 28; Sunday, 18; Monday, 18; Tuesday, 24; Wednesday, 25; Thursday, 16; Friday, 21. The police department wants to schedule police officers to minimize the number whose days off are not consecutive. Formulate an LP that will accomplish this goal. (Hint: Have a constraint for each day of the week that ensures that the proper number of officers are not working on the given day.)
8. Alexis Peabody makes her living buying and selling corn. On January 1, she has 50 tons of corn and \$1,000. On the first day of each month Alexis can buy corn at the following prices per ton: January, \$300; February, \$350; March, \$400; April, \$500. On the last day of each month, Alexis can sell corn at the following prices per ton: January, \$250; February, \$400; March, \$350; April, \$550. Alexis stores her corn in a warehouse that can hold at most 100 tons of corn. She must be able to pay cash for all corn at the time of purchase. Use linear programming to determine how Alexis can maximize her cash on hand at the end of April.
  9. At the beginning of month 1, Incompeto has \$400 in cash. At the beginning of months 1, 2, 3, and 4, Incompeto receives certain revenues, after which it pays bills (see Table 14). Any money left over may be invested for one month at the interest rate of 0.1% per month; for two months at 0.5% per month; for three months at 1% per month; or for four months at 2% per month. Use linear programming to determine an investment strategy that maximizes cash on hand at the beginning of month 5.

**TABLE 14**

| Month | Revenues (\$) | Bills (\$) |
|-------|---------------|------------|
| 1     | 400           | 600        |
| 2     | 800           | 500        |
| 3     | 300           | 500        |
| 4     | 300           | 250        |

10. City 1 produces 500 tons of waste per day, and city 2 produces 400 tons of waste per day. Waste must be incinerated at incinerator 1 or 2, and each incinerator can process up to 500 tons of waste per day.

The cost to incinerate waste is \$40/ton at incinerator 1 and \$30/ton at 2.

Incineration reduces each ton of waste to 0.2 tons of debris, which must be dumped at one of two landfills. Each landfill can receive at most 200 tons of debris per day. It costs \$3 per mile to transport a ton of material (either debris or waste). Distances (in miles) between locations are shown in Table 15. Formulate an LP that can be used to minimize the total cost of disposing of the waste of both cities.

**Table 15**

| City        | Incinerator |    |
|-------------|-------------|----|
|             | 1           | 2  |
| 1           | 30          | 5  |
| 2           | 36          | 42 |
| Landfill    |             |    |
| Incinerator | 1           | 2  |
| 1           | 5           | 8  |
| 2           | 9           | 6  |

11. Fred Turner is the international funds manager for Countrybank. Each day Fred's job is to determine how the bank's current holdings of dollars, pounds, marks, and yen should be adjusted to meet the day's currency needs. Today the exchange rates between the various currencies are given in Table 16. For example, one dollar can be converted to 0.58928 pounds, or one pound can be converted to 1.697 dollars.

**Table 16**

| From    | To       |         |        |        |
|---------|----------|---------|--------|--------|
|         | Dollars  | Pounds  | Euro   | Yen    |
| Dollars | 1        | 0.58928 | 1.743  | 138.3  |
| Pounds  | 1.697    | 1       | 2.9579 | 243.7  |
| Euro    | 0.57372  | 0.33808 | 1      | 79.346 |
| Yen     | 0.007233 | 0.00426 | 0.0126 | 1      |

At the beginning of the day, Countrybank has the currency holdings given in Table 17.

At the end of the day, Countrybank must have at least the amounts of each currency given in Table 18.

Fred's goal is to each day transfer funds in a way that makes currency holdings satisfy the previously listed mini mums, and maximizes the dollar value of the currency holdings at the end of the day.

To figure out the dollar value of, say, one pound, average the two conversion rates. Thus, one pound is worth approximately

$$\frac{1.697 + \left(\frac{1}{0.58928}\right)}{2} = 1.696993 \text{ dollars}$$

TABLE 17

| Currency | (in Billions) |
|----------|---------------|
| Dollars  | 8             |
| Pounds   | 1             |
| Marks    | 8             |
| Yen      | 0             |

TABLE 18

| Currency | (in Billions) |
|----------|---------------|
| Dollars  | 6             |
| Pounds   | 3             |
| Marks    | 1             |
| Yen      | 10            |

## 4.10. Linear programming using *SCILAB*

The following LPs use built-in LP solving capability of *Octave* and *SCILAB* with the **quapro** add-in module. It is best to write the programs (or copy and paste them) in a text editor, like Notepad ++, and execute them from within *SCILAB*.

For *SCILAB*, you need to start *SCILAB* and at the command prompt, install quapro with the command `atomsInstall("quapro");` and the restart *SCILAB*. We will use the `linpro` function to solve LPs.

### 4.10.1. Linear programming *SCILAB* using quapro

As we have seen, linear programming is an optimization method applied to the solution of problems in which both the objective function and the constraint function are linear function of the design variables. The problem may be expressed as

$$\text{minimize } f(x) = c^T x$$

subject to

$$Ax = b,$$

$$Gx \leq h,$$

$$x_L \leq x \leq x_U$$

where  $x$  a vector containing the  $n$  design variables, i.e.,  $x = [x_1, x_2, \dots, x_n]$ ,  $c$  is the constant column vector containing the  $n$  coefficients of the objective function,  $A$  is an  $m \times n$  matrix,  $b$  is a column vector with  $m$  components,  $G$  is an  $p \times q$  matrix,  $h$  is a column vector with  $p$  components and  $x_L$  and  $x_U$  are vectors indicating constraints on the values of the design variables.

#### EXAMPLE 12

As an example, consider the problem of minimizing the objective function

$$3x_1 + 5x_2 - 2x_3$$

$$x_1 + 3x_2 = 5$$

$$x_1 + x_2 - x_3 = 2$$

$$x_1, x_2, x_3 \geq 0$$

$$x_1 \leq 5, x_2 \leq 10, x_3 \leq 3$$

*SCILAB* provides function *linpro* for the solution of linear programs. The function can be called using any of the following forms:

$$[xopt, logr, fopt] = \text{linpro}(c, C, d[, x0])$$

$$[xopt, logr, fopt] = \text{linpro}(c, C, d, xL, xU, [, x0])$$

$$[xopt, logr, fopt] = \text{linpro}(c, C, d, xL, xU, m, [, x0])$$

$$[xopt, logr, fopt] = \text{linpro}(c, C, d, xL, xU, m, x0[, imp])$$

The function returns values *xopt*, *logr*, and *fopt*, where *xopt* is the value of the design variables vector that minimizes the objective function,  $f(x) = c' * x$ , *logr* is a vector of Lagrange multipliers (see below), and *fopt* is the optimal value of the function, i.e.,  $fopt = f(xopt)$ . The arguments of the function call are:

- the vector of  $n$  coefficients of the objective function, *c*,
- a matrix *C* of  $(m + p)$  rows and  $n$  columns containing the rows of matrices *A* and *G* in that order,
- a column vector *d* with  $(m + p)$  rows containing the elements of vectors *b* and *h* in that order,
- column vectors *x<sub>L</sub>* and *x<sub>U</sub>* representing the vectors of lower and upper bounds of the design variables, i.e., *x<sub>L</sub>* and *x<sub>U</sub>*, respectively ,
- *m* is the number of equality constraints (i.e., the number of rows in matrix *A*),
- *x0*, an optional argument for the first three forms of the function call, is the vector containing the  $n$  initial guesses for the solution,
- *imp*, also an optional argument is an integer value that

determines the amount of information provided by function *linpro* ( try values  $imp = 7, 8, \dots$  )

The first form of the call to *linpro*, namely,  $[xopt, lagr, fopt] = linpro(c, C, d, xL, xU, [x0])$ , is used when there are only inequality constraints. The second form,  $[xopt, lagr, fopt] = linpro(c, C, d, xL, xU, [x0])$ , is used when inequality constraints as well as lower and upper boundaries for the design variables. The second form of the function call,  $[xopt, lagr, fopt] = linpro(c, C, d, xL, xU, m, [x0])$ , is used when there are inequality and equality constraints ( $m$  equality constraints), as well as lower and upper boundaries of the design variables. Finally, the last form of the function call,  $[xopt, lagr, fopt] = linpro(c, C, d, xL, xU, m, x0[, imp])$ , requires an initial guess  $x0$  besides equality and inequality constraints and upper and lower bounds of the design variables.

### EXAMPLE 13

To illustrate the application of function *linpro* we will use the problem stated earlier, namely, minimize  $f(x) = c^T x$ , subject to  $Ax = b$ ,  $Gx \leq h$ , and  $x_L \leq x \leq x_U$ , with

$$c = \begin{bmatrix} 3 \\ 5 \\ -2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, A = \begin{bmatrix} 1 & 3 & 0 \\ 1 & 1 & -1 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 2 \end{bmatrix}, G = \begin{bmatrix} 1 & 3 & 0 \\ 1 & 1 & -1 \end{bmatrix}, h = \begin{bmatrix} 5 \\ 2 \end{bmatrix}, x_L = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, x_U = \begin{bmatrix} 5 \\ 10 \\ 3 \end{bmatrix}$$

We will illustrate the use of function *linpro* by changing the number of constraints of the problem. First, we load the constant vectors and matrices shown using SCI LAB:

```
c=[3;5;-2];
A=[1,3,0;1,1,-1];
b=[5;2];
G=[1,3,0;1,1,-1];
h=[5;2];
xL=zeros(3,1);
xU=[5;10;3];
```

#### 4.10.1.1 Applications of function ***linpro***– case 1: inequality constraints and bounds present

The following *SCILAB* commands set up and solve the linear programming problem using only inequality constraints and bounds for the design variables. For this case we take  $C = G$ ,  $d = h$ , and obtain the solution to the linear programming problem:

```
C=G, d=h
[xopt,lagr,fopt]=linpro (c,c,d,xL,xU)
c =
  1.    3.    0.
  1.    1.   - 1.
d =
  5.
  2.

-->[xopt,lagr,fopt]=linpro (c,c,d,xL,xU)
fopt =
  - 6.
lagr =
  - 3.
  - 5.
  2.
  0.
  0.
xopt =
  0.
  0.
  3.
```

The optimal solution is  $x_1 = x_2 = 0$  and  $x_3 = 3$ , corresponding to a function value  $f(x) = -6$ .

#### 4.10.1.2 Applications of function ***linpro*** – case 2: inequality constraints and bounds present

The following *SCILAB* commands set up and solve the linear programming problem using both equality and inequality constraints

and bounds for the design variables. For this case we take  $C = [A; G]$ ,  $d = [b; h]$ , as follows:

```
C=[A;G],d=[b;h]
[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2)
C =
    1.      3.      0.
    1.      1.    - 1.
    1.      3.      0.
    1.      1.    - 1.
d =
    5.
    2.
    5.
    2.

-->[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2)
fopt =
    9.
lagr =
    0.
    0.
    0.
    - 1.
    - 2.
    0.
    0.
xopt =
    0.5
    1.5
    0.
```

Alternative calls to function *linpro* may include an initial guess for the solution as the last argument in the function call. In the first case, a specific value of  $x$  is provided. In the second case, the option ‘*v*’ indicates that a vertex of the feasible region is to be used as an initial guess. In the third case, the option ‘*g*’ indicates that an arbitrary initial value—generated by the function—is to be used. You can check, by using *SCILAB*, that the solution is the same in any of the following function calls:

```

[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2,[0;0;0])
[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2,'v')
[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2,'g')

-->[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2,[0;0;0])
fopt =
9.
lagr =
0.
0.
0.
- 1.
- 2.
0.
0.
xopt =
0.5
1.5
3.172D-16

-->[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2,'v')
fopt =
9.
lagr =
0.
0.
0.
- 1.
- 2.
0.
0.
xopt =
0.5
1.5
0.

-->[xopt,lagr,fopt]=linpro(c,C,d,xL,xU,2,'g')
fopt =
9.
lagr =
0.

```

```

0.
0.
- 1.
- 2.
0.
0.
xopt  =
0.5
1.5
3.172D-16

```

#### 4.10.2. Graphical illustration of linear programming solution

A linear programming involving two design variables ( $x, y$ ) can be used to illustrate the solution of the problem through graphics. The problem to be solved is to minimize  $f(x, y) = \frac{x_1}{4} + \frac{x_2}{3}$ , subject to the constraints  $-5x_1 - x_2 \leq -5$ ,  $-2x_1 + 5x_2 \leq -10$ ,  $x_1 \geq 0$ ,  $x_2 \geq 0$ . The problem can be set up and solved using *SCILAB* as follows:

```

c=[1/4;1/3]
a=[-5,-1;-2,-5]
b=[-5;-10]
xL=[0;0]
xU=[1e10;1e10]
[xopt,lagr,fopt]=linpro(c,a,b,xL,xU,0)
fopt  =
0.7427536
lagr  =
0.
0.
0.0253623
0.0615942
xopt  =
0.6521739
1.7391304

```

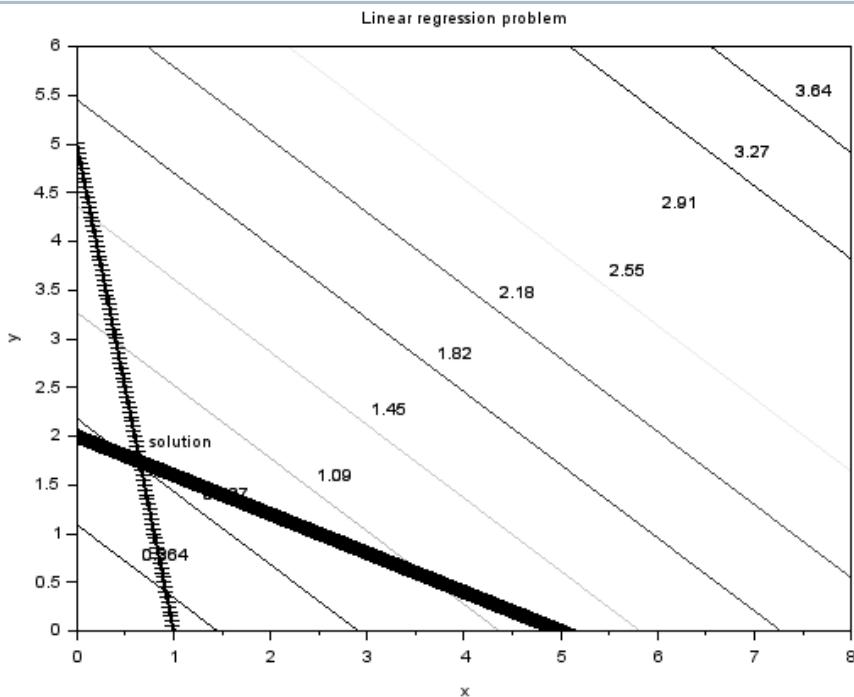
#### EXAMPLE 14

To illustrate the problem graphically we first define the objective function,  $w = f(x, y)$ , and plot a set of contours for it. We also define functions representing the constraints,  $y_1 = f_1(x)$  and  $y_2 = f_2(x)$ , and plot the lines corresponding to these constraints. The solution is the vertex where the two constraint lines intercept as labeled in the plot.

```

deff(' [w]=f(x,y)', 'w=c(1)*x+c(2)*y')
xx=[0:1:8];
yy=[0:1:6];
zz=feval(xx,yy,f);
contour(xx,yy,zz,10)
deff(' [y1]=f1(x)', 'y1=-5*x+5')
deff(' [y2]=f2(x)', 'y2=(10-2*x)/5')
xxx=[0:0.01:8];
yy1=f1(xxx);
yy2=f2(xxx);
contour(xx,yy,zz,10)
plot2d([xxx' xxx'],[yy1' yy2'],[-1,-1],'011',' ',[0 0 8 6])
xstring(xopt(1)+0.05, xopt(2)+0.05,'solution')
xtitle('Linear regression problem','x','y')

```



(made with *SCILAB*)

### 4.10.3. Lagrange multipliers

The elements in the vector of Lagrange multipliers *lagr* provide information about the effect of upper and lower boundaries as well as on any constraints. If vectors of lower and upper bounds, i.e.,  $x_L$  and  $x_U$ , are provided, the vector of Lagrange multipliers, *lagr*, will have  $(n + m + p)$  components. Components 1 to  $n$  are associated with the  $n$  upper and lower bounds of the constraint variables, components  $n + 1$  through  $(n + m)$  are associated with the  $m$  equality constraints, while components  $(n + m + 1)$  through  $(n + m + q)$  are associated with the  $q$  inequality constraints.

If any component of the Lagrange multiplier vector is zero, it means that that particular bound or constraint is not active (i.e., the solution was found without using that particular condition). If an lower bound constraint is active, then the corresponding Lagrange multiplier is negative. On the other hand, if an upper bound constraint is active, the corresponding Lagrange multiplier is positive.

#### EXAMPLE 15

For example, in the first application of function *linpro* used above, we found the following results for the vector of Lagrange multipliers:

```
1agr =  
      - 3.  
      - 5.  
      2.  
      0.  
      0.
```

The first three elements are related to the upper and lower bounds of the variables  $x_1$ ,  $x_2$  and  $x_3$ , respectively. The fact that  $lagr(1) = -3$  and  $lagr(2) = -5$  indicates that the lower bound constraint was used for variables  $x_1$  and  $x_2$ . What this means, in this case, is that  $x_1 = 0$  and  $x_2 = 0$ , i.e., variables  $x_1$  and  $x_2$  took their lower bounds in the optimal solution. The fact that  $lagr(3) = 2$  indicates that the upper bound constraint was used for  $x_3$ , i.e.,  $x_3 = 3.0$ . Finally, the fact that  $lagr(4) = lagr(5) = 0$  indicates that none of the two inequality constrains in the

problem was used. The optimal solution found was  $x_1 = 0$ ,  $x_2 = 0$ , and  $x_3 = 3$ .

### EXAMPLE 16

In the second example of *linpro* presented above the Lagrange multipliers vector was found to be

```
lagr  =
0.
0.
0.
- 1.
- 2.
0.
0.
```

This is interpreted as indicating that none of the three upper or lower bound constraints for the design variables were used ( $lagr(1) = lagr(2) = lagr(3) = 0$ ), neither were any of the two inequality constraints ( $lagr(6) = lagr(7) = 0$ ). The equality constraints, represented by  $lagr(4) = -1$  and  $lagr(5) = -2$ , were both used in the solution.

#### 4.10.4. What are Lagrange multipliers?

Lagrange multipliers are variables introduced in an optimization problem to incorporate the problem constraints into an expanded objective function. For example, given the objective function  $f(x_1, x_2, \dots, x_n)$  subject to the constraints *represented* by  $m (< n)$  equations

$$\varphi_1(x_1, x_2, \dots, x_n) = 0, \varphi_2(x_1, x_2, \dots, x_n) = 0, \dots, \varphi_n(x_1, x_2, \dots, x_n) = 0$$

we can form the expanded objective function

$$\Phi_1(x_1, x_2, \dots, x_n) = f(x_1, x_2, \dots, x_n) + \sum_{j=1}^n \lambda_j \cdot \varphi_j(x_1, x_2, \dots, x_n)$$

The variables  $\lambda_j$  introduced in function  $\Phi$  are known as Lagrange multipliers. The solution to the problem is obtained by solving simultaneously the  $m$  constraint equations and the  $n$  equations resulting from the conditions

$$\frac{\partial \Phi}{\partial x_1} = 0, \frac{\partial \Phi}{\partial x_2} = 0, \dots, \frac{\partial \Phi}{\partial x_n} = 0$$

The resulting problem has  $(n + m)$  unknowns, i.e.,  $x_1, x_2, \dots, x_n$  and  $\lambda_1, \lambda_2, \dots, \lambda_n$  and  $(n + m)$  equations. Thus, the problem has a unique solution that provides the minimum (maximum) value of the objective function  $f(x_1, x_2, \dots, x_n)$ .

#### 4.10.5. SCILAB's karmarkar function

The function karmarkar computes the solution of linear programming problems.

This function has the two following modes.

- If no inequality constraints and no bound is given (i.e. if ( $A == []$  &  $lb == []$  &  $ub == []$ )), the function ensures that the variable is nonnegative.
- If any inequality constraints or any bound is given (i.e. if ( $A <> []$  or  $lb <> []$  or  $ub <> []$ )), the function takes into account for this inequality or bound (and does not ensure that the variable is nonnegative).

If no inequality constraints and no bound is given (i.e. if ( $A == []$  &  $lb == []$  &  $ub == []$ )), solves the linear optimization problem:

$$\text{minimize } c^T = x$$

$$A_{eq}x = b_{eq}$$

$$x_i \geq 0$$

If any inequality constraints or any bound is given (i.e. if ( $A <> []$  |  $lb <> []$  |  $ub <> []$ )), solves the linear optimization problem:

$$\text{minimize } c^T = x$$

$$A_{eq}x = b_{eq}$$

$$Ax_1 \leq b$$

$$x_L \leq x \leq x_U$$

Any optional parameter equal to the empty matrix [] is replaced by its default value.

The exitflag parameter allows to know why the algorithm terminated.

- $\text{exitflag} = 1$  if algorithm converged.
- $\text{exitflag} = 0$  if maximum number of iterations was reached.
- $\text{exitflag} = -1$  if no feasible point was found
- $\text{exitflag} = -2$  if problem is unbounded.
- $\text{exitflag} = -3$  if search direction became zero.
- $\text{exitflag} = -4$  if algorithm stopped on user's request.

### EXAMPLE 17. Equality Constraints

In the following example, we solve a linear optimization problem with 2 linear equality constraints and 3 unknowns. The linear optimization problem is

minimize  $-x_1 - x_2$

$$\begin{aligned}x_1 - x_2 &= 0 \\x_1 + x_2 + x_3 &= 2 \\x_1, x_2, x_3 &\geq 0\end{aligned}$$

The following script solves the problem.

```
Aeq = [
1 -1 0
1 1 1
];
beq = [0;2];
c = [-1;-1;0];
x0 = [0.1;0.1;1.8];
[xopt,fopt,exitflag,iter,yopt]=karmarkar(Aeq,beq,c)
xstar=[1 1 0]'
```

Or using *linpro*:

```
L=[0;0;0]; xU=[];
[xopt,lagr,fopt]=linpro(c,Aeq,beq,xL,xU,2)
fopt =
- 2.
lagr =
0.
0.
- 1.
- 1.743D-32
1.
xopt =
1.
1.
0.
```

### EXAMPLE 18. Inequality Constraints

Consider the following linear program with inequality constraints.

minimize  $-20x_1 - 24x_2$

$$\begin{aligned}x_{31} + 6x_2 &\leq 60 \\4x_1 + 2x_2 &\leq 32 \\x_1, x_2 &\geq 0\end{aligned}$$

```
c = [-20 -24]';
A = [3 6; 4 2];
b = [60 32]';
[xopt,fopt]=karmarkar([],[],c,[],[],[],[],[],A,b)
fopt =
- 271.99804
xopt =
3.9999125
7.9999912
```

Or using *linpro*:

```
[xopt,lagr,fopt]=linpro(c,A,b)
```

```

fopt =
- 272.
lagr =
3.1111111
2.6666667
xopt =
4.
8.

```

### EXAMPLE 19

For this problem there's a total of 6 feasible lease periods that are indicated by  $x_{ij}$ . Where,  $i$  represents the start month and  $j$  the duration of lease.

For example is a lease starting from month 1, lasting 3 months. The space required  $x_{13}$  in months 2 and 4 is less than any of the other months encompassing them and are discarded.

The objective function is based on the duration of the lease.

$$Z = 65x_{11} + 135x_{13} + 190x_{15} + 65x_{31} + 135x_{33} + 65x_{51}$$

The constraints are based on required space each month.

$$\begin{aligned}
x_{11} + x_{13} + x_{15} &\geq 30,000 \\
x_{13} + x_{15} &\geq 20,000 \\
x_{13} + x_{15} + x_{31} + x_{33} &\geq 40,000 \\
x_{15} + x_{33} &\geq 10,000 \\
x_{15} + x_{33} + x_{51} &\geq 50,000
\end{aligned}$$

From the above constraints, we can build a matrix table where 1 represents that chosen period to be in the constraint. For example the first constraint would be equivalent to [1, 1, 1, 0, 0, 0] as only  $x_{11}$ ,  $x_{13}$  and  $x_{15}$  are used.

Using this I formulate the problem into an easily solvable *SCILAB* program.

Matrix Format

```

c = [ 65; 135; 190; 65; 135; 65 ]
a = [
      1, 1, 1, 0, 0, 0;
      0, 1, 1, 0, 0, 0;
      0, 1, 1, 1, 1, 0;
      0, 0, 1, 0, 1, 0;
      0, 0, 1, 0, 1, 1]
b = [ 30000; 20000; 40000; 10000; 50000 ]

```

*SCILAB* program

```

# matrix notation of problem
c = [65; 135; 190; 65; 135; 65]
a = [
      1, 1, 1, 0, 0, 0;
      0, 1, 1, 0, 0, 0;
      0, 1, 1, 1, 1, 0;
      0, 0, 1, 0, 1, 0;
      0, 0, 1, 0, 1, 1]
b = [30000; 20000; 40000; 10000; 50000]

[xopt,fopt]=karmarkar([],[],c,[],[],[],[],[],a,b)
fopt =
      5584174.3
xopt =
      4999.9853
      11246.474
      8753.4033
      9376.7698
      1246.4087
      19999.991

```

From these results, the minimal value of \$5 854 174 is the cost for the company and it will have to lease 30 000 sq feet of space for the whole 5 months and an extra 10 000 sq ft in month 3 and 20 000 sq ft more in month 5. There will also be spare space in month 2 (10 000 sq ft spare) and month 4 (20 000 sq ft spare).

#### EXAMPLE 20. With bounds

Consider the following linear optimization problem. The problem is used in the *SCILAB* port of the *Lipsol* toolbox by Rubio Scola (example0.sce). The original *Lipsol* toolbox was created by Yin Zhang.

$$\text{minimize } 2x_2 + 5x_2 - 2.5x + 3$$

Subject to

$$\begin{aligned}x_1 + s4x_3 &\leq 5 \\E2x_1 - x_2 - x_3 &\geq 0 \\-2 < +x_1 &\leq 2 \\1 \leq x_2 &\\0 \leq x_3 &\leq 3\end{aligned}$$

where  $S4 = \frac{\sin(\frac{\pi}{4})}{4}$  and  $E2 = \exp(2)$ .

```
c = [ 2; 5; -2.5];
S4 = sin(%pi/4)/4;
E2 = exp(2);
A =
[ 1 0 S4
E2 -1 -1
];
b = [ 5; 0];
lb = [ -2; 1 ; 0 ];
ub = [ 2; %inf; 3 ];
xstar = [-2;1;3];
[xopt,fopt,exitflag,iter,yopt]=karmarkar([],[],c,[],[],[],[],[],A,b,lb,ub)
```

The previous script produces the following output.

```
[xopt,fopt,exitflag,iter,yopt]=karmarkar([],[],c,[],[],[],[],[],A,b,lb,ub)
yopt =
ineqlin: [2x1 constant]
eqlin: [0x0 constant]
lower: [3x1 constant]
upper: [3x1 constant]
iter =
76.
exitflag =
1.
fopt =
- 6.4999482
xopt =
- 1.9999914
1.0000035
2.9999931
```

# 5. What is the Simplex Algorithm?

In Chapter 3, we saw how to solve two-variable linear programming problems graphically. Unfortunately, most real-life LPs have many variables, so a method is needed to solve LPs with more than two variables. We devote most of this chapter to a discussion of the simplex algorithm, which is used to solve even very large LPs. In many industrial applications, the simplex algorithm is used to solve LPs with thousands of constraints and variables.

In this chapter, we explain how the simplex algorithm can be used to find optimal solutions to LPs. We also detail how two state-of-the-art computer packages (*SCILAB* and *Excel*) can be used to solve LPs. Briefly, we also discuss Karmarkar's pioneering approach for solving LPs. We close the chapter with an introduction to goal programming, which enables the decision maker to consider more than one objective function.

## 5.1. How to Convert an LP to Standard Form

We have seen that an LP can have both equality and inequality constraints. It also can have variables that are required to be nonnegative as well as those allowed to be unrestricted in sign (urs). Before the simplex algorithm can be used to solve an LP, the LP must be converted into an equivalent problem in which all constraints are equations and all variables are nonnegative. An LP in this form is said to be in standard form.

To convert an LP into standard form, each inequality constraint must be replaced by an equality constraint. We illustrate this procedure using the following problem.

### EXAMPLE 1. Leather Lashes

Leather Lashes manufactures two types of belts: the deluxe model and the regular model. Each type requires 1 sq yd of leather. A regular belt requires 1 hour of skilled labor, and a deluxe belt requires 2 hours. Each week, 40 sq yd of leather and 60 hours of skilled labor are available. Each regular belt contributes \$3 to profit and each deluxe belt, \$4. If we define

$x_1$  = number of deluxe belts produced weekly

$x_2$  = number of regular belts produced weekly

the appropriate LP is

$$\max z = 4x_1 + 3x_2 \quad (\text{LP 1})$$

$$\text{s.t.} \quad x_1 + x_2 \leq 40 \quad (\text{Leather constraint}) \quad (5.1)$$

$$2x_1 + x_2 \leq 60 \quad (\text{Labor constraint}) \quad (5.2)$$

$$x_1, x_2 \geq 0$$

How can we convert (5.1) and (5.2) to equality constraints? We define for each  $\leq$  constraint a slack variable  $s_i$  ( $s_i$  = slack variable for  $i$ th constraint), which is the amount of the resource unused in the  $i$ th constraint. Because  $x_1 + x_2$  sq yd of leather are being used, and 40 sq yd are available, we define  $s_1$  by

$$s_1 = 40 - x_1 - x_2 \quad \text{or} \quad x_1 + x_2 + s_1 = 40$$

Similarly, we define  $s_2$  by

$$s_2 = 60 - 2x_1 - x_2 \quad \text{or} \quad 2x_1 + x_2 + s_2 = 60$$

Observe that a point  $(x_1, x_2)$  satisfies the  $i$ th constraint if and only if  $s_i \geq 0$ . For example,

$x_1 = 15, x_2 = 20$  satisfies (1) because  $s_1 = 40 - 15 - 20 = 5 \geq 0$ .

Intuitively, (5.1) is satisfied by the point  $(15, 20)$ , because  $s_1 = 5$  sq yd of leather are unused. Similarly,  $(15, 20)$  satisfies (5.2), because  $s_2 = 60 - 2(15) - 20 = 10$  labor hours are unused. Finally, note that the point  $x_1 = x_2 = 25$  fails to satisfy (2), because  $s_2 = 60 - 2(25) - 25 = -15$  indicates that  $(25, 25)$  uses more labor than is available.

In summary, to convert (5.1) to an equality constraint, we replace (5.1) by  $s_1 = 40 - x_1 - x_2$  (or  $x_1 + x_2 + s_1 = 40$ ) and  $s_1 \geq 0$ . To convert (5.2) to an equality constraint, we replace (5.2) by  $s_2 = 60 - 2x_1 - x_2$  (or  $2x_1 + x_2 + s_2 = 60$ ) and  $s_2 \geq 0$ . This converts LP 1 to

$$\max z = 4x_1 + 3x_2$$

s.t.

$$\begin{aligned}x_1 + x_2 + s_1 + s_2 &= 40 \\2x_1 + x_2 + s_1 + s_2 &= 60 \\x_1, x_2, s_1, s_2 &\geq 0\end{aligned}$$

(LP 1 )

Note that LP 1 is in standard form. In summary, if constraint  $i$  of an LP is  $a \geq$  constraint, then we convert it to an equality constraint by adding a slack variable  $s_i$  to the  $i$ th constraint and adding the sign restriction  $s_i \geq 0$ .

To illustrate how  $a \geq$  constraint can be converted to an equality constraint, let's consider the following LP.

$$\min z = 50x_1 + 20x_2 + 30x_3 + 80x_4$$

$$\text{s.t.} \quad (\text{Inequalities 5.3-5.6})$$

$$\begin{aligned}400x_1 + 200x_2 + 150x_3 + 500x_4 &\geq 500 \quad (\text{Calorie constraint}) \\3x_1 + 2x_2 &\geq 6 \quad (\text{Chocolate constraint}) \\2x_1 + 2x_2 + 4x_3 + 4x_4 &\geq 10 \quad (\text{Sugar constraint}) \\2x_1 + 4x_2 + x_3 + 5x_4 &\geq 8 \quad (\text{Fat constraint}) \\x_1, x_2, x_3, x_4 &\geq 0\end{aligned}$$

To convert the  $i$ th  $\geq$  constraint to an equality constraint, we define an excess variable (sometimes called a surplus variable)  $e_i$ . ( $e_i$  will always be the excess variable for the  $i$ th constraint.) We define  $e_i$  to be the amount by which the  $i$ th constraint is oversatisfied. Thus, for the diet problem,

$$e_1 = 400x_1 + 200x_2 + 150x_3 + 500x_4 - 500, \quad \text{or} \quad (5.3')$$

$$400x_1 + 200x_2 + 150x_3 + 500x_4 - e_1 = 500$$

$$e_2 = 3x_1 + 2x_2 - 6, \quad \text{or} \quad 3x_1 + 2x_2 - e_2 = 6 \quad (5.4')$$

$$e_3 = 2x_1 + 2x_2 + 4x_3 + 4x_4 - 10, \quad \text{or}$$

$$2x_1 + 2x_2 + 4x_3 + 4x_4 - e_3 = 10 \quad (5.5')$$

$$e_4 = 2x_1 + 4x_2 + x_3 + 5x_4 - 8, \quad \text{or}$$

$$2x_1 + 4x_2 + x_3 + 5x_4 - e_4 = 8 \quad (5.6')$$

A point  $(x_1, x_2, x_3, x_4)$  satisfies the  $i$ th  $\geq$  constraint if and only if  $e_i$  is nonnegative. For example, from (4),  $e_2 \geq 0$  if and only if  $3x_1 + 2x_2 \geq 6$ . For a numerical example, consider the point  $x_1 = 2, x_3 = 4, x_2 = x_4 = 0$ , which satisfies all four of the diet problem's constraints. For this point,

$$e_1 = 400(2) + 150(4) - 500 = 900 > 0$$

$$e_2 = 3(2) - 6 = 0 > 0$$

$$e_3 = 2(2) + 4(4) - 10 = 10 > 0$$

$$e_4 = 2(2) + 4 - 8 = 0 > 0$$

As another example, consider  $x_1 = x_2 = 1, x_3 = x_4 = 0$ . This point is infeasible; it violates the chocolate, sugar, and fat constraints. The infeasibility of this point is indicated by

$$e_2 = 3(1) + 2(1) - 6 = -1 < 0$$

$$e_3 = 2(1) + 2(1) - 10 = -6 < 0$$

$$e_4 = 2(1) + 4(1) - 8 = -2 < 0$$

Thus, to transform the diet problem into standard form, replace (5.3) by (5.3'); (5.4) by (5.4'); (5.5) by (5.5'); and (5.6) by (5.6'). We must also add the sign restrictions  $e_i > 0$  ( $i = 1, 2, 3, 4$ ). The resulting LP is in standard form and may be written as

$$\min z = 50x_1 + 20x_2 + 30x_3 + 80x_4$$

s.t.

$$400x_1 + 200x_2 + 150x_3 + 500x_4 - e_1 = 500$$

$$3x_1 + 2x_2 + 150x_3 + 500x_4 - e_2 = 6$$

$$2x_1 + 2x_2 + 4x_3 + 4x_4 - e_3 = 10$$

$$2x_1 + 4x_2 + x_3 + 5x_4 - e_4 = 8$$

$$x_i, e_i > 0, (i = 1, 2, 3, 4)$$

In summary, if the  $i$ th constraint of an LP is a  $\geq$  constraint, then it can be converted to an equality constraint by subtracting an excess variable  $e_i$  from the  $i$ th constraint and adding the sign restriction  $e_i > 0$ .

If an LP has both  $\leq$  and  $\geq$  constraints, then simply apply the procedures we have described to the individual constraints. As an example, let's convert the short-term financial planning model of Chapter 3 to standard form. Recall that the original LP was

$$\max z = 20x_1 + 15x_2$$

s.t.

$$\begin{aligned} x_1 + 35x_2 &\leq 100 \\ x_1 + 35x_2 &\leq 100 \\ 50x_1 + 35x_2 &\leq 6,000 \\ 20x_1 + 15x_2 &\geq 2,000 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Following the procedures described previously, we transform this LP into standard form by adding slack variables  $s_1$ ,  $s_2$ , and  $s_3$ , respectively, to the first three constraints and subtracting an excess variable  $e_4$  from the fourth constraint. Then we add the sign restrictions  $s_1 > 0$ ,  $s_2 > 0$ ,  $s_3 > 0$ , and  $e_4 > 0$ . This yields the following LP in standard form:

$$\max z = 20x_1 + 15x_2$$

s.t.

$$\begin{aligned} x_1 + 15x_2 + s_1 &= 100 \\ 50x_1 + 15x_2 + s_2 &= 100 \\ 50x_1 + 35x_2 + s_3 &= 6,000 \\ 20x_1 + 15x_2 - e_4 &= 2,000 \\ x_i > 0, (i = 1, 2); \quad s_i > 0, (i = 1, 2, 3); \quad e_4 > 0 \end{aligned}$$

Of course, we could easily have labeled the excess variable for the fourth constraint  $e_1$  (because it is the first excess variable). We chose to call it  $e_4$  rather than  $e_1$  to indicate that  $e_4$  is the excess variable for the fourth constraint.

### 5.1.1. Problems

- Convert the following LP to standard form:

$$\min z = 3x_1 + x_2$$

s.t.

$$\begin{aligned}x_1 &\geq 3 \\x_1 + x_2 &\leq 4 \\2x_1 - x_2 &= 3 \\x_1, x_2 &\geq 0\end{aligned}$$

2. Convert the following LP to standard form:

$$\max z = x_1 + 4x_2 + x_3$$

s.t.

$$\begin{aligned}2x_1 - x_3 &\leq 5 \\x_1 + x_2 + x_3 &\leq 6 \\x_1 - 2x_2 &= 5 \\x_1, x_2, x_3 &\geq 0\end{aligned}$$

## 5.2. Preview of the Simplex Algorithm

Suppose we have converted an LP with  $m$  constraints into standard form. Assuming that the standard form contains  $n$  variables (labeled for convenience  $x_1, x_2, \dots, x_n$ ), the standard form for such an LP is

$$\max z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

(or min)

s.t.

(5.7)

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

⋮

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$x_i > 0, (i = 1, 2, \dots, n)$$

If we define

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

the constraints for (5.7) may be written as the system of equations  $A\mathbf{x} = \mathbf{b}$ . Before proceeding further with our discussion of the simplex algorithm, we must define the concept of a basic solution to a linear system.

### 5.2.1. Basic and Nonbasic Variables

Consider a system  $A\mathbf{x} = \mathbf{b}$  of  $m$  linear equations in  $n$  variables (assume  $n > m$ ).

**DEFINITION ■** A basic solution to  $A\mathbf{x} = \mathbf{b}$  is obtained by setting  $n - m$  variables equal to 0 and solving for the values of the remaining  $m$  variables. This assumes that setting the  $n - m$  variables equal to 0 yields unique values for the remaining  $m$  variables or, equivalently, the columns for the remaining  $m$  variables are linearly independent. ■

To find a basic solution to  $A\mathbf{x} = \mathbf{b}$ , we choose a set of  $n - m$  variables (the **nonbasic variables**, or **NBV**) and set each of these variables equal to 0. Then we solve for the values of the remaining  $n - (n - m) = m$  variables (the **basic variables**, or **BV**) that satisfy  $A\mathbf{x} = \mathbf{b}$ .

Of course, the different choices of nonbasic variables will lead to different basic solutions. To illustrate, we find all the basic solutions to the following system of two equations in three variables:

$$\begin{aligned} x_1 + x_2 &= 3 \\ -x_2 + x_3 &= -1 \end{aligned} \tag{5.8}$$

We begin by choosing a set of  $3 - 2 = 1$  (3 variables, 2 equations) nonbasic variables. For example, if  $NBV = \{x_3\}$ , then  $BV = \{x_1, x_2\}$ . We obtain the values of the basic variables by setting  $x_3 = 0$  and solving

$$\begin{aligned} x_1 + x_2 &= 3 \\ -x_2 &= -1 \end{aligned}$$

We find that  $x_1 = 2$ ,  $x_2 = 1$ . Thus,  $x_1 = 2$ ,  $x_2 = 1$ ,  $x_3 = 0$  is a basic solution to (8). However, if we choose  $NBV = \{x_1\}$  and  $BV = \{x_2, x_3\}$ ,

we obtain the basic solution  $x_1 = 0, x_2 = 3, x_3 = 2$ . If we choose  $NBV = \{x_2\}$ , we obtain the basic solution  $x_1 = 3, x_2 = 0, x_3 = -1$ . The reader should verify these results.

Some sets of  $m$  variables do not yield a basic solution. For example, consider the following linear system:

$$x_1 + 2x_2 + x_3 = 1$$

$$2x_1 + 4x_2 + x_3 = 3$$

If we choose  $NBV = \{x_3\}$  and  $BV = \{x_1, x_2\}$ , the corresponding basic solution would be obtained by solving

$$x_1 + 2x_2 = 1$$

$$2x_1 + 4x_2 = 3$$

Because this system has no solution, there is no basic solution corresponding to  $BV = \{x_1, x_2\}$ .

### 5.2.2. Feasible Solutions

A certain subset of the basic solutions to the constraints  $Ax = b$  of an LP plays an important role in the theory of linear programming.

**DEFINITION** ■ Any basic solution to (7) in which all variables are nonnegative is a **basic feasible solution** (or **bfs**). ■

Thus, for an LP with the constraints given by (8), the basic solutions  $x_1 = 2, x_2 = 1, x_3 = 0$ , and  $x_1 = 0, x_2 = 3, x_3 = 2$  are basic feasible solutions, but the basic solution  $x_1 = 3, x_2 = 0, x_3 = -1$  fails to be a basic solution (because  $x_3 < 0$ ).

In the rest of this section, we assume that all LPs are in standard form. Recall from Chapter 3 that the feasible region for any LP is a convex set. Let  $S$  be the feasible region for an LP in standard form. Recall that a point  $P$  is an extreme point of  $S$  if all line segments that contain  $P$  and are completely contained in  $S$  have  $P$  as an endpoint. It turns out that the extreme points of an LP's feasible region and the LP's basic feasible solutions are actually one and the same. More formally,

**THEOREM 1**

A point in the feasible region of an LP is an extreme point if and only if it is a basic feasible solution to the LP.

See (Luenberger, 1984) for a proof of Theorem 1.

**EXAMPLE 2.** To illustrate the correspondence between extreme points and basic feasible solutions outlined in Theorem 1, let's look at the Leather Lashes example again. Recall that the LP was

$$\max z = 4x_1 + 3x_2$$

$$x_1 + x_2 \leq 40 \quad (\text{LP 1})$$

$$2x_1 + x_2 \leq 60 \quad (5.1)$$

$$x_1, x_2 \geq 0 \quad (5.2)$$

**Solution.** By adding slack variables  $s_1$  and  $s_2$ , respectively, to (5.1) and (5.2), we obtain LP 1 in standard form:

$$\max z = 4x_1 + 3x_2$$

$$\text{s.t.} \quad (\text{LP1}')$$

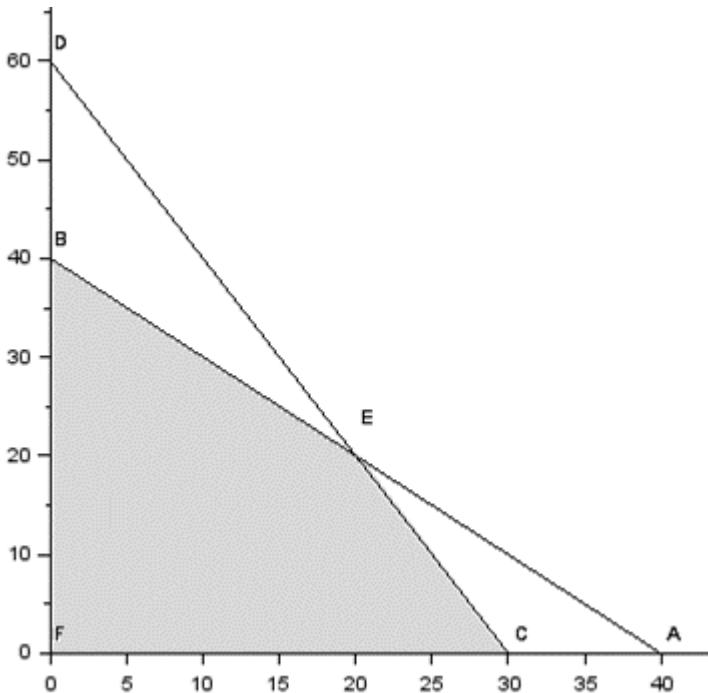
$$x_1 + x_2 + s_1 + s_2 = 40$$

$$2x_1 + x_2 + s_1 + s_2 = 60$$

$$x_1, x_2, s_1, s_2 > 0$$

The feasible region for the Leather Lashes problem is graphed in Figure 1. Both inequalities are satisfied: (5.1) by all points below or on the line  $AB$  ( $x_1 + x_2 = 40$ ), and (5.2) by all points on or below the line  $CD$  ( $2x_1 + x_2 = 60$ ). Thus, the feasible region for LP 1 is the shaded region bounded by the quadrilateral  $BECF$ . The extreme points of the feasible region are  $B = (0,40)$ ,  $C = (30,0)$ ,  $E = (20,20)$ , and  $F = (0,0)$ .

Table 1 shows the correspondence between the basic feasible solutions to LP 1 and the extreme points of the feasible region for LP 1. This example should make it clear that the basic feasible solutions to the standard form of an LP correspond in a natural fashion to the LP's extreme points.



**FIGURE 1.** Feasible Region for Leather Lashes (made with SCILAB)

In the context of the Leather Lashes example, it is easy to show why any bfs is an extreme point. The converse is harder! We now show that for the LL problem, any bfs is an extreme point. Any point in the feasible region for LL may be specified as a four-dimensional column vector with the four elements of the vector denoting  $x_1, x_2, s_1$ , and  $s_2$ , respectively. Consider the bfs  $B$  with  $BV = \{x_2, s_2\}$ . If  $B$  is not an extreme point, then there exists two distinct feasible points  $v_1$  and  $v_2$  and non-negative  $\sigma_1$  and  $\sigma_2$  satisfying  $0 < \sigma_1 < 1$  and  $\sigma_1 + \sigma_2 = 1$ , such that

$$\begin{bmatrix} 0 \\ 40 \\ 0 \\ 20 \end{bmatrix} = \sigma_1 v_1 + \sigma_2 v_2$$

Clearly, both  $v_1$  and  $v_2$  must both have  $x_1 = s_2 = 0$ . But because  $v_1$  and  $v_2$  are both feasible, the values of  $x_2$  and  $s_2$  for both  $v_1$  and  $v_2$  can be determined by solving  $x_2 = 40$  and  $x_2 + s_2 = 60$ . These equations have a unique solution (because columns corresponding to

basic variables  $x_2$  and  $s_2$  are linearly independent). This shows that  $v_1 = v_2$ , so  $B$  is indeed an extreme point.

**TABLE 1.** Correspondence between Basic Feasible Solutions and Corner Points for Leather Lashes

| Basic Variables | Nonbasic Variables | Basic Feasible Solution              | Corresponds to Corner Point |
|-----------------|--------------------|--------------------------------------|-----------------------------|
| $x_1, x_2$      | $s_1, s_2$         | $s_1 = s_2 = 0, x_1 = x_2 = 20$      | $E$                         |
| $x_1, s_1$      | $x_2, s_2$         | $x_2 = s_2 = 0, x_1 = 30, s_1 = 10$  | $C$                         |
| $x_1, s_2$      | $x_2, s_1$         | $x_2 = s_1 = 0, x_1 = 40, s_2 = -20$ | Not bfs, $s_2 < 0$          |
| $x_2, s_1$      | $x_1, s_2$         | $x_1 = s_2 = 0, s_1 = -20, x_2 = 60$ | Not bfs, $s_1 < 0$          |
| $x_2, s_2$      | $x_1, s_1$         | $x_1 = s_1 = 0, x_2 = 40, s_2 = 20$  | $B$                         |
| $s_1, s_2$      | $x_1, x_2$         | $x_1 = x_2 = 0, s_1 = 40, s_2 = 60$  | $F$                         |

We note that more than one set of basic variables may correspond to a given extreme point. If this is the case, then we say the LP is **degenerate**.

We will soon see that if an LP has an optimal solution, then it has a bfs that is optimal. This is important because any LP has only a finite number of bfs's. Thus we can find the optimal solution to an LP by searching only a finite number of points. Because the feasible region for any LP contains an infinite number of points, this helps us a lot!

Before explaining why any LP that has an optimal solution has an optimal bfs, we need to define the concept of a **direction of unboundedness**.

### 5.2.3. Direction of Unboundedness

Consider an LP in standard form with feasible region  $S$  and constraints  $A\mathbf{x} = \mathbf{b}$  and  $\mathbf{x} > 0$ . Assuming that our LP has  $n$  variables,  $0$  represents an  $n$ -dimensional column vector consisting of all 0's. A nonzero vector  $\mathbf{d}$  is a direction of unboundedness if for all  $\mathbf{x} \in S$  and any  $c > 0$ ,  $\mathbf{x} + c\mathbf{d} \in S$ . In short, if we are in the LP's feasible region, then we can move as far as we want in the direction  $\mathbf{d}$  and remain in the feasible region. Figure 2

displays the feasible region for the Phil Short Auto example (Example 2 of Chapter 3). In standard form, the Phil Short example is

$$\min z = 50x_1 + 100x_2$$

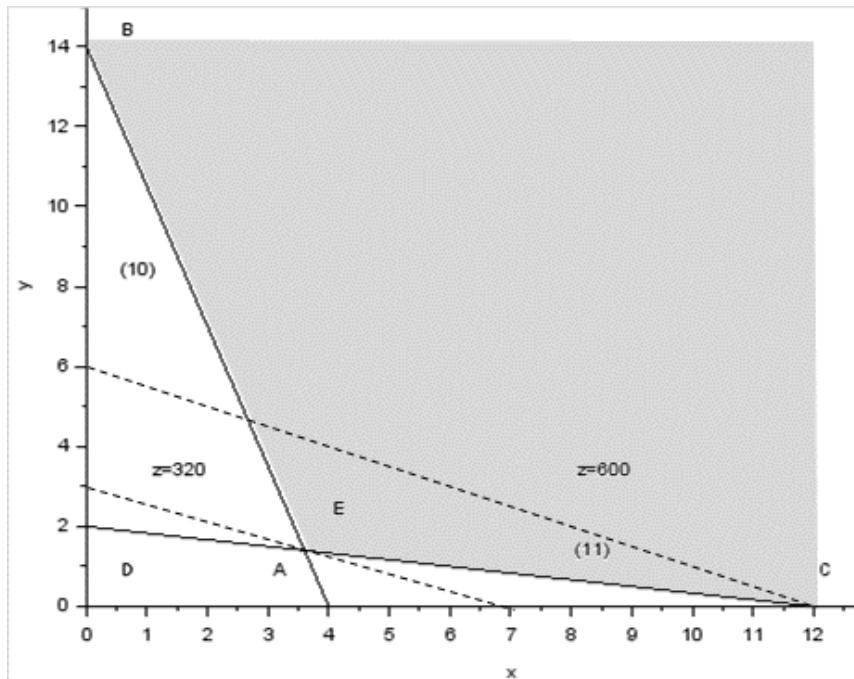
$$7x_1 + 2x_2 - e_1 = 28$$

$$2x_1 + 12x_2 - e_2 = 24$$

$$x_1, x_2, e_1, e_2 > 0$$

Looking at Figure 2 it is clear that if we start at any feasible point and move up and to the right at a 45-degree angle, we will remain in the feasible region. This means that is a direction of unboundedness for this LP. It is easy to show (see Problem 6) that  $\mathbf{d}$  is a direction of unboundedness if and only if  $A\mathbf{d} = 0$  and  $\mathbf{d} > 0$ .

The following Representation Theorem [for a proof, see Nash and Sofer (1996)] is the key insight needed to show why any LP with an optimal solution has an optimal bfs.



**FIGURE 2.** Graphical Solution of Phil Short Problem (made with SCILAB)

## THEOREM 2

Consider an LP in standard form, having bfs  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ . Any point  $\mathbf{x}$  in the LP's feasible region may be written in the form

$$\mathbf{x} = \mathbf{d} + \sum_{i=1}^{i=k} \sigma_i \mathbf{b}_i$$

where  $\mathbf{d}$  is 0 or a direction of unboundedness and  $\sum_{i=1}^{i=k} \sigma_i = 1$  and  $\sigma_i \geq 0$ .

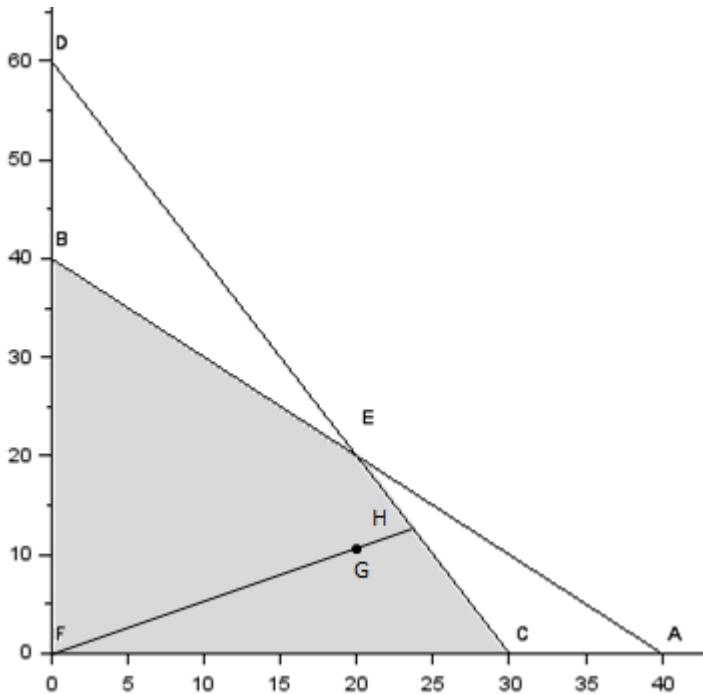
If the LP's feasible region is bounded, then  $\mathbf{d} = 0$ , and we may write  $\mathbf{x} = \mathbf{d} + \sum_{i=1}^{i=k} \sigma_i \mathbf{b}_i$  where the  $\sigma_i$  are nonnegative weights adding to 1. In this case, we see that any feasible  $\mathbf{x}$  may be written as a convex combination of the LP's bfs. We now give two illustrations of Theorem 2.

Consider the Leather Lashes example. The feasible region is bounded. To illustrate Theorem 2, we can write the point  $G = (20,10)$  ( $G$  is not a bfs!) in Figure 3 as a convex combination of the LP's bfs. Note from Figure 3 that point  $G$  may be written as  $\frac{1}{6}F + \frac{5}{6}H$  [here  $H = (24, 12)$ ]. Then note that point  $H$  may be written as  $0.6E + 0.4C$ . Putting these two relationships together, we may write point  $G$  as  $\frac{1}{6}F + \frac{5}{6}(0.6E + 0.4C) = \frac{1}{6}F + \frac{1}{2}E + \frac{1}{3}C$ . This expresses point  $G$  as a convex combination of the LP's extreme points.

To illustrate Theorem 2 for an unbounded LP, let's consider Example 2 of Chapter 3 (the Phil Short example; see Figure 4) and try to express the point  $F = (14, 4)$  in the representation given in Theorem 2. Recall that in standard form the constraints for the Phil Short example are given by

$$7x_1 + 2x_2 - e_1 = 28$$

$$2x_1 + 12x_2 - e_2 = 24$$



**FIGURE 3.** Writing  $(20, 10)$  as a Convex Combination of bfs (made with SCILAB)

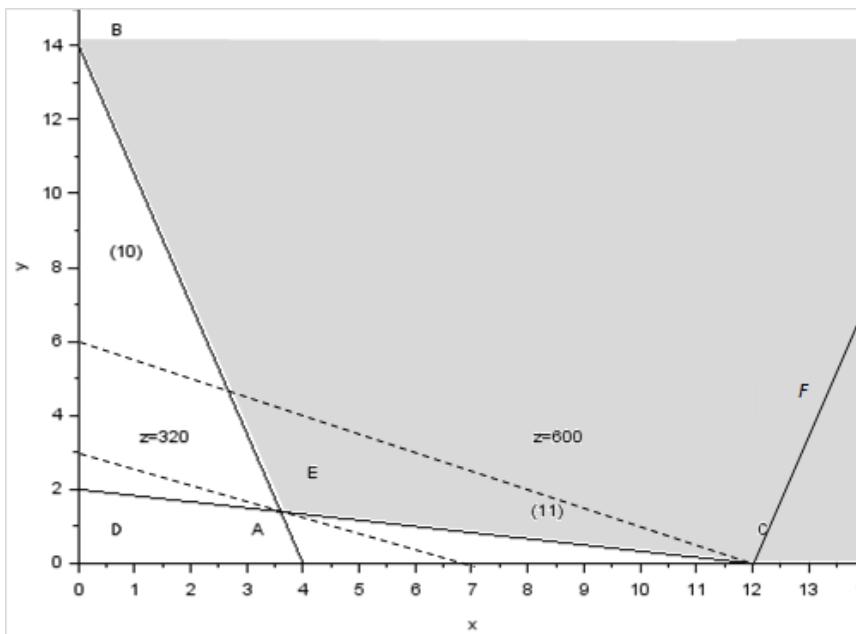
From Figure 4, we see that to move from bfs  $C$  to point  $F$  we need to move up and to the right along a line having slope  $\frac{4-0}{14-12} = 2$ . This line corresponds to the direction of unboundedness

$$\mathbf{d} = \begin{bmatrix} 2 \\ 4 \\ 22 \\ 52 \end{bmatrix}$$

Letting

$$\mathbf{b}_1 = \begin{bmatrix} 12 \\ 0 \\ 56 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} 14 \\ 4 \\ 78 \\ 52 \end{bmatrix}$$

We may write  $\mathbf{x} = \mathbf{d} + \mathbf{b}_1$ , which is the desired representation.



**FIGURE 4.** Expressing  $F = (14, 4)$  Using Theorem 2 (made with SCILAB)

#### 5.2.4. Why Does an LP Have an Optimal bfs?

Consider an LP with objective function  $\max \mathbf{c}\mathbf{x}$  and constraints  $A\mathbf{x} = \mathbf{b}$ . Suppose this LP has an optimal solution. We now sketch a proof of the fact that the LP has an optimal bfs.

#### THEOREM 3

If an LP has an optimal solution, then it has an optimal bfs.

**Proof.** Let  $\mathbf{x}$  be an optimal solution to our LP. Because  $\mathbf{x}$  is feasible, Theorem 2 tells us that we may write  $\mathbf{x} = \mathbf{d} + \sum_{i=1}^{i=k} \sigma_i \mathbf{b}_i$ , where  $\mathbf{d}$  is 0 or a direction of unboundedness and  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$  are the LP's bfs. Also,  $\sum_{i=1}^{i=k} \sigma_i = 1$  and  $\sigma_i \geq 0$ . If  $\mathbf{c}\mathbf{d} > 0$ , then for any  $k > 0$ ,  $k\mathbf{d} + \sum_{i=1}^{i=k} \sigma_i \mathbf{b}_i$  is feasible, and as  $k$  grows larger and larger, the objective function value approaches infinity. This contradicts the fact that the LP has an optimal solution. If  $\mathbf{c}\mathbf{d} < 0$ , then the feasible point  $\sum_{i=1}^{i=k} \sigma_i \mathbf{b}_i$  has a larger objective function value than  $\mathbf{x}$ . This contradicts the optimality of  $\mathbf{x}$ . In short, we have shown that if  $\mathbf{x}$  is optimal, then  $\mathbf{c}\mathbf{d} = 0$ . Now the objective function value for  $\mathbf{x}$  is given by

$$\mathbf{c}\mathbf{x} = \mathbf{c}\mathbf{d} + \sum_{i=1}^{i=k} \sigma_i \mathbf{c}\mathbf{b}_i = \sum_{i=1}^{i=k} \sigma_i \mathbf{c}\mathbf{b}_i$$

Suppose that  $\mathbf{b}_1$  is the bfs with the largest objective function value. Because  $\sum_{i=1}^{i=k} \sigma_i = 1$  and  $\sigma_i \geq 0$

$$\mathbf{c}\mathbf{b}_1 \geq \mathbf{c}\mathbf{x}$$

Because  $\mathbf{x}$  is optimal, this shows that  $\mathbf{b}_1$  is also optimal, and the LP does indeed have an optimal bfs.

### 5.2.5. Problems

1. For the Gepetto problem (Example 1 in Chapter 3), show how the basic feasible solutions to the LP in standard form correspond to the extreme points of the feasible region.
2. For the Phil Short problem (Example 2 in Chapter 3), show how the basic feasible solutions to the LP in standard form correspond to the extreme points of the feasible region.

## 5.3. The Simplex Algorithm

We now describe how the simplex algorithm can be used to solve LPs in which the goal is to maximize the objective function. The solution of minimization problems is discussed in Section 4.4.

The simplex algorithm proceeds as follows:

**Step 1.** Convert the LP to standard form (see Chapter 4).

**Step 2.** Obtain a bfs (if possible) from the standard form.

**Step 3.** Determine whether the current bfs is optimal.

**Step 4.** If the current bfs is not optimal, then determine which nonbasic variable should become a basic variable and which basic variable should become a nonbasic variable to find a new bfs with a better objective function value.

**Step 5.** Use EROs to find the new bfs with the better objective function value. Go back to step 3.

In performing the simplex algorithm, write the objective function

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

in the form

$$z - c_1x_1 - c_2x_2 - \cdots - c_nx_n = 0$$

We call this format the **row 0 version** of the objective function (row 0 for short).

### EXAMPLE 3. Colorado Furniture Company

The Colorado Furniture Company manufactures desks, tables, and chairs. The manufacture of each type of furniture requires lumber and two types of skilled labor: finishing and carpentry. The amount of each resource needed to make each type of furniture is given in Table 4.

Currently, 48 board feet of lumber, 20 finishing hours, and 8 carpentry hours are available. A desk sells for \$60, a table for \$30, and a chair for \$20. Colorado believes that demand for desks and chairs is unlimited, but at most five tables can be sold. Because the available resources have already been purchased, Colorado wants to maximize total revenue.

**TABLE 4.** Resource Requirements for Colorado Furniture

| Resource          | Desk | Table | Chair |
|-------------------|------|-------|-------|
| Lumber (board ft) | 8    | 6.5   | 1.5   |
| Finishing hours   | 4    | 2.5   | 1.5   |
| Carpentry hours   | 2    | 1.5   | 0.5   |

Defining the decision variables as

$x_1$  = number of desks produced

$x_2$  = number of tables produced

$x_3$  = number of chairs produced

it is easy to see that Colorado should solve the following LP:

$$\max z = 60x_1 + 30x_2 + 20x_3$$

s.t.

$$\begin{aligned}
8x_1 + 6x_2 + x_3 &\leq 48 \text{(Lumber constraint)} \\
4x_1 + 2x_2 + 1.5x_3 &\leq 20 \text{(Finishing constraint)} \\
2x_1 + 1.5x_2 + 0.5x_3 &\leq 8 \text{(Carpentry constraint)} \\
x_2 &\leq 5 \text{(Limitation on table demand)} \\
x_1, x_2, x_3 &\geq 0
\end{aligned}$$

### 5.3.1. Convert the LP to Standard Form

We begin the simplex algorithm by converting the constraints of the LP to the standard form discussed in Section 5.1. Then we convert the LP's objective function to the row 0 format. To put the constraints in standard form, we simply add slack variables  $s_1$ ,  $s_2$ ,  $s_3$ , and  $s_4$ , respectively, to the four constraints. We label the constraints row 1, row 2, row 3, and row 4, and add the sign restrictions  $s_i > 0$ , ( $i = 1, 2, 3, 4$ ). Note that the row 0 format for our objective function is

$$z - 60x_1 - 30x_2 - 20x_3 = 0$$

Putting rows 1–4 together with row 0 and the sign restrictions yields the equations and basic variables shown in Table 5. A system of linear equations (such as canonical form 0, shown in Table 5) in which each equation has a variable with a coefficient of 1 in that equation (and a zero coefficient in all other equations) is said to be in canonical form. We will soon see that if the right-hand side of each constraint in a canonical form is nonnegative, a basic feasible solution can be obtained by inspection.

**TABLE 5.** Canonical Form 0

| Row |                                    |             |        | Basic Variable |
|-----|------------------------------------|-------------|--------|----------------|
| 0   | $z - 60x_1 - .30x_2 - .20x_3$      |             | $= 0$  | $z_1 = 0$      |
| 1   | $z - 8x_1 + 6x_2 + x_3 + s_1$      |             | $= 48$ | $s_1 = 48$     |
| 2   | $z - 4x_1 + 2x_2 + 1.5x_3 + s_2$   |             | $= 20$ | $s_2 = 20$     |
| 3   | $z - 2x_1 + 1.5x_2 + 0.5x_3 + s_3$ | $= 8$       |        | $s_3 = 8$      |
| 4   | $x_2$                              | $+ s_4 = 5$ |        | $s_4 = 5$      |

From Section 5.2, we know that the simplex algorithm begins with an initial basic feasible solution and attempts to find better ones. After

obtaining a canonical form, we therefore search for the initial bfs. By inspection, we see that if we set  $x_1 = x_2 = x_3 = 0$ , we can solve for the values of  $s_1, s_2, s_3$ , and  $s_4$  by setting  $s_i$  equal to the right-hand side of row  $i$ .

$$BV = \{s_1, s_2, s_3, s_4\} \quad \text{and} \quad NBV = \{x_1, x_2, x_3\}$$

The basic feasible solution for this set of basic variables is  $s_1 = 48, s_2 = 20, s_3 = 8, s_4 = 5, x_1 = x_2 = x_3 = 0$ . Observe that each basic variable may be associated with the row of the canonical form in which the basic variable has a coefficient of 1. Thus, for canonical form 0,  $s_1$  may be thought of as the basic variable for row 1, as may  $s_2$  for row 2,  $s_3$  for row 3, and  $s_4$  for row 4.

To perform the simplex algorithm, we also need a basic (although not necessarily nonnegative) variable for row 0. Because  $z$  appears in row 0 with a coefficient of 1, and  $z$  does not appear in any other row, we use  $z$  as its basic variable. With this convention, the basic feasible solution for our initial canonical form has

$$BV = \{z, s_1, s_2, s_3, s_4\} \quad \text{and} \quad NBV = \{x_1, x_2, x_3\}$$

For this basic feasible solution,  $z = 0, s_1 = 48, s_2 = 20, s_3 = 8, s_4 = 5, x_1 = x_2 = x_3 = 0$ .

As this example indicates, a slack variable can be used as a basic variable for an equation if the right-hand side of the constraint is nonnegative.

### 5.3.2. Is the Current Basic Feasible Solution Optimal?

Once we have obtained a basic feasible solution, we need to determine whether it is optimal; if the bfs is not optimal, then we try to find a bfs adjacent to the initial bfs with a larger  $z$ -value. To do this, we try to determine whether there is any way that  $z$  can be increased by increasing some nonbasic variable from its current value of zero while holding all other nonbasic variables at their current values of zero. If we solve for  $z$  by rearranging row 0, then we obtain

$$z = 60x_1 + 30x_2 + 20x_3 \tag{5.9}$$

For each nonbasic variable, we can use (5.9) to determine whether increasing a nonbasic variable (and holding all other nonbasic variables

at zero) will increase  $z$ . For example, suppose we increase  $x_1$  by 1 (holding the other nonbasic variables  $x_2$  and  $x_3$  at zero). Then (5.9) tells us that  $z$  will increase by 60. Similarly, if we choose to increase  $x_2$  by 1 (holding  $x_1$  and  $x_3$  at zero), then (5.9) tells us that  $z$  will increase by 30. Finally, if we choose to increase  $x_3$  by 1 (holding  $x_1$  and  $x_2$  at zero), then (5.9) tells us that  $z$  will increase by 20. Thus, increasing any of the nonbasic variables will increase  $z$ . Because a unit increase in  $x_1$  causes the largest rate of increase in  $z$ , we choose to increase  $x_1$  from its current value of zero. If  $x_1$  is to increase from its current value of zero, then it will have to become a basic variable. For this reason, we call  $x_1$  the entering variable. Observe that  $x_1$  has the most negative coefficient in row 0.

### 5.3.3. Determine the Entering Variable

We choose the entering variable (in a max problem) to be the nonbasic variable with the most negative coefficient in row 0 (ties may be broken in an arbitrary fashion). Because each one-unit increase of  $x_1$  increases  $z$  by 60, we would like to make  $x_1$  as large as possible. What limits how large we can make  $x_1$ ? Note that as  $x_1$  increases, the values of the current basic variables ( $s_1, s_2, s_3$ , and  $s_4$ ) will change. This means that increasing  $x_1$  may cause a basic variable to become negative. With this in mind, we look at how increasing  $x_1$  (while holding  $x_2 = x_3 = 0$ ) changes the values of the current set of basic variables. From row 1, we see that  $s_1 = 48 - 8x_1$  (remember that  $x_2 = x_3 = 0$ ). Because the sign restriction  $s_1 \geq 0$  must be satisfied, we can only increase  $x_1$  as long as  $s_1 \geq 0$ , or  $48 - 8x_1 \geq 0$ , or  $x_1 \leq \frac{48}{8} = 6$ . From row 2,  $s_2 = 20 - 4x_1$ . We can only increase  $x_1$  as long as  $s_2 \geq 0$ , so  $x_1$  must satisfy  $20 - 4x_1 > 0$  or  $x_1 \leq \frac{20}{4} = 5$ . From row 3,  $s_3 = 8 - 2x_1$  so  $x_1 \leq \frac{8}{2} = 4$ . Similarly, we see from row 4 that  $s_4 = 5$ . Thus, whatever the value of  $x_1$ ,  $s_4$  will be nonnegative. Summarizing,

$$s_1 > 0 \quad \text{for } x < \frac{48}{8} = 6$$

$$s_2 > 0 \quad \text{for } x < \frac{20}{5} = 5$$

$$s_3 > 0 \quad \text{for } x < \frac{8}{2} = 4$$

$$s_4 > 0 \text{ for all values of } x_1$$

This means that to keep all the basic variables nonnegative, the largest that we can make  $x_1$  is  $\min\left\{\frac{48}{8}, \frac{20}{4}, \frac{8}{2}\right\} = 4$ . If we make  $x_1 > 4$ , then  $s_3$  will become negative, and we will no longer have a basic feasible solution. Notice that each row in which the entering variable had a positive coefficient restricted how large the entering variable could become. Also, for any row in which the entering variable had a positive coefficient, the row's basic variable became negative when the entering variable exceeded

$$\frac{\text{Right-hand side of row}}{\text{Coefficient of entering variable in row}}$$

(5.10)

If the entering variable has a nonpositive coefficient in a row (such as  $x_1$  in row 4), the row's basic variable will remain positive for all values of the entering variable. Using (5.10), we can quickly compute how large  $x_1$  can become before a basic variable becomes negative.

$$\text{Row 1: limit on } x_1 = \frac{48}{8} = 6$$

$$\text{Row 2: limit on } x_2 = \frac{20}{4} = 5$$

$$\text{Row 3: limit on } x_3 = \frac{8}{2} = 4$$

Row 4: limit on  $x_1 = \text{no limit}$  (Because coefficient of  $x_1$  in row 4 is nonpositive)

We can state the following rule for determining how large we can make an entering variable.

#### 5.3.4. The Ratio Test

When entering a variable into the basis, compute the ratio in (5.10) for every constraint in which the entering variable has a positive coefficient.

The constraint with the smallest ratio is called the winner of the ratio test. The smallest ratio is the largest value of the entering variable that will keep all the current basic variables nonnegative. In our example, row 3 was the winner of the ratio test for entering  $x_1$  into the basis.

### 5.3.5. Find a New Basic Feasible Solution: Pivot in the Entering Variable

Returning to our example, we know that the largest we can make  $x_1$  is 4. For  $x_1$  to equal 4, it must become a basic variable. Looking at rows 1–4, we see that if we make  $x_1$  a basic variable in row 1, then  $x_1$  will equal  $\frac{48}{8} = 6$ ; in row 2,  $x_1$  will equal  $\frac{20}{4} = 5$ ; in row 3,  $x_1$  will equal  $\frac{8}{2} = 4$ . Also, because  $x_1$  does not appear in row 4,  $x_1$  cannot be made a basic variable in row 4. Thus, if we want to make  $x_1 = 4$ , then we have to make it a basic variable in row 3. The fact that row 3 was the winner of the ratio test illustrates the following rule.

### 5.3.6. In Which Row Does the Entering Variable Become Basic?

Always make the entering variable a basic variable in a row that wins the ratio test (ties may be broken arbitrarily).

To make  $x_1$  a basic variable in row 3, we use elementary row operations to make  $x_1$  have a coefficient of 1 in row 3 and a coefficient of 0 in all other rows. This procedure is called pivoting on row 3; and row 3 is the pivot row. The final result is that  $x_1$  replaces  $s_3$  as the basic variable for row 3. The term in the pivot row that involves the entering basic variable is called the pivot term. Proceeding as we did when we studied the Gauss–Jordan method in Chapter 2, we make  $x_1$  a basic variable in row 3 by performing the following EROs.

**ERO 1.** Create a coefficient of 1 for  $x_1$  in row 3 by multiplying row 3 by 1. The resulting row (marked with a prime to show it is the first iteration) is

$$x_1 + 0.75x_2 + 0.25x_3 + 0.5s_3 = 4 \quad (\text{row } 3')$$

**ERO 2.** To create a zero coefficient for  $x_1$  in row 0, replace row 0 with  $60(\text{row } 3) + \text{row } 0$ .

$$z + 15x_2 - 5x_3 + 30s_3 = 240 \quad (\text{row } 0')$$

**ERO 3.** To create a zero coefficient for  $x_1$  in row 1, replace row 1 with  $-8(\text{row } 3) + \text{row } 1$ .

$$-x_3 + s_1 - 4s_3 = 16 \quad (\text{row } 1')$$

**ERO 4.** To create a zero coefficient for  $x_1$  in row 2, replace row 2 with  $-4(\text{row } 3) + \text{row } 2$ .

$$-x_2 + 0.5x_3 + s_2 - 2s_3 = 4 \quad (\text{row } 2')$$

Because  $x_1$  does not appear in row 4, we don't need to perform an ero to eliminate  $x_1$  from row 4. Thus, we may write the "new" row 4 (call it row 4 to be consistent with other notation) as

$$x_2 + s_4 = 5 \quad (\text{row } 4')$$

Putting rows 0–4 together, we obtain the canonical form shown in Table 6.

Looking for a basic variable in each row of the current canonical form, we find that

$$BV = \{z, s_1, s_2, x_1, s_4\} \quad \text{and} \quad NBV = \{s_3, x_2, x_3\}$$

**TABLE 6.** Canonical Form 1

| Row |                           |                |             |         |            | Basic Variable |
|-----|---------------------------|----------------|-------------|---------|------------|----------------|
| 0   | $z + 0.15x_2 - 0.25x_3$   |                | $+ 30s_3$   | $= 240$ | $z = 240$  |                |
| 1   |                           | $-x_3 + s_1$   | $- 4s_3$    | $= 16$  | $s_1 = 16$ |                |
| 2   | $-x_2 + 0.5x_3$           | $+ s_2 - 2s_3$ |             | $= 4$   | $s_2 = 4$  |                |
| 3   | $x_1 + 0.75x_2 + 0.25x_3$ |                | $+ 0.5s_3$  | $= 4$   | $x_1 = 4$  |                |
| 4   | $x_2$                     |                | $+ s_4 = 5$ |         | $s_4 = 5$  |                |

Thus, canonical form 1 yields the basic feasible solution  $z = 240$ ,  $s_1 = 16$ ,  $s_2 = 4$ ,  $x_1 = 4$ ,  $s_4 = 5$ ,  $x_2 = x_3 = s_3 = 0$ . We could have predicted that the value of  $z$  in canonical form 1 would be 240 from the fact that

each unit by which  $x_1$  is increased increases  $z$  by 60. Because  $x_1$  was increased by 4 units (from  $x_1 = 0$  to  $x_1 = 4$ ), we would expect that

$$\begin{aligned}\text{Canonical form 1 } z - \text{value} &= \text{initial } z - \text{value} + 4(60) \\ &= 0 + 240 = 240\end{aligned}$$

In obtaining canonical form 1 from the initial canonical form, we have gone from one bfs to a better (larger  $z$ -value) bfs. Note that the initial bfs and the improved bfs are adjacent. This follows because the two basic feasible solutions have  $4 - 1 = 3$  basic variables ( $s_1, s_2$ , and  $s_4$ ) in common (excluding  $z$ , which is a basic variable in every canonical form). Thus, we see that in going from one canonical form to the next, we have proceeded from one bfs to a better adjacent bfs. The procedure used to go from one bfs to a better adjacent bfs is called an **iteration** (or sometimes, a pivot) of the simplex algorithm.

We now try to find a bfs that has a still larger  $z$ -value. We begin by examining canonical form 1 (Table 6) to see if we can increase  $z$  by increasing the value of some nonbasic variable (while holding all other nonbasic variables equal to zero). Rearranging row 0 to solve for  $z$  yields

$$z = 240 - 15x_2 + 5x_3 - 30s_3 \quad (5.11)$$

From (5.11), we see that increasing the nonbasic variable  $x_2$  by 1 (while holding  $x_3 = s_3 = 0$ ) will decrease  $z$  by 15. We don't want to do that! Increasing the nonbasic variable  $s_3$  by 1 (holding  $x_2 = x_3 = 0$ ) will decrease  $z$  by 30. Again, we don't want to do that. On the other hand, increasing  $x_3$  by 1 (holding  $x_2 = s_3 = 0$ ) will increase  $z$  by 5. Thus, we choose to enter  $x_3$  into the basis. Recall that our rule for determining the entering variable is to choose the variable with the most negative coefficient in the current row 0. Because  $x_3$  is the only variable with a negative coefficient in row 0, it should be entered into the basis.

Increasing  $x_3$  by 1 will increase  $z$  by 5, so it is to our advantage to make  $x_3$  as large as possible. We can increase  $x_3$  as long as the current basic variables ( $s_1, s_2, x_1$ , and  $s_4$ ) remain nonnegative. To determine how large  $x_3$  can be, we must solve for the values of the current basic variables in terms of  $x_3$  (holding  $x_2 = s_3 = 0$ ). We obtain

From row 1 :  $s_1 = 16 + x_3$

From row 2 :  $s_2 = 4 - 0.5x_3$

From row 3 :  $x_1 = 4 - 0.25x_3$

From row 4 :  $s_4 = 5$

These equations tell us that  $s_1 \geq 0$  and  $s_4 \geq 0$  will hold for all values of  $x_3$ . From row 2 , we see that  $s_2 \geq 0$  will hold if  $4 - 0.5x_3 \geq 0$ , or  $x_3 \leq \frac{4}{0.5} = 8$ . From row 3 ,  $x_1 \geq 0$  will hold if  $4 - 0.25x_3 \geq 0$ , or  $x_3 < \frac{4}{0.25} = 16$ . This shows that the largest we can make  $x_3$  is  $\min\left\{\frac{4}{0.5}, \frac{4}{0.25}\right\} = 8$ . This fact could also have been discovered by using (5.10) and the ratio test, as follows:

Row 1 : no ratio  $(x_3$  has negative coefficient in row 1)

Row 2 :  $\frac{4}{0.5} = 8$

Row 3 :  $\frac{4}{0.25} = 16$

Row 4 : no ratio  $(x_3$  has a nonpositive coefficient in row 4)

Thus, the smallest ratio occurs in row 2, and row 2 wins the ratio test. This means that we should use EROs to make  $x_3$  a basic variable in row 2.

**ERO 1.** Create a coefficient of 1 for  $x_3$  in row 2 by replacing row 2 with 2(row 2 ):

$$-2x_2 + x_3 + 2s_2 - 4s_3 = 8 \quad (\text{row } 2'')$$

**ERO 2.** Create a coefficient of 0 for  $x_3$  in row 0 by replacing row 0 with 5(row 2)" + row 0 :

$$z + 5x_2 + 10s_2 + 10s_3 = 280 \quad (\text{row } 0'')$$

**ERO 3.** Create a coefficient of 0 for  $x_3$  in row 1 by replacing row 1 with row 2" + row 1 :

$$-2x_2 + s_1 + 2s_2 - 8s_3 = 24 \quad (\text{row } 1'')$$

**ERO 4.** Create a coefficient of 0 for  $x_3$  in row 3 , by replacing row 3 with - 1 (row 2") + 3 :

$$x_1 + 1.25x_2 - 0.5s_2 + 1.5s_3 = 2 \quad (\text{row 3}'')$$

Because  $x_3$  already has a zero coefficient in row 4, we may write

$$x_2 + s_4 = 5 \quad (\text{row 4}''')$$

Combining rows 0"– 4" gives the canonical form shown in Table 7.

**TABLE 7.** Canonical Form 2

| Row |     |                 |                       |           | =     | Basic Variable |
|-----|-----|-----------------|-----------------------|-----------|-------|----------------|
| 0"  | $z$ | $5x_2$          | $+ 10s_2 + 10s_3$     |           | = 280 | $z = 280$      |
| 1"  |     | $- 2x_2$        | $+ s_1 + 2s_2 - 8s_3$ |           | = 24  | $s_1 = 24$     |
| 2"  |     | $- 2x_2 + x_3$  | $+ 2s_2 - 4s_3$       |           | = 8   | $x_3 = 8$      |
| 3"  |     | $x_1 + 1.25x_2$ | $- 0.5s_2 + 1.5s_3$   |           | = 2   | $x_1 = 2$      |
| 4"  |     | $- x_2$         |                       | $+ s_4 =$ | 5     | $s_4 = 5$      |

Looking for a basic variable in each row of canonical form 2, we find

$$BV = \{z, s_1, x_3, x_1, s_4\} \text{ and } NBV = \{s_2, s_3, x_2\}$$

Canonical form 2 yields the following bfs:  $z = 280, s_1 = 24, x_3 = 8, x_1 = 2, s_4 = 5, s_2 = s_3 = x_2 = 0$ . We could have predicted that canonical form 2 would have  $z = 280$  from the fact that each unit of the entering variable  $x_3$  increased  $z$  by 5, and we have increased  $x_3$  by 8 units. Thus,

$$\begin{aligned} \text{Canonical form 2 } z - \text{value} &= \text{canonical form 1 } z - \text{value} + 8(5) \\ &= 240 + 40 = 280 \end{aligned}$$

Because the bfs's for canonical forms 1 and 2 have (excluding  $z$ )  $4 - 1 = 3$  basic variables in common ( $s_1, s_4, x_1$ ), they are adjacent basic feasible solutions.

Now that the second iteration (or pivot) of the simplex algorithm has been completed, we examine canonical form 2 to see if we can find a better bfs. If we rearrange row 0" and solve for  $z$ , we obtain

$$z = 280 - 5x_2 - 10s_2 - 10s_3 \quad (5.12)$$

From (5.12), we see that increasing  $x_2$  by 1 (while holding  $s_2 = s_3 = 0$ ) will decrease  $z$  by 5; increasing  $s_2$  by 1 (holding  $s_3 = x_2 = 0$ ) will decrease  $z$  by 10; increasing  $s_3$  by 1 (holding  $x_2 = s_2 = 0$ ) will decrease  $z$  by 10. Thus, increasing any nonbasic variable will cause  $z$  to decrease. This might lead us to believe that our current bfs from canonical form 2 is an optimal solution. This is indeed correct! To see why, look at (5.12). We know that any feasible solution to the Colorado Furniture problem must have  $x_2 \geq 0$ ,  $s_2 \geq 0$ , and  $s_3 \geq 0$ , and  $-5x_2 \leq 0$ ,  $-10s_2 \leq 0$ , and  $-10s_3 \leq 0$ . Combining these inequalities with (5.12), it is clear that any feasible solution must have  $z = 280 +$  terms that are  $\leq 0$ , and  $z \leq 280$ . Our current bfs from canonical form 2 has  $z = 280$ , so it must be optimal.

The argument that we just used to show that canonical form 2 is optimal revolved around the fact that each of its nonbasic variables had a nonnegative coefficient in row 0". This means that we can determine whether a canonical form's bfs is optimal by applying the following simple rule.

### 5.3.7. Is a Canonical Form Optimal (Max Problem)?

A canonical form is optimal (for a max problem) if each nonbasic variable has a non-negative coefficient in the canonical form's row 0.

#### 5.3.7.1            REMARKS

1. The coefficient of a decision variable in row 0 is often referred to as the variable's **reduced cost**. Thus, in our optimal canonical form, the reduced costs for  $x_1$  and  $x_3$  are 0, and the reduced cost for  $x_2$  is 5. The reduced cost of a nonbasic variable is the amount by which the value of  $z$  will decrease if we increase the value of the nonbasic variable by 1 (while all the other nonbasic variables remain equal to 0). For example, the reduced cost for the variable "tables" ( $x_2$ ) in canonical form 2 is 5. From (12.5), we see that increasing  $x_2$  by 1 will reduce  $z$  by 5. Note that because all basic variables (except  $z$ , of course) must have zero coefficients in row 0, the reduced cost for a basic variable will always be 0. In Chapters 6, we discuss the concept of reduced costs in much greater detail.

These comments are correct only if the values of all the basic

variables remain nonnegative after the nonbasic variable is increased by 1. Increasing  $x_2$  to 1 leaves  $x_1$ ,  $x_3$ , and  $s_1$  all nonnegative, so our comments are valid.

2. From canonical form 2, we see that the optimal solution to the Colorado Furniture problem is to manufacture 2 desks ( $x_1 = 2$ ) and 8 chairs ( $x_3 = 8$ ). Because  $x_2 = 0$ , no tables should be made. Also,  $s_1 = 24$  is reasonable because only  $8 + 8(2) = 24$  board feet of lumber are being used. Thus,  $48 - 24 = 24$  board feet of lumber are not being used. Similarly,  $s_4 = 5$  makes sense because, although up to 5 tables could have been produced, 0 tables are actually being produced. Thus, the slack in constraint 4 is  $5 - 0 = 5$ . Because  $s_2 = s_3 = 0$ , all available finishing and carpentry hours are being utilized, so the finishing and carpentry constraints are binding.
3. We have chosen the entering variable to be the one with the most negative coefficient in row 0, but this may not always lead us quickly to the optimal bfs. Actually, even if we choose the variable with the smallest (in absolute value) negative coefficient, the simplex algorithm will eventually find the LP's optimal solution.
4. Although any variable with a negative row 0 coefficient may be chosen to enter the basis, the pivot row must be chosen by the ratio test. To show this formally, suppose that we have chosen to enter  $x_i$  into the basis, and in the current tableau  $x_i$  is a basic variable in row  $k$ . Then row  $k$  may be written as

$$\bar{a}_{ki}x_i + \dots = \bar{b}_k$$

Consider any other constraint (say, row  $j$ ) in the canonical form. Row  $j$  in the current canonical form may be written as

$$\bar{a}_{ji}x_i + \dots = \bar{b}_j$$

If we pivot on row  $k$ , row  $k$  becomes

$$x_i + \dots = \frac{\bar{b}_k}{\bar{a}_{ki}}$$

The new row  $j$  after the pivot will be obtained by adding  $-\bar{a}_{ji}$  times the last equation to row  $j$  of the current canonical form. This yields a new row  $j$  of

$$0x_i + \dots = \bar{b}_j - \frac{\bar{b}_k \bar{a}_{ji}}{\bar{a}_{ki}}$$

We know that after the pivot, each constraint must have a nonnegative right-hand side. Thus,  $\bar{a}_{ki} \geq 0$  must hold to ensure that row  $k$  has a nonnegative right-hand side after the pivot. Suppose  $\bar{a}_{ji} \geq 0$ . Then, to ensure that row  $j$  will have a nonnegative right-hand side after the pivot, we must have

$$\frac{\bar{b}_j - \bar{b}_k \bar{a}_{ji}}{\bar{a}_{ki}} \geq 0$$

or (because  $\bar{a}_{ji} \geq 0$ )

$$\frac{\bar{b}_j}{\bar{a}_{ji}} = \frac{\bar{b}_k}{\bar{a}_{ki}}$$

Thus, row  $k$  must be a “winner” of the ratio test to ensure that row  $j$  will have a nonnegative right-hand side after the pivot is completed.

If  $\bar{a}_{ji} \leq 0$ , then the right-hand side of row  $j$  will surely be nonnegative after the pivot. This follows because

$$\frac{\bar{b}_k \bar{a}_{ji}}{\bar{a}_{ki}} \geq 0$$

will now hold.

As promised earlier, we have outlined an algorithm that proceeds from one bfs to a better bfs. The algorithm stops when an optimal solution has been found. The convergence of the simplex algorithm is discussed further in this chapter.

### 5.3.8. Summary of the Simplex Algorithm for a Max Problem

**Step 1.** Convert the LP to standard form.

**Step 2.** Find a basic feasible solution. This is easy if all the constraints are  $\leq$  with non-negative right-hand sides. Then the slack variable  $s_i$  may be used as the basic variable for row  $i$ . If no bfs is readily apparent, then use the techniques discussed in earlier in this chapter to find a bfs.

**Step 3.** If all nonbasic variables have nonnegative coefficients in row 0, then the current bfs is optimal. If any variables in row 0 have negative coefficients, then choose the variable with the most negative coefficient in row 0 to enter the basis. We call this variable the entering variable.

**Step 4.** Use EROs to make the entering variable the basic variable in any row that wins the ratio test (ties may be broken arbitrarily). After the EROs have been used to create a new canonical form, return to step 3, using the current canonical form.

When using the simplex algorithm to solve problems, there should never be a constraint with a negative right-hand side (it is okay for row 0 to have a negative right-hand side). A constraint with a negative right-hand side is usually the result of an error in the ratio test or in performing one or more EROs. If one (or more) of the constraints has a negative right-hand side, then there is no longer a bfs, and the rules of the simplex algorithm may not lead to a better bfs.

### 5.3.9. Representing Simplex Tableaus

Rather than writing each variable in every constraint, we often used a shorthand display called a simplex tableau. For example, the canonical form

$$\begin{aligned} z + 3x_1 + x_2 &= 6 \\ 3x_1 + x_2 + s_1 &= 4 \end{aligned}$$

**TABLE 8.** A Simplex Tableau

| $z$ | $x_1$ | $x_2$ | $s_1$ | $s_2$ | rhs | Basic Variable |
|-----|-------|-------|-------|-------|-----|----------------|
| 1   | 3     | 1     | 0     | 0     | 6   | $z_2 = 6$      |
| 0   | 1     | 0     | 1     | 0     | 4   | $s_1 = 4$      |
| 0   | 2     | 1     | 0     | 1     | 3   | $s_2 = 3$      |

### 5.3.10. Problems

1. Use the simplex algorithm to solve the Gepetto problem (Example 1 in Chapter 3).
2. Use the simplex algorithm to solve the following LP:

$$\max z = 2x_1 + 3x_2$$

s.t.

$$x_1 + 2x_2 \leq 6$$

$$2x_1 + x_2 \leq 8$$

$$x_1, x_2 \geq 0$$

3. Use the simplex algorithm to solve the following problem:

$$\max z = 2x_1 - x_2 + x_3$$

s.t.

$$3x_1 + x_2 + x_3 \leq 60$$

$$2x_1 + x_2 + 2x_3 \leq 20$$

$$2x_1 + 2x_2 + x_3 \leq 20$$

$$x_1, x_2, x_3 \geq 0$$

## 5.4. Using the Simplex Algorithm to Solve Minimization Problems

There are two different ways that the simplex algorithm can be used to solve minimization problems. We illustrate these methods by solving the following LP:

$$\min z = 2x_1 - 3x_2$$

s.t.

(LP 2)

$$x_1 + x_2 \leq 4$$

$$x_1 - x_2 \leq 6$$

$$x_1, x_2 \geq 0$$

### 5.4.1. Method 1

The optimal solution to LP 2 is the point  $(x_1, x_2)$  in the feasible region for LP 2 that makes  $z = 2x_1 - 3x_2$  the smallest. Equivalently, we may

say that the optimal solution to LP 2 is the point in the feasible region that makes  $-z = -2x_1 + 3x_2$  the largest. This means that we can find the optimal solution to LP 2 by solving LP 2:

$$\begin{aligned} \max -z &= -2x_1 + 3x_2 \\ \text{s.t.} & \end{aligned} \tag{LP 2}$$

$$\begin{aligned} x_1 + x_2 &\leq 4 \\ x_1 - x_2 &\leq 6 \\ x_1, x_2 &\geq 0 \end{aligned}$$

In solving LP 2, we will use  $-z$  as the basic variable for row 0. After adding slack variables  $s_1$  and  $s_2$  to the two constraints, we obtain the initial tableau in Table 9. Because  $x_2$  is the only variable with a negative coefficient in row 0, we enter  $x_2$  into the basis. The ratio test indicates that  $x_2$  should enter the basis in the first constraint, row 1. The resulting tableau is shown in Table 10. Because each variable in row 0 has a nonnegative coefficient, this is an optimal tableau. Thus, the optimal solution to LP 2 is  $-z = 12$ ,  $x_2 = 4$ ,  $s_2 = 10$ ,  $x_1 = s_1 = 0$ . Then the optimal solution to LP 2 is  $z = -12$ ,  $x_2 = 4$ ,  $s_2 = 10$ ,  $x_1 = s_1 = 0$ . Substituting the values of  $x_1$  and  $x_2$  into LP 2's objective function, we obtain

$$z = 2x_1 - 3x_2 = 2(0) - 3(4) = -12$$

In summary, multiply the objective function for the min problem by -1 and solve the problem as a maximization problem with objective function  $-z$ . The optimal solution to the max problem will give you the optimal solution to the min problem. Remember that (optimal z-value for min problem)= -(optimal objective function value z for max problem).

**TABLE 9.** Initial Tableau for LP 2—Method 1

| $-z$ | $x_1$ | $x_2$ | $s_1$ | $s_2$ | rhs | Basic Variable | Ratio               |
|------|-------|-------|-------|-------|-----|----------------|---------------------|
| 1    | 2     | -3    | 0     | 0     | 0   | $-z = 0$       |                     |
| 0    | 1     | 1     | 1     | 0     | 4   | $s_1 = 4$      | $\frac{4}{1} = 4 *$ |
| 0    | 1     | -1    | 0     | 1     | 6   | $-s_2 = 6$     | None                |

**TABLE 10.** Optimal Tableau for LP 2—Method 1

| $-z$ | $x_1$ | $x_2$ | $s_1$ | $s_2$ | rhs | Basic Variable |
|------|-------|-------|-------|-------|-----|----------------|
| 1    | 5     | 0     | 3     | 0     | 12  | $-z_2 = 12$    |
| 0    | 1     | 1     | 1     | 0     | 14  | $-x_2 = 41$    |
| 0    | 2     | 0     | 1     | 1     | 10  | $-s_2 = 10$    |

### 5.4.2. Method 2

A simple modification of the simplex algorithm can be used to solve min problems directly. Modify Step 3 of the simplex as follows: If all nonbasic variables in row 0 have nonpositive coefficients, then the current bfs is optimal. If any nonbasic variable in row 0 has a positive coefficient, choose the variable with the “most positive” coefficient in row 0 to enter the basis.

This modification of the simplex algorithm works because increasing a nonbasic variable with a positive coefficient in row 0 will decrease  $z$ . If we use this method to solve LP 2, then our initial tableau will be as shown in Table 11. Because  $x_2$  has the most positive coefficient in row 0, we enter  $x_2$  into the basis. The ratio test says that  $x_2$  should enter the basis in row 1, resulting in Table 12. Because each variable in row 0 has a nonpositive coefficient, this is an optimal tableau. Thus, the optimal solution to LP 2 is (as we have already seen)  $z = -12$ ,  $x_2 = 4$ ,  $s_2 = 10$ ,  $x_1 = s_1 = 0$ .

**TABLE 11.** Initial Tableau for LP 2—Method 2

| $z$ | $x_1$ | $x_2$ | $s_1$ | $s_2$ | rhs | Basic Variable | Ratio               |
|-----|-------|-------|-------|-------|-----|----------------|---------------------|
| 1   | -2    | 3     | 0     | 0     | 0   | $z = 0$        |                     |
| 0   | 1     | 1     | 0     | 0     | 4   | $z_2 = 4$      | $\frac{4}{1} = 4 *$ |
| 0   | 1     | -1    | 0     | 1     | 6   | $s_2 = 6$      | None                |

**TABLE 12.** Optimal Tableau for LP 2—Method 2

| $z$ | $x_1$ | $x_2$ | $s_1$ | $s_2$ | rhs | Basic Variable |
|-----|-------|-------|-------|-------|-----|----------------|
| 1   | -5    | 0     | -3    | 0     | -12 | $z_2 = -12$    |

$$\begin{array}{ccccccc}
 0 & -1 & 1 & -1 & 0 & -14 & \\
 0 & -2 & 0 & -1 & 1 & -10 & \\
 & & & & & & x_2 = 4 \\
 & & & & & & s_2 = 10
 \end{array}$$

To see that this tableau is optimal, note that from row 0,  $z = -12 + 5x_1 + 3s_1$ . Because  $x_1 \geq 0$  and  $s_1 \geq 0$ , this shows that  $z \geq -12$ . Thus, the current bfs (which has  $z = -12$ ) must be optimal.

### 5.4.3. Problems

1. Use the simplex algorithm to find the optimal solution to the following LP:

$$\min z = 4x_1 - x_2$$

s.t.

$$\begin{aligned}
 2x_1 + x_2 &\leq 8 \\
 2x_1 + x_2 &\leq 5 \\
 x_1 - x_2 &\leq 4 \\
 x_1, x_2 &\geq 0
 \end{aligned}$$

2. Use the simplex algorithm to find the optimal solution to the following LP:

$$\min z = -x_1 - x_2$$

s.t.

$$\begin{aligned}
 x_1 - x_2 &\leq 1 \\
 x_1 + x_2 &\leq 2 \\
 x_1, x_2 &\geq 0
 \end{aligned}$$

### 5.5. Unbounded LPs

Recall from Chapter 3 that for some LPs, there exist points in the feasible region for which  $z$  assumes arbitrarily large (in max problems) or arbitrarily small (in min problems) values. When this situation occurs, we say that LP is unbounded. In this section, we show how the simplex algorithm can be used to determine whether an LP is unbounded.

**EXAMPLE 4.** Allbread Bakeries bakes two kinds of bread: French and sourdough. Each loaf of French bread can be sold for 36¢, and each loaf of sourdough bread for 30¢. A loaf of French bread requires 1 yeast packet and 6 oz of flour; sourdough requires 1 yeast packet and 5 oz of flour. At present, Allbread has 5 yeast packets and 10 oz of flour. Additional yeast packets can be purchased at 3¢ each, and additional flour at 4¢/oz. Formulate and solve an LP that can be used to maximize Allbread's profits = (revenues – costs).

**Solution.** Define

$x_1$  = number of loaves of French bread baked

$x_2$  = number of loaves of sourdough bread baked

$x_3$  = number of yeast packets purchased

$x_4$  = number of ounces of flour purchased

Then Allbread's objective is to maximize  $z$  = revenues – costs, where

$$\text{Revenues} = 36x_1 + 30x_2 \quad \text{and} \quad \text{Costs} = 3x_3 + 4x_4$$

Thus, Allbread's objective function is

$$\max z = 36x_1 + 30x_2 - 3x_3 - 4x_4$$

Allbread faces the following two constraints:

**Constraint 1.** Number of yeast packages used to bake bread cannot exceed available yeast plus purchased yeast.

**Constraint 2.** Ounces of flour used to bake breads cannot exceed available flour plus purchased flour.

Because

$$\text{Available yeast} + \text{purchased yeast} = 5 + x_3$$

$$\text{Available flour} + \text{purchased flour} = 10 + x_4$$

Constraint 1 may be written as

$$x_1 + x_2 \leq 5 + x_3 \quad \text{or} \quad x_1 + x_2 - x_3 \leq 5$$

and Constraint 2 may be written as

$$6x_1 + 5x_2 \leq 10 + x_4 \quad \text{or} \quad 6x_1 + 5x_2 - x_4 \leq 10$$

Adding the sign restrictions  $x_i \geq 0$ , ( $i = 1, 2, 3, 4$ ) yields the following LP:

$$\max z = 36x_1 + 30x_2 - 3x_3 - 4x_4$$

s.t.

$$\begin{aligned} x_1 + x_2 - x_3 - x_4 &\leq 5 && (\text{Yeast constraint}) \\ 6x_1 + 5x_2 - x_3 - x_4 &\leq 10 && (\text{Flour constraint}) \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Adding slack variables  $s_1$  and  $s_2$  to the two constraints, we obtain the tableau in Table 13.

**TABLE 13.** Initial Tableau for Allbread

| $z$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | rhs | Basic Variable | Ratio                          |
|-----|-------|-------|-------|-------|-------|-------|-----|----------------|--------------------------------|
| 1   | -36   | -30   | -3    | -4    | 0     | 0     | 10  | $z_2 = 0$      |                                |
| 0   | -61   | -61   | -1    | -0    | 1     | 0     | 15  | $s_1 = 5$      | $\frac{5}{1} = 5$              |
| 0   | -66   | -65   | -0    | -1    | 0     | 1     | 10  | $s_2 = 10$     | $\frac{10}{6} = \frac{5}{3} *$ |

**TABLE 14.** First Tableau for Allbread

| $z$ | $x_1$ | $x_2$         | $x_3$ | $x_4$          | $s_1$ | $s_2$          | rhs            | Basic Variable       | Ratio                                     |
|-----|-------|---------------|-------|----------------|-------|----------------|----------------|----------------------|---|
| 1   | 0     | 0             | 3     | -2             | 0     | 6              | 60             | $z_2 = 60$           |   |
| 0   | 0     | $\frac{1}{6}$ | -1    | $\frac{1}{6}$  | 1     | $-\frac{1}{6}$ | $\frac{10}{3}$ | $s_1 = \frac{10}{3}$ | $\frac{\frac{10}{3}}{\frac{1}{6}} = 20 *$ |
| 0   | 1     | $\frac{5}{6}$ | 0     | $-\frac{1}{6}$ | 0     | $\frac{1}{6}$  | $\frac{5}{3}$  | $s_2 = \frac{5}{3}$  | None                                      |

**TABLE 15.** Second Tableau for Allbread

| $z$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | rhs | Basic Variable | Ratio |
|-----|-------|-------|-------|-------|-------|-------|-----|----------------|-------|
| 1   | 0     | 2     | -9    | 0     | 12    | 4     | 100 | $z_2 = 100$    |       |
| 0   | 0     | 1     | -6    | 1     | 6     | -1    | 20  | $x_4 = 20$     | None  |
| 0   | 1     | 1     | -1    | 0     | 1     | 0     | 5   | $x_1 = 5$      | None  |

Because  $-36 < -30$ , we enter  $x_1$  into the basis. The ratio test indicates that  $x_1$  should enter the basis in row 2. Entering  $x_1$  into the basis in row 2 yields the tableau in Table 14. Because  $x_4$  has the only negative coefficient in row 0, we enter  $x_4$  into the basis. The ratio test indicates

that  $x_4$  should enter the basis in row 1, with the resulting tableau in Table 15. Because  $x_3$  has the most negative coefficient in row 0, we would like to enter  $x_3$  into the basis. The ratio test, however, fails to indicate the row in which  $x_3$  should enter the basis. What is happening? Going back to the basic ideas that led us to the ratio test, we see that as  $x_3$  is increased (holding the other nonbasic variables at zero), the current basic variables,  $x_4$  and  $x_1$ , change as follows:

$$x_4 = 20 + 6x_3 \quad (5.13)$$

$$x_1 = 5 + x_3 \quad (5.14)$$

As  $x_3$  is increased, both  $x_4$  and  $x_1$  increase. This means that no matter how large we make  $x_3$ , the inequalities  $x_4 \geq 0$  and  $x_1 \geq 0$  will still be true. Because each unit by which we increase  $x_3$  will increase  $z$  by 9, we can find points in the feasible region for which  $z$  assumes an arbitrarily large value. For example, can we find a feasible point with  $z > 1,000$ ? To do this, we need to increase  $z$  by  $1,000 - 100 = 900$ . Each unit by which  $x_3$  is increased will increase  $z$  by 9, so increasing  $x_3$  by  $\frac{900}{9} = 100$  should give us  $z = 1,000$ . If we set  $x_3 = 100$  (and hold the other nonbasic variables at zero), then (5.13) and (5.14) show that  $x_4$  and  $x_1$  must now equal

$$x_4 = 20 + 6(100) = 620$$

$$x_1 = 5 + 6(100) = 105$$

Thus,  $x_1 = 105$ ,  $x_3 = 100$ ,  $x_4 = 620$ ,  $x_2 = 0$  is a point in the feasible region with  $z = 1,000$ . In a similar fashion, we can find points in the feasible region having arbitrarily large  $z$ -values. This means the Allbread problem is an unbounded LP.

From the Allbread example, we see that an unbounded LP occurs in a max problem if there is a nonbasic variable with a negative coefficient in row 0 and there is no constraint that limits how large we can make the nonbasic variable. This situation will occur if a nonbasic variable (such as  $x_3$ ) has a negative coefficient in row 0 and nonpositive coefficients in each constraint. To summarize, an unbounded LP for a max problem occurs when a variable with a negative coefficient in row 0 has a nonpositive coefficient in each constraint.

If an LP is unbounded, one will eventually come to a tableau where one wants to enter a variable (such as  $x_3$ ) into the basis, but the ratio test will fail. This is probably the easiest way to spot an unbounded LP.

As we noted in Chapter 4, an unbounded LP is usually caused by an incorrect formulation. In the Allbread example, we obtained an unbounded LP because we allowed Allbread to pay  $3 + 6(4) = 27\text{¢}$  for the ingredients in a loaf of French bread and then sell the loaf for  $36\text{¢}$ . Thus, each loaf of French bread earns a profit of  $9\text{¢}$ . Because unlimited purchases of yeast and flour are allowed, it is clear that our model allows Allbread to manufacture as much French bread as it desires, thereby earning arbitrarily large profits. This is the cause of the unbounded LP.

Of course, our formulation of the Allbread example ignored several aspects of reality. First, we assumed that demand for Allbread's products is unlimited. Second, we ignored the fact that certain resources to make bread (such as ovens and labor) are in limited supply. Finally, we made the unrealistic assumption that unlimited quantities of yeast and flour could be purchased.

### 5.5.1. Unbounded LPs and Directions of Unboundedness

Consider an LP with an objective function  $c_1x_1 + c_2x_2 + \dots + c_nx_n$ . Let  $\mathbf{c} = [c_1 \ c_2 \ \dots \ c_n]$ . If the LP is a maximization problem, then the LP will be unbounded if and only if it has a direction of unboundedness  $\mathbf{d}$  satisfying  $\mathbf{cd} > 0$ . If the LP is a minimization problem, then the LP will be unbounded if and only if it has a direction of unboundedness  $\mathbf{d}$  satisfying  $\mathbf{cd} < 0$ . In Example 3, the last tableau shows us that if we start at the point

$$[5 \ 0 \ 0 \ 20 \ 0 \ 0]^T$$

(the variables are listed in the same order they are listed in the tableau), we can find a direction of unboundedness as follows. Every unit by which  $x_3$  is increased will maintain feasibility if we increase  $x_1$  by one unit and  $x_4$  by six units and leave  $x_2$ ,  $s_1$ , and  $s_2$  unchanged. Because we can increase  $x_3$  without limit, this indicates that

$$\mathbf{d} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 6 \\ 0 \\ 0 \end{bmatrix}$$

is a direction of unboundedness. Because

$$c\mathbf{d} = [36 \quad 30 \quad -3 \quad -4 \quad 0 \quad 0] \begin{bmatrix} 1 \\ 0 \\ 1 \\ 6 \\ 0 \\ 0 \end{bmatrix} = 9$$

we know that LP is unbounded. This follows because each time we move in the direction  $\mathbf{d}$  an amount that increases  $x_3$  by one unit, we increase  $z$  by 9, and we can move as far as we want in the direction  $\mathbf{d}$ .

### 5.5.2. Problems

1. Show that the following LP is unbounded:

$$\max z = 2x_2$$

s.t.

$$\begin{aligned} -x_1 - x_2 &\leq 4 \\ -x_1 + x_2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Find a point in the feasible region with  $z > 10,000$ .

2. State a rule that can be used to determine if a min problem has an unbounded optimal solution (that is,  $z$  can be made arbitrarily small). Use the rule to show that

$$\min z = -2x_1 - 3x_1$$

s.t.

$$\begin{aligned} x_1 - x_2 &\leq 1 \\ x_1 - 2x_2 &\leq 2 \\ x_1, x_2 &\geq 0 \end{aligned}$$

In an unbounded LP.

3. Suppose that in solving an LP, we obtain the tableau in Table 16.

Although  $x_1$  can enter the basis, this LP is unbounded. Why?

TABLE 16

| $z$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | rhs |
|-----|-------|-------|-------|-------|-----|
| 1   | -3    | -2    | 0     | 0     | 0   |
| 0   | 1     | -1    | 1     | 0     | 3   |
| 0   | 2     | 0     | 0     | 1     | 4   |

## 5.6. The Big M Method

Recall that the simplex algorithm requires a starting bfs. In all the problems we have solved so far, we found a starting bfs by using the slack variables as our basic variables. If an LP has any  $>$  or equality constraints, however, a starting bfs may not be readily apparent. Example 6 will illustrate that a bfs may be hard to find. When a bfs is not readily apparent, the Big M method (or the two-phase simplex method) may be used to solve the problem. In this section, we discuss the Big M method, a version of the simplex algorithm that first finds a bfs by adding “artificial” variables to the problem. The objective function of the original LP must, of course, be modified to ensure that the artificial variables are all equal to 0 at the conclusion of the simplex algorithm. The following example illustrates the Big M method.

### EXAMPLE 5. Orangej

Orangej manufactures an orange-flavored soft drink called Orange Smash by combining orange soda and orange juice. Each ounce of orange soda contains 0.5 oz of sugar and 1 mg of vitamin C. Each ounce of orange juice contains 0.25 oz of sugar and 3 mg of vitamin C. It costs Orangej 2¢ to produce an ounce of orange soda and 3¢ to produce an ounce of orange juice. Orangej’s marketing department has decided that each 10-oz bottle of Orange Smash must contain at least 20 mg of vitamin C and at most 4 oz of sugar. Use linear programming to determine how Orangej can meet the marketing department’s requirements at minimum cost.

**Solution.** Let

$x_1$  = number of ounces of orange soda in a bottle of Orange Smash

$x_2$  = number of ounces of orange juice in a bottle of Orange Smash

Then the appropriate LP is

$$\begin{aligned} \min z &= 2x_1 + 3x_2 \\ \text{s.t.} & \end{aligned} \tag{5.15}$$

$$\begin{aligned} \frac{1}{2}x_1 + \frac{1}{4}x_2 &\leq 4 && \text{(Sugar constraint)} \\ x_1 + 3x_2 &\geq 20 && \text{(Vitamin C constraint)} \\ x_1 + x_2 &= 10 && \text{(10 oz in bottle of Orange Smash)} \\ x_1, x_2 &\geq 0 \end{aligned}$$

(The solution will be continued later in this section.)

To put (5.15) into standard form, we add a slack variable  $s_1$  to the sugar constraint and subtract an excess variable  $e_2$  from the vitamin C constraint. After writing the objective function as  $z - 2x_1 - 3x_2 = 0$ , we obtain the following standard form:

$$\begin{array}{lll} \text{Row 0: } & z - 2x_1 - 3x_2 + s_1 - e_2 & = 0 \\ \text{Row 1: } & \frac{1}{2}x_1 + \frac{1}{4}x_2 + s_1 & = 4 \\ \text{Row 2: } & 3x_2 & - e_2 = 20 \\ \text{Row 3: } & x_1 + x_2 & = 10 \end{array}$$

All variables nonnegative

In searching for a bfs, we see that  $s_1 = 4$  could be used as a basic (and feasible) variable for row 1. If we multiply row 2 by  $-1$ , we see that  $e_2 = -20$  could be used as a basic variable for row 2. Unfortunately,  $e_2 = -20$  violates the sign restriction  $e_2 \geq 0$ . Finally, in row 3 there is no readily apparent basic variable. Thus, in order to use the simplex to solve (5.15), rows 2 and 3 each need a basic (and feasible) variable. To remedy this problem, we simply “invent” a basic feasible variable for each constraint that needs one. Because these variables are created by us and are not real variables, we call them artificial variables. If an artificial variable is added to row  $i$ , we label it  $a_i$ . In the current problem, we need

to add an artificial variable  $a_2$  to row 2 and an artificial variable  $a_3$  to row 3. The resulting set of equations is

$$\begin{aligned}
 z - 2x_1 - 3x_2 &= 0 \\
 \frac{1}{2}x_1 + \frac{1}{4}x_2 + s_1 &= 4 \\
 x_1 + 3x_2 - e_2 + a_2 &= 20 \\
 x_1 + x_2 + a_3 &= 10
 \end{aligned} \tag{5.16}$$

We now have a bfs:  $z = 0$ ,  $s_1 = 4$ ,  $a_2 = 20$ ,  $a_3 = 10$ . Unfortunately, there is no guarantee that the optimal solution to (5.16) will be the same as the optimal solution to (5.15). In solving (5.16), we might obtain an optimal solution in which one or more artificial variables are positive. Such a solution may not be feasible in the original problem (5.15). For example, in solving (5.16), the optimal solution may easily be shown to be  $z = 0$ ,  $s_1 = 4$ ,  $a_2 = 20$ ,  $a_3 = 10$ ,  $x_1 = x_2 = 0$ . This “solution” contains no vitamin C and puts 0 ounces of soda in a bottle, so it cannot possibly solve our original problem. If the optimal solution to (5.16) is to solve (5.15), then we must make sure that the optimal solution to (5.16) sets all artificial variables equal to zero. In a min problem, we can ensure that all the artificial variables will be zero by adding a term  $Ma_i$  to the objective function for each artificial variable  $a_i$ . (In a max problem, add a term  $-Ma_i$  to the objective function.) Here  $M$  represents a “very large” positive number. Thus, in (5.16), we would change our objective function to

$$\min z = 2x_1 + 3x_2 + Ma_2 + Ma_3$$

Then row 0 will change to

$$z - 2x_1 - 3x_2 - Ma_2 - Ma_3 = 0$$

Modifying the objective function in this way makes it extremely costly for an artificial variable to be positive. With this modified objective function, it seems reasonable that the optimal solution to (5.16) will have  $a_2 = a_3 = 0$ . In this case, the optimal solution to (5.16) will solve the original problem (5.15). It sometimes happens, however, that in solving the analog of (5.16), some of the artificial variables may assume

positive values in the optimal solution. If this occurs, the original problem has no feasible solution.

For obvious reasons, the method we have just outlined is often called the Big M method. We now give a formal description of the Big M method.

### 5.6.1. Description of Big M Method

**Step 1.** Modify the constraints so that the right-hand side of each constraint is non-negative. This requires that each constraint with a negative right-hand side be multiplied through by  $-1$ . Remember that if you multiply an inequality by any negative number, the direction of the inequality is reversed. For example, our method would transform the inequality  $x_1 + x_2 \geq -1$  into  $-x_1 - x_2 \leq 1$ . It would also transform  $x_1 - x_2 \leq -2$  into  $-x_1 + x_2 \geq 2$ .

**Step 1'.** Identify each constraint that is now (after step 1) an  $=$  or  $\geq$  constraint. In step 3, we will add an artificial variable to each of these constraints.

**Step 2.** Convert each inequality constraint to standard form. This means that if constraint  $i$  is a  $\leq$  constraint, we add a slack variable  $s_i$ , and if constraint  $i$  is a  $\geq$  constraint, we subtract an excess variable  $e_i$ .

**Step 3.** If (after step 1 has been completed) constraint  $i$  is a  $\geq$  or  $=$  constraint, add an artificial variable  $a_i$ . Also add the sign restriction  $a_i > 0$ .

**Step 4.** Let  $M$  denote a very large positive number. If the LP is a min problem, add (for each artificial variable)  $Ma_i$  to the objective function. If the LP is a max problem, add (for each artificial variable)  $-Ma_i$  to the objective function.

**Step 5.** Because each artificial variable will be in the starting basis, all artificial variables must be eliminated from row 0 before beginning the simplex. This ensures that we begin with a canonical form. In choosing the entering variable, remember that  $M$  is a very large positive number. For example,  $4M - 2$  is more positive than  $3M + 900$ , and  $-6M - 5$  is more negative than  $-5M - 40$ . Now solve the transformed problem by the simplex. If all artificial variables are equal to zero in the optimal

solution, then we have found the optimal solution to the original problem. If any artificial variables are positive in the optimal solution, then the original problem is infeasible.

When an artificial variable leaves the basis, its column may be dropped from future tableaus because the purpose of an artificial variable is only to get a starting basic feasible solution. Once an artificial variable leaves the basis, we no longer need it. Despite this fact, we often maintain the artificial variables in all tableaus.

**Solution.** Example 5 (Continued)

**Step 1.** Because none of the constraints has a negative right-hand side, we don't have to multiply any constraint through by  $-1$ .

**Step 2.** Constraints 2 and 3 will require artificial variables.

**Step 3.** Add a slack variable  $s_1$  to row 1 and subtract an excess variable  $e_2$  from row 2. The result is

$$\min z = 2x_1 + 3x_2$$

$$\text{Row 1: } \frac{1}{2}x_1 + \frac{1}{4}x_2 + s_1 = 4$$

$$\text{Row 2: } x_1 + 3x_2 - e_2 = 20$$

$$\text{Row 3: } x_1 + x_2 = 10$$

**Step 4.** Add an artificial variable  $a_2$  to row 2 and an artificial variable  $a_3$  to row 3. The result is

$$\min z = 2x_1 + 3x_2$$

$$\text{Row 1: } \frac{1}{2}x_1 + \frac{1}{4}x_2 + s_1 = 4$$

$$\text{Row 2: } x_1 + 3x_2 - e_2 + a_2 = 20$$

$$\text{Row 3: } x_1 + x_2 + a_3 = 10$$

From this tableau, we see that our initial bfs will be  $s_1 = 4$ ,  $a_2 = 20$ , and  $a_3 = 10$ .

**Step 5.** Because we are solving a min problem, we add  $Ma_2 + Ma_3$  to the objective function (if we were solving a max problem, we would add  $-Ma_2 - Ma_3$ ). This makes  $a_2$  and  $a_3$  very unattractive, and the act of

minimizing  $z$  will cause  $a_2$  and  $a_3$  to be zero. The objective function is now

$$\min z = 2x_1 + 3x_2 + Ma_2 + Ma_3$$

**Step 6.** Row 0 is now

$$z - 2x_1 - 3x_2 - Ma_2 - Ma_3 = 0$$

Because  $a_2$  and  $a_3$  are in our starting bfs (that's why we introduced them), they must be eliminated from row 0. To eliminate  $a_2$  and  $a_3$  from row 0, simply replace row 0 by row 0 +  $M$ (row 2) +  $M$ (row 3). This yields

$$\begin{aligned} \text{Row 0: } z - 2x_1 - 3x_2 - Ma_2 - Ma_3 &= 0 \\ M(\text{row 2}): \quad Mx_1 + 3Mx_2 - Me_2 + Ma_2 &= 20M \\ M(\text{row 3}): \quad Mx_1 + Mx_2 &+ Ma_3 = 10M \\ \text{New row 0: } z + (2M - 2)x_1 + (4M - 3)x_2 - Me_2 &= 30M \end{aligned}$$

Combining the new row 0 with rows 1–3 yields the initial tableau shown in Table 17.

We are solving a min problem, so the variable with the most positive coefficient in row 0 should enter the basis. Because  $4M - 3 > 2M - 2$ , variable  $x_2$  should enter the basis. The ratio test indicates that  $x_2$  should enter the basis in row 2, which means the artificial variable  $a_2$  will leave the basis. The most difficult part of doing the pivot is eliminating

**TABLE 17.** Initial Tableau for Orangej

| $z$ | $x_1$         | $x_2$         | $s_1$ | $e_2$ | $a_2$ | $a_3$ | rhs   | Basic Variable | Ratio            |
|-----|---------------|---------------|-------|-------|-------|-------|-------|----------------|------------------|
| 1   | $2M - 2$      | $4M - 3$      | 0     | $-M$  | 0     | 0     | $30M$ | $z_2 = 30M$    |                  |
| 0   | $\frac{1}{2}$ | $\frac{1}{4}$ | 1     | 0     | 0     | 0     | 4     | $s_1 = 4$      | 16               |
| 0   | 1             | 3             | 0     | -1    | 1     | 0     | 20    | $a_2 = 20$     | $\frac{20}{3} *$ |
| 0   | 1             | 1             | 0     | 0     | 0     | 1     | 10    | $a_3 = 10$     | 10               |

**TABLE 18.** First Tableau for Orangej

| $z$ | $x_1$            | $x_2$ | $s_1$ | $e_2$           | $a_2$            | $a_3$ | rhs                | Basic Variable         | Ratio |
|-----|------------------|-------|-------|-----------------|------------------|-------|--------------------|------------------------|-------|
| 1   | $\frac{2M-3}{3}$ | 0     | 0     | $\frac{M-3}{3}$ | $\frac{3-4M}{3}$ | 0     | $\frac{60+10M}{3}$ | $z = \frac{60+10M}{2}$ |       |

$$\begin{array}{ccccccc}
 0 & \frac{5}{12} & 0 & 1 & \frac{1}{12} & -\frac{1}{12} & 0 & \frac{7}{3} & s_1 = \frac{7}{3} & \frac{28}{5} \\
 0 & \frac{1}{3} & 1 & 0 & -\frac{1}{3} & \frac{1}{3} & 0 & \frac{20}{3} & x_2 = \frac{20}{3} & 20^* \\
 0 & \frac{2}{3} & 0 & 0 & \frac{1}{3} & -\frac{1}{3} & 1 & \frac{10}{3} & a_3 = \frac{10}{3} & 5^*
 \end{array}$$

$x_2$  from row 0. First, replace row 2 by 1 (row 2). Thus, the new row 2 is

$$\frac{1}{3}x_1 + x_2 - \frac{1}{3}e_2 + \frac{1}{3}a_2 = \frac{20}{3}$$

We can now eliminate  $x_2$  from row 0 by adding  $-(4M - 3)$  (new row 2) to row 0 or  $(3 - 4M)$  (new row 2) + row 0. Now

$$\begin{aligned}
 (3 - 4M)(\text{new row 2}) &= \\
 \frac{3 - 4M}{3}x_1 + (3 - 4M)x_2 - \frac{3 - 4M}{3}e_2 + \frac{3 - 4M}{3}a_2 &= \frac{20(3 - 4M)}{3} \\
 \text{Row 0: } z + (2M - 2)x_1 + (4M - 3)x_2 - Me_2 &= 30M \\
 \text{New row 0: } z + \frac{2M - 3}{3}x_1 + \frac{M - 3}{3}e_2 + \frac{3 - 4M}{3}a_2 &= \frac{60 + 10M}{3}
 \end{aligned}$$

After using EROs to eliminate  $x_2$  from row 1 and row 3, we obtain the tableau in Table 18. Because  $\frac{2M-3}{3} > \frac{M-3}{3}$ , we next enter  $x_1$  into the basis. The ratio test indicates that  $x_1$  should enter the basis in the third row of the current tableau. Then  $a_3$  will leave the basis, and our next tableau will have  $a_2 = a_3 = 0$ . To enter  $x_1$  into the basis in row 3, we first replace row 3 by 3 (row 3). Thus, new row 3 will be

$$x_1 + \frac{e_2}{2} - \frac{a_2}{2} + \frac{3a_3}{2} = 5$$

To eliminate  $x_1$  from row 0, we replace row 0 by row 0 +  $(3 - 2M)$  (new row 3)/3.

$$\begin{aligned}
 \text{Row 0: } z + \frac{(2M - 3)}{3}x_1 + \frac{M - 3}{3}e_2 + \frac{3 - 4M}{3}a_2 &= \frac{60 + 10M}{3} \\
 \frac{(3 - 2M)(\text{new row 3})}{3} : \frac{3 - 2M}{3}x_1 + \frac{3 - 2M}{6}e_2 + \frac{2M - 3}{6}a_2 & \\
 + \frac{3 - 2M}{2}a_3 &= \frac{15 - 10M}{3} \\
 \text{New row 0: } z + \frac{e_2}{2} + \frac{1 - 2M}{2}a_2 + \frac{3 - 2M}{2}a_3 &= 25
 \end{aligned}$$

New row 1 and new row 2 are computed as usual, yielding the tableau in Table 19. Because all variables in row 0 have nonpositive coefficients, this is an optimal tableau; all artificial variables are equal to zero in this tableau, so we have found the optimal solution to the Orangej problem:

$z = 25, x_1 = 5, x_2 = 5, s_1 = \frac{1}{4}, e_2 = 0$ . This means that Orangej can hold the cost of producing a 10-oz bottle of Orange Smash to 25¢ by mixing 5 oz of orange soda and 5 oz of orange juice. Note that the  $a_2$  column could have been dropped after  $a_2$  left the basis (at the conclusion of the first pivot), and the  $a_3$  column could have been dropped after  $a_3$  left the basis (at the conclusion of the second pivot).

**TABLE 19.** Optimal Tableau for Orangej

| $z$ | $x_1$ | $x_2$ | $s_1$ | $e_2$          | $a_2$            | $a_3$            | rhs           | Basic Variable      |
|-----|-------|-------|-------|----------------|------------------|------------------|---------------|---------------------|
| 1   | 0     | 0     | 0     | $-\frac{1}{2}$ | $\frac{1-2M}{2}$ | $\frac{3-2M}{2}$ | 25            | $z = 25$            |
| 0   | 0     | 0     | 1     | $-\frac{1}{8}$ | $\frac{1}{8}$    | $-\frac{5}{8}$   | $\frac{1}{4}$ | $s_1 = \frac{1}{4}$ |
| 0   | 0     | 1     | 0     | $-\frac{1}{2}$ | $\frac{1}{2}$    | $-\frac{1}{2}$   | 5             | $x_2 = 5$           |
| 0   | 1     | 0     | 0     | $\frac{1}{2}$  | $-\frac{1}{2}$   | $\frac{3}{2}$    | 5             | $x_1 = 5$           |

### 5.6.2. How to Spot an Infeasible LP

We now modify the Orangej problem by requiring that a 10-oz bottle of Orange Smash contain at least 36 mg of vitamin C. Even 10 oz of orange juice contain only  $3(10) = 30$  mg of vitamin C, so we know that Orangej cannot possibly meet the new vitamin C requirement. This means that Orangej's LP should now have no feasible solution. Let's see how the Big M method reveals the LP's infeasibility. We have changed Orangej's LP to

$$\begin{aligned} \min z &= 2x_1 + 3x_2 \\ \text{s.t.} & \\ \frac{1}{2}x_1 + \frac{1}{4}x_2 &\leq 4 \quad (\text{Sugar constraint}) \\ x_1 + 3x_2 &\geq 20 \quad (\text{Vitamin C constraint}) \end{aligned} \tag{5.17}$$

$$\begin{aligned}x_1 + x_2 &= 10 \quad (10 \text{ oz in bottle of Orange Smash}) \\x_1, x_2 &\geq 0\end{aligned}$$

After going through Steps 1–6 of the Big M method, we obtain the initial tableau in Table 20. Because  $4M - 3 > 2M - 2$ , we enter  $x_2$  into the basis. The ratio test indicates that  $x_2$  should be entered in row 3, causing  $a_3$  to leave the basis. After entering  $x_2$  into the basis, we obtain the tableau in Table 21. Because each variable has a nonpositive coefficient in row 0, this is an optimal tableau. The optimal solution indicated by this tableau is  $z = 30 + 6M$ ,  $s_1 = 3$ ,  $a_2 = 6$ ,  $x_2 = 10$ ,  $a_3 = e_2 = x_1 = 0$ . An artificial variable ( $a_2$ ) is positive in the optimal tableau, so Step 5 shows that the original LP has no feasible solution. In summary, if any artificial variable is positive in the optimal Big M tableau, then the original LP has no feasible solution.

**TABLE 20.** Initial Tableau for Orangej (Infeasible)

| $z$ | $x_1$         | $x_2$         | $s_1$ | $e_2$ | $a_2$ | $a_3$ | rhs   | Basic Variable | Ratio  |
|-----|---------------|---------------|-------|-------|-------|-------|-------|----------------|--------|
| 1   | $2M$          | $4M$          | 0     | $-M$  | 0     | 0     | $46M$ | $z_2 = 46M$    |        |
|     | $-2$          | $-3$          |       |       |       |       |       |                |        |
| 0   | $\frac{1}{2}$ | $\frac{1}{4}$ | 1     | 0     | 0     | 0     | 4     | $s_1 = 4$      | 16     |
| 0   | 1             | 3             | 0     | -1    | 1     | 0     | 36    | $a_2 = 36$     | 12     |
| 0   | 1             | 1             | 0     | 0     | 0     | 1     | 10    | $a_3 = 10$     | $10^*$ |

To explain why (19) can have no feasible solution, suppose that it does  $(x_1, x_2)$ . Clearly, if we set  $a_3 = a_2 = 0$ ,  $(x_1, x_2)$  will be feasible for our modified LP (the LP with artificial variables). If we substitute  $(x_1, x_2)$  into the modified objective function ( $z = 2x_1 + 3x_2 + Ma_2 + Ma_3$ ), we obtain  $z = 2x_1 + 3x_2$  (this follows because  $a_3 = a_2 = 0$ ). Because  $M$  is large, this  $z$ -value is certainly less than  $6M + 30$ . This contradicts the fact that the best  $z$ -value for our modified objective function is  $6M + 30$ . This means that our original LP (19) must have no feasible solution.

**TABLE 21.** Tableau Indicating Infeasibility for Orangej (Infeasible)

| $z$ | $x_1$         | $s_2$ | $s_1$ | $e_2$ | $a_2$ | $a_3$         | rhs           | Basic Variable      |
|-----|---------------|-------|-------|-------|-------|---------------|---------------|---------------------|
| 1   | $1 - 2M$      | 0     | 0     | $-M$  | 0     | $3 - 4M$      | $30 + 6M$     | $z_2 = 6M + 30$     |
| 0   | $\frac{1}{4}$ | 0     | 1     | 0     | 0     | $\frac{1}{4}$ | $\frac{3}{2}$ | $s_1 = \frac{3}{2}$ |

$$\begin{array}{ccccccccc} 0 & -2 & 0 & 0 & -1 & 1 & -3 & 6 & \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 & 10 & \\ \end{array} \quad \begin{array}{l} a_2 = 6 \\ x_2 = 10 \end{array}$$

Note that when the Big M method is used, it is difficult to determine how large  $M$  should be. Generally,  $M$  is chosen to be at least 100 times larger than the largest coefficient in the original objective function. The introduction of such large numbers into the problem can cause roundoff errors and other computational difficulties. For this reason, most computer codes solve LPs by using the two-phase simplex method.

### 5.6.3. Problems

Use the Big M method to solve the following LPs:

1.  $\min z = 4x_1 + 4x_2 + x_3$

s.t.

$$\begin{aligned} 2x_1 + x_2 + 3x_3 &\leq 2 \\ 2x_1 + x_2 &\leq 3 \\ 2x_1 + x_2 + 3x_3 &\geq 3 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

2.  $\min z = 2x_1 + 3x_2$

s.t.

$$\begin{aligned} 2x_1 + x_2 &\geq 4 \\ x_1 - x_2 &\geq -1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

3.  $\max z = 3x_1 + x_2$

s.t.

$$\begin{aligned} x_1 + x_2 &\geq 3 \\ 2x_1 + x_2 &\leq 4 \\ x_1 + x_2 &= 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

## 5.7. Multiattribute Decision Making in the Absence of Uncertainty: Goal Programming

In some situations, a decision maker may face multiple objectives, and there may be no point in an LP's feasible region satisfying all objectives. In such a case, how can the decision maker choose a satisfactory decision? Goal programming is one technique that can be used in such situations. The following example illustrates the main ideas of goal programming.

### EXAMPLE 6. Truthful Goal Programming

The Iam Truthful Advertising Agency is trying to determine a TV advertising schedule for Affordable Auto Company. Affordable has three goals:

**Goal 1.** Its ads should be seen by at least 40 million high-income men (HIM).

**Goal 2.** Its ads should be seen by at least 60 million low-income people (LIP).

**Goal 3.** Its ads should be seen by at least 35 million high-income women (HIW).

Iam Truthful can purchase two types of ads: those shown during football games and those shown during soap operas. At most, \$600,000 can be spent on ads. The advertising costs and potential audiences of a one-minute ad of each type are shown in Table 52. Iam Truthful must determine how many football ads and soap opera ads to purchase for Affordable.

**Solution.** Let

$x_1$  = number of minutes of ads shown during football games

$x_2$  = number of minutes of ads shown during soap operas

Then any feasible solution to the following LP would meet Affordable's goals:

$$\min(\text{or } \max) z = 0x_1 + 0x_2 \quad \text{or any other objective function}$$

$$\text{s.t.} \quad (5.18)$$

$$7x_1 + 3x_2 \geq 40 \quad (\text{HIM constraint})$$

$$\begin{aligned}
10x_1 + 5x_2 &\geq 60 && \text{(LIP constraint)} \\
5x_1 + 4x_2 &\geq 35 && \text{(HIW constraint)} \\
100x_1 + 60x_2 &\leq 600 && \text{(Budget constraint)} \\
x_1, x_2 &\geq 0
\end{aligned}$$

From Figure 14, we find that no point that satisfies the budget constraint meets all three of Affordable's goals. Thus, (21) has no feasible solution. It is impossible to meet all of Affordable's goals, so Truthful might ask Affordable to identify, for each goal, a cost (per-unit short of meeting each goal) that is incurred for failing to meet the goal. Suppose Affordable determines that

**TABLE 22.** Cost and Number of Viewers of Ads for Affordable

| Millions of Viewers |     |     |     |           |
|---------------------|-----|-----|-----|-----------|
| Ad                  | HIM | LIP | HIW | Cost (\$) |
| Football            | 7   | 10  | 5   | 100,000   |
| Soap opera          | 3   | 5   | 4   | 160,000   |

Each million exposures by which Affordable falls short of the HIM goal costs Affordable a \$200,000 penalty because of lost sales.

Each million exposures by which Affordable falls short of the LIP goal costs Affordable a \$100,000 penalty because of lost sales.

Each million exposures by which Affordable falls short of the HIW goal costs Affordable a \$50,000 penalty because of lost sales.

Truthful can now formulate an LP that minimizes the cost incurred in deviating from Affordable's three goals. The trick is to transform each inequality constraint in (5.18) that represents one of Affordable's goals into an equality constraint. Because we don't know whether the cost-minimizing solution will undersatisfy or oversatisfy a given goal, we need to define the following variables:

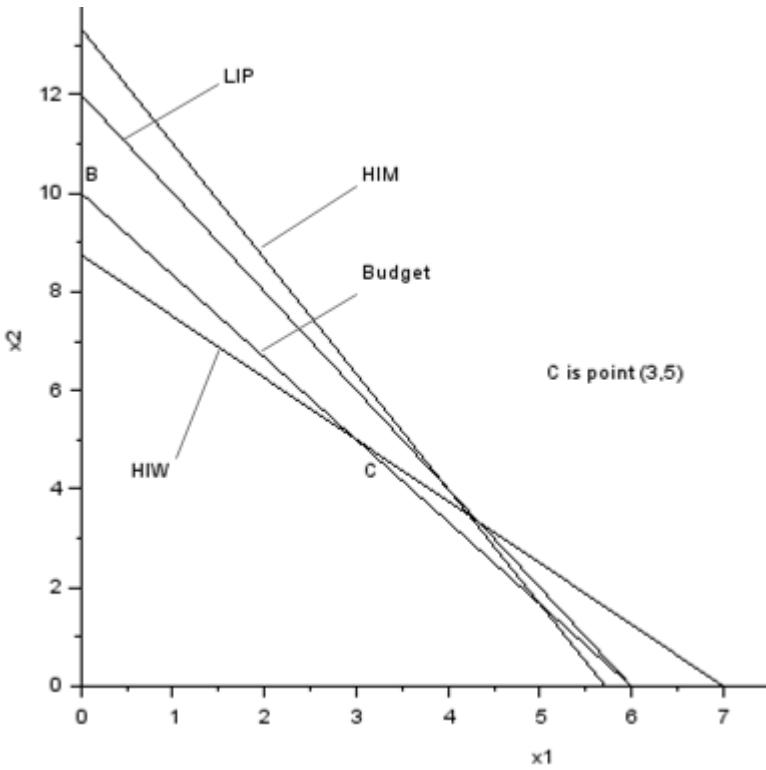
$s_i^+$  = amount by which we numerically exceed the  $i$ th goal

$s_i^-$  = amount by which we are numerically under the  $i$ th goal

The  $s_i^+$  and  $s_i^-$  are referred to as deviational variables. For the Affordable problem, we assume that each  $s_i^+$  and  $s_i^-$  is measured in

millions of exposures. Using the deviational variables, we can rewrite the first three constraints in (5.18) as

$$\begin{aligned} 7x_1 + 3x_2 + s_1^- - s_1^+ &= 40 && \text{(HIM constraint)} \\ 10x_1 + 5x_2 + s_2^- - s_2^+ &= 60 && \text{(LIP constraint)} \\ 5x_1 + 4x_2 + s_3^- - s_3^+ &= 35 && \text{(HIW constraint)} \end{aligned}$$



**FIGURE 14.** Constraints for Affordable (made with SCILAB)

For example, suppose that  $x_1 = 5$  and  $x_2 = 2$ . This advertising schedule yields  $7(5) + 3(2) = 41$  million HIM exposures. This exceeds the HIM goal by  $41 - 40 = 1$  million exposures, so  $s_1^- = 0$  and  $s_1^+ = 1$ . Also, this schedule yields  $10(5) + 5(2) = 60$  million LIP exposures. This exactly meets the LIP requirement, and  $s_2^- = s_2^+ = 0$ . Finally, this schedule yields  $5(5) + 4(2) = 33$  million HIW exposures. We are numerically under the HIW goal by  $35 - 33 = 2$  million exposures, so  $s_3^- = 2$  and  $s_3^+ = 0$ .

Suppose Affordable wants to minimize the total penalty from the lost sales. In terms of the deviational variables, the total penalty from lost sales (in thousands of dollars) caused by deviation from the three goals is  $200s_1^- + 100s_2^- + 50s_3^-$ . The objective function coefficient for the variable associated with goal  $i$  is called the weight for goal  $i$ . The most important goal has the largest weight, and so on. Thus, in the Affordable example, goal 1 (HIM) is most important, goal 2 (LIP) is second most important, and goal 3 (HIW) is least important.

Truthful can minimize the penalty from Affordable's lost sales by solving the following LP:

$$\begin{aligned} \min z &= 200s_1^- + 100s_2^- + 50s_3^- \\ \text{s.t.} & \\ 7x_1 + 3x_2 + s_1^- - s_1^+ &= 40 \quad (\text{HIM constraint}) \\ 10x_1 + 5x_2 + s_2^- - s_2^+ &= 60 \quad (\text{LIP constraint}) \\ 5x_1 + 4x_2 + s_3^- - s_3^+ &= 35 \quad (\text{HIW constraint}) \\ 100x_1 + 60x_2 &\leq 600 \quad (\text{Budget constraint}) \end{aligned} \tag{5.19}$$

All variables nonnegative

The optimal solution to this LP is  $z = 250$ ,  $x_1 = 6$ ,  $x_2 = 0$ ,  $s_1^+ = 2$ ,  $s_2^+ = 0$ ,  $s_3^+ = 0$ ,  $s_1^- = 0$ ,  $s_2^- = 0$ ,  $s_3^- = 5$ . This meets goal 1 and goal 2 (the goals with the highest costs, or weights, for each unit of deviation from the goal) but fails to meet the least important goal (goal 3).

**REMARKS.** If failure to meet goal  $i$  occurs when the attained value of an attribute is numerically smaller than the desired value of goal  $i$ , then a term involving  $s_i^-$  will appear in the objective function. If failure to meet goal  $i$  occurs when the attained value of an attribute is numerically larger than the desired value of goal  $i$ , then a term involving  $s_i^+$  will appear in the objective function. Also, if we want to meet a goal exactly and a penalty is assessed for going both over and under a goal, then terms involving both  $s_i^-$  and  $s_i^+$  will occur in the objective function.

Suppose we modify the Affordable example by deciding that the budget restriction of \$600,000 is a goal. If we decide that a \$1 penalty is assessed for each dollar by which this goal is unmet, then the appropriate goal programming formulation would be

$$\begin{aligned} \min z &= 200s_1^- + 100s_2^- + 50s_3^- + s_4^+ \\ \text{s.t.} & \end{aligned} \tag{5.20}$$

$$\begin{aligned} 7x_1 + 3x_2 + s_1^- - s_1^+ &= 40 && (\text{HIM constraint}) \\ 10x_1 + 5x_2 + s_2^- - s_2^+ &= 60 && (\text{LIP constraint}) \\ 5x_1 + 4x_2 + s_3^- - s_3^+ &= 35 && (\text{HIW constraint}) \\ 100x_1 + 60x_2 + s_4^- + s_4^+ &\leq 600 && (\text{Budget constraint}) \end{aligned}$$

All variables nonnegative

In contrast to our previous optimal solution, the optimal solution to this LP is  $z = 33\frac{1}{3}$ ,  $x_1 = 4\frac{1}{3}$ ,  $x_2 = 3\frac{1}{3}$ ,  $s_1^+ = \frac{1}{3}$ ,  $s_2^+ = 0$ ,  $s_3^+ = 0$ ,  $s_4^+ = 33\frac{1}{3}$ ,  $s_1^- = 0$ ,  $s_2^- = 0$ ,  $s_3^- = 0$ ,  $s_4^- = 0$ . Thus, when we define the budget restriction to be a goal, the optimal solution is to meet all three advertising goals by going  $\$33\frac{1}{3}$  thousand over budget.

### 5.7.1. Preemptive Goal Programming

In our LP formulation of the Truthful example, we assumed that Affordable could exactly determine the relative importance of the three goals. For instance, Affordable determined that the HIM goal was  $200/100 = 2$  times as important as the LIP goal, and the LIP goal was  $100/50 = 2$  times as important as the HIW goal. In many situations, however, a decision maker may not be able to determine precisely the relative importance of the goals. When this is the case, preemptive goal programming may prove to be a useful tool. To apply preemptive goal programming, the decision maker must rank his or her goals from the most important (goal 1) to least important (goal  $n$ ). The objective function coefficient for the variable representing goal  $i$  will be  $P_i$ . We assume that

$$P_1 >>> P_2 >>> P_3 >>> \dots >>> P_n$$

Thus, the weight for goal 1 is much larger than the weight for goal 2, the weight for goal 2 is much larger than the weight for goal 3, and so on. This definition of the  $P_1, P_2, \dots, P_n$  ensures that the decision maker first tries to satisfy the most important (goal 1) goal. Then, among all points that satisfy goal 1, the decision maker tries to come as close as possible to satisfying goal 2, and so forth. We continue in this fashion until the

only way we can come closer to satisfying a goal is to increase the deviation from a higher-priority goal.

For the Affordable problem, the preemptive goal programming formulation is obtained from (22) by replacing (22)'s objective function by  $P_1s_1^- + P_2s_2^- + P_3s_3^-$ . Thus, the preemptive goal programming formulation of the Affordable problem is

$$\begin{aligned} \min z &= P_1s_1^- + P_2s_2^- + P_3s_3^- \\ \text{s.t.} & \\ 7x_1 + 3x_2 + s_1^- - s_1^+ &= 40 \quad (\text{HIM constraint}) \\ 10x_1 + 5x_2 + s_2^- - s_2^+ &= 60 \quad (\text{LIP constraint}) \\ 5x_1 + 4x_2 + s_3^- - s_3^+ &= 35 \quad (\text{HIW constraint}) \\ 100x_1 + 60x_2 &\leq 600 \quad (\text{Budget constraint}) \end{aligned} \quad (5.21)$$

All variables nonnegative

Assume the decision maker has  $n$  goals. To apply preemptive goal programming, we must separate the objective function into  $n$  components, where component  $i$  consists of the objective function term involving goal  $i$ . We define

$$z_i = \text{objective function term involving goal } i$$

For the Affordable example,  $z_1 = P_1s_1^-$ ,  $z_2 = P_2s_2^-$ , and  $z_3 = P_3s_3^-$ . Preemptive goal programming problems can be solved by an extension of the simplex known as the goal programming simplex. To prepare a problem for solution by the goal programming simplex, we must compute  $n$  row 0's, with the  $i$ th row 0 corresponding to goal  $i$ . Thus, for the Affordable problem, we have

$$\text{Row 0 (goal 1): } z_1 - P_1s_1^- = 0$$

$$\text{Row 0 (goal 2): } z_2 - P_2s_2^- = 0$$

$$\text{Row 0 (goal 3): } z_3 - P_3s_3^- = 0$$

From (5.21), we find that  $BV = \{s_1^-, s_2^-, s_3^-, s_4\}$  ( $s_4$  = slack variable for fourth constraint) is a starting basic feasible solution that could be used to solve (5.22) via the simplex algorithm (or goal programming simplex

algorithm). As with the regular simplex, we must first eliminate all variables in the starting basis from each row 0. Adding  $P_1$  (HIM constraint) to row 0 (goal 1) yields

$$\text{Row 0 (goal 1): } z_1 + 7P_1x_1 + 3P_1x_2 - P_1s_1^+ = 40P_1 \quad (\text{HIM})$$

Adding  $P_2$  (LIP constraint) to row 0 (goal 2) yields

$$\text{Row 0 (goal 2): } z_2 + 10P_2x_1 + 5P_2x_2 - P_2s_2^+ = 60P_2 \quad (\text{LIP})$$

Adding  $P_3$  (HIW constraint) to row 0 (goal 3) yields

$$\text{Row 0 (goal 3): } z_3 + 5P_3x_1 + 4P_3x_2 - P_3s_3^+ = 35P_3 \quad (\text{HIW})$$

The Affordable problem can now be solved by the goal programming simplex.

The differences between the goal programming simplex and the ordinary simplex are as follows:

1. The ordinary simplex has a single row 0, whereas the goal programming simplex requires  $n$  row 0's (one for each goal).
2. In the goal programming simplex, the following method is used to determine the entering variable: Find the highest-priority goal (goal  $i$ ) that has not been met (or find the highest-priority goal  $i$  having  $z_i > 0$ ). Find the variable with the most positive coefficient in row 0 (goal  $i$ ) and enter this variable (subject to the following restriction) into the basis. This will reduce  $z_i$  and ensure that we come closer to meeting goal  $i$ . If, however, a variable has a negative coefficient in row 0 associated with a goal having a higher priority than  $i$ , then the variable cannot enter the basis. Entering such a variable in the basis would increase the deviation from some higher-priority goal. If the variable with the most positive coefficient in row 0 (goal  $i$ ) cannot be entered into the basis, then try to find another variable with a positive coefficient in row 0 (goal  $i$ ). If no variable for row 0 (goal  $i$ ) can enter the basis, then there is no way to come closer to meeting goal  $i$  without increasing the deviation from some higher-priority goal. In this case, move on to row 0

(goal  $i + 1$ ) in an attempt to come closer to meeting goal  $i + 1$ .

3. When a pivot is performed, row 0 for each goal must be updated.
4. A tableau will yield the optimal solution if all goals are satisfied (that is,  $z_1 = z_2 = \dots = z_n = 0$ ), or if each variable that can enter the basis and reduce the value of  $z_i$  for an unsatisfied goal  $i$  will increase the deviation from some goal  $i$  having a higher priority than goal  $i$ .

We now use the goal programming simplex to solve the Affordable example. In each tableau, the row 0's are listed in order of the goal's priorities (from highest priority to lowest priority). The initial tableau is Table 23. The current bfs is  $s_1^- = 40$ ,  $s_2^- = 60$ ,  $s_3^- = 35$ ,  $s_4 = 600$ . Because  $z_1 = 40P_1$ , goal 1 is not satisfied. To reduce the penalty associated with not meeting goal 1, we enter the variable with the most positive coefficient ( $x_1$ ) in row 0 (HIM). The ratio test indicates that  $x_1$  should enter the basis in the HIM constraint.

**TABLE 23.** Initial Tableau for Preemptive Goal Programming for Affordable

|        |   | $x_1$   | $x_2$  | $s_1^+$ | $s_2^+$ | $s_3^+$ | $s_1^-$ | $s_2^-$ | $s_3^-$ | $s_4$ | rhs           |
|--------|---|---------|--------|---------|---------|---------|---------|---------|---------|-------|---------------|
| Row    | 0 | $7P_1$  | $3P_1$ | $-P_1$  | 0       | 0       | 0       | 0       | 0       | 0     | $z_1 = 40P_1$ |
| (HIM)  |   |         |        |         |         |         |         |         |         |       |               |
| Row    | 0 | $10P_2$ | $5P_2$ | 0       | $-P_2$  | 0       | 0       | 0       | 0       | 0     | $z_2 = 60P_2$ |
| (LIP)  |   |         |        |         |         |         |         |         |         |       |               |
| Row    | 0 | $5P_3$  | $4P_3$ | 0       | 0       | $-P_3$  | 0       | 0       | 0       | 0     | $z_3 = 35P_3$ |
| (HIW)  |   |         |        |         |         |         |         |         |         |       |               |
| HIM    |   | 7       | 3      | -1      | 0       | 0       | 1       | 0       | 0       | 0     | 40            |
| LIP    |   | 10      | 5      | 0       | -1      | 0       | 0       | 1       | 0       | 0     | 60            |
| HIW    |   | 5       | 4      | 0       | 0       | -1      | 0       | 0       | 1       | 0     | 35            |
| Budget |   | 100     | 60     | 0       | 0       | 0       | 0       | 0       | 0       | 1     | 600           |

After entering  $x_1$  into the basis, we obtain Table 24. The current basic solution is  $x_1 = \frac{40}{7}$ ,  $s_2^- = \frac{20}{7}$ ,  $s_3^- = \frac{45}{7}$ ,  $s_4 = \frac{200}{7}$ . Because  $s_1^- = 0$  and  $z_1 = 0$ , goal 1 is now satisfied. We now try to satisfy goal 2 (while ensuring that the higher-priority goal 1 is still satisfied). The variable with the most positive coefficient in row 0 (LIP) is  $s_1^+$ . Observe that entering  $s_1^+$  into the basis will not increase  $z_1$  [because the coefficient of  $s_1^+$  in

row 0 (HIM) is 0]. Thus, after entering  $s_1^+$  into the basis, goal 1 will still be satisfied. The ratio test indicates that  $s_1^+$  could enter the basis in either the LIP or the budget constraint. We arbitrarily choose to enter  $s_1^+$  into the basis in the budget constraint.

**TABLE 24.** First Tableau for Preemptive Goal Programming for Affordable

|        | $x_1$ | $x_2$           | $s_1^+$           | $s_2^+$           | $s_3^+$ | $s_1^-$          | $s_2^-$            | $s_3^-$ | $s_4$ | rhs                     |
|--------|-------|-----------------|-------------------|-------------------|---------|------------------|--------------------|---------|-------|-------------------------|
| Row    | 0     | 0               | 0                 | 0                 | 0       | $-P_1$           | 0                  | 0       | 0     | $z_1 = 0$               |
| (HIM)  |       |                 |                   |                   |         |                  |                    |         |       |                         |
| Row    | 0     | 0               | $\frac{5P_2}{7}$  | $\frac{10P_2}{7}$ | $-P_2$  | 0                | $-\frac{10P_2}{7}$ | 0       | 0     | $z_2 = \frac{20P_2}{7}$ |
| (LIP)  |       |                 |                   |                   |         |                  |                    |         |       |                         |
| Row    | 0     | 0               | $\frac{13P_3}{7}$ | $\frac{5P_3}{7}$  | 0       | $-P_3$           | $-\frac{5P_3}{7}$  | 0       | 0     | $z_3 = \frac{45P_3}{7}$ |
| (HIW)  |       |                 |                   |                   |         |                  |                    |         |       |                         |
| HIM    | 1     | $\frac{3}{7}$   | $-\frac{1}{7}$    | 0                 | 0       | $\frac{1}{7}$    | 0                  | 0       | 0     | $\frac{40}{7}$          |
| LIP    | 0     | $\frac{5}{7}$   | $\frac{10}{7}$    | -1                | 0       | $-\frac{10}{7}$  | 1                  | 0       | 0     | $\frac{20}{7}$          |
| HIW    | 0     | $\frac{13}{7}$  | $\frac{5}{7}$     | 0                 | -1      | $-\frac{5}{7}$   | 0                  | 1       | 0     | $\frac{45}{7}$          |
| Budget | 0     | $\frac{120}{7}$ | $\frac{100}{7}$   | 0                 | 0       | $-\frac{100}{7}$ | 0                  | 0       | 1     | $\frac{200}{7}$         |

After pivoting  $s_1^+$  into the basis, we obtain Table 25. Because  $z_1 = z_2 = 0$ , goals 1 and 2 are met. Because  $z_3 = 5P_3$ , however, goal 3 is unmet. The current bfs is  $x_1 = 6$ ,  $s_2^- = 0$ ,  $s_3^- = 5$ ,  $s_1^+ = 2$ . We now try to come closer to meeting goal 3 (without violating either goal 1 or goal 2). Because  $x_2$  is the only variable with a positive coefficient in row 0 (HIW), the only way to come closer to meeting goal 3 (HIW) is to enter  $x_2$  into the basis. Observe, however, that  $x_2$  has a negative coefficient in row 0 for goal 2 (LIP). Thus, the only way we can come closer to meeting goal 3 (HIW) is to violate a higher-priority goal, goal 2 (LIP). This is therefore an optimal tableau. The preemptive goal programming solution is to purchase 6 minutes of football ads and no soap opera ads. Goals 1 and 2 (HIM and LIP) are met, and Affordable falls 5 million exposures short of meeting goal 3 (HIW).

**TABLE 25.** Optimal Tableau for Preemptive Goal Programming for Affordable

|       | $x_1$ | $x_2$ | $s_1^+$ | $s_2^+$ | $s_3^+$ | $s_1^-$ | $s_2^-$ | $s_3^-$ | $s_4$ | rhs       |
|-------|-------|-------|---------|---------|---------|---------|---------|---------|-------|-----------|
| Row   | 0     | 0     | 0       | 0       | 0       | $-P_1$  | 0       | 0       | 0     | $z_1 = 0$ |
| (HIM) |       |       |         |         |         |         |         |         |       |           |

|        |   |               |        |    |        |        |   |   |                 |                   |              |
|--------|---|---------------|--------|----|--------|--------|---|---|-----------------|-------------------|--------------|
| Row    | 0 | 0             | $-P_2$ | 0  | $-P_2$ | 0      | 0 | 0 | 0               | $-\frac{P_2}{10}$ | $z_2 = 0$    |
| (LIP)  |   |               |        |    |        |        |   |   |                 |                   |              |
| Row    | 0 | 0             | $P_3$  | 0  | 0      | $-P_3$ | 0 | 0 | 0               | $-\frac{P_3}{20}$ | $z_3 = 5P_3$ |
| (HIW)  |   |               |        |    |        |        |   |   |                 |                   |              |
| HIM    | 0 | $\frac{3}{5}$ | 0      | 0  | 0      | 0      | 0 | 0 | $\frac{1}{100}$ | 6                 |              |
| HIW    | 0 | -1            | 0      | -1 | 0      | 0      | 1 | 0 | $-\frac{1}{10}$ | 0                 |              |
| HIW    | 0 | 1             | 0      | 0  | -1     | 0      | 0 | 1 | $-\frac{1}{20}$ | 5                 |              |
| Budget | 0 | $\frac{6}{5}$ | 1      | 0  | 0      | -1     | 0 | 0 | $\frac{7}{100}$ | 2                 |              |

If the analyst has access to a computerized goal programming code, then by reordering the priorities assigned to the goals, many solutions can be generated. From among these solutions, the decision maker can choose a solution that she feels best fits her preferences. Table 26 lists the solutions found by the preemptive goal programming method for each possible set of priorities. Thus, we see that different ordering of priorities can lead to different advertising strategies.

**TABLE 26.** Optimal Solutions for Affordable Found by Preemptive Goal Programming

| Priorities |                |        | Optimal        |                |     | Deviations from |                |  |
|------------|----------------|--------|----------------|----------------|-----|-----------------|----------------|--|
| Highest    | Second Highest | Lowest | $x_1$<br>Value | $x_2$<br>Value | HIM | LIP             | HIW            |  |
| HIM        | LIP            | HIW    | 6              | 0              | 0   | 0               | 5              |  |
| HIM        | HIW            | LIP    | 5              | $\frac{5}{3}$  | 0   | $\frac{5}{3}$   | $\frac{10}{3}$ |  |
| LIP        | HIM            | HIW    | 6              | 0              | 0   | 0               | 5              |  |
| LIP        | HIW            | HIM    | 6              | 0              | 0   | 0               | 5              |  |
| HIW        | HIM            | LIP    | 3              | 5              | 4   | 5               | 0              |  |
| HIW        | LIP            | HIM    | 3              | 5              | 4   | 5               | 0              |  |

When a preemptive goal programming problem involves only two decision variables, the optimal solution can be found graphically. For example, suppose HIW is the highest-priority goal, LIP is the second-highest, and HIM is the lowest. From Figure 14, we find that the set of points satisfying the highest-priority goal (HIW) and the budget

constraint is bounded by the triangle  $ABC$ . Among these points, we now try to come as close as we can to satisfying the second-highest-priority goal (LIP). Unfortunately, no point in triangle  $ABC$  satisfies the LIP goal. We see from the figure, however, that among all points satisfying the highest-priority goal, point  $C$  ( $C$  is where the HIW goal is exactly met and the budget constraint is binding) is the unique point that comes the closest to satisfying the LIP goal. Simultaneously solving the equations

$$\begin{aligned} 5x_1 + 4x_2 &= 35 && (\text{HIW goal exactly met}) \\ 100x_1 + 60x_2 &= 600 && (\text{Budget constraint binding}) \end{aligned}$$

we find that point  $C = (3, 5)$ . Thus, for this set of priorities, the preemptive goal programming solution is to purchase 3 football game ads and 5 soap opera ads.

Goal programming is not the only approach used to analyze multiple objective decision-making problems under certainty. See Steuer (1985) and Zonts and Wallenius (1976) for other approaches to multiple objective decision making under certainty.

### 5.7.2. Problems

- 1 Graphically determine the preemptive goal programming solution to the Affordable example for the following priorities:
  - a. LIP is highest-priority goal, followed by HIW and then HIM.
  - b. HIM is highest-priority goal, followed by LIP and then HIW.
  - c. HIM is highest-priority goal, followed by HIW and then LIP.
  - d. HIW is highest-priority goal, followed by HIM and then LIP.
- 2 The Touche Young accounting firm must complete three jobs during the next month. Job 1 will require 500 hours of work, job 2 will require 300 hours of work, and job 3 will require 100 hours of work. Currently, the firm consists of 5 partners, 5 senior employees, and 5 junior employees, each of whom can work up to 40 hours per month. The dollar amount (per hour) that the company can bill depends on the type of accountant who is assigned to each job, as shown in Table 27. (The X indicates that a junior employee does not have enough experience to work on job 1.) All jobs must be completed. Touche Young has also set the following goals, listed in order of

priority:

**Goal 1.** Monthly billings should exceed \$68,000.

**Goal 2.** At most, 1 partner should be hired.

**Goal 3.** At most, 3 senior employees should be hired.

**Goal 4.** At most, 5 junior employees should be hired.

TABLE 27

|                 | Job 1 | Job 2 | Job 3 |
|-----------------|-------|-------|-------|
| Partner         | 160   | 120   | 110   |
| Senior employee | 120   | 190   | 170   |
| Junior employee | X     | 150   | 140   |

3 A small aerospace company is considering eight projects:

**Project 1** Develop an automated test facility.

**Project 2** Barcode all company inventory and machinery.

**Project 3** Introduce a CAD/CAM system.

**Project 4** Buy a new lathe and deburring system.

**Project 5** Institute FMS (flexible manufacturing system).

**Project 6** Install a LAN (local area network).

**Project 7** Develop AIS (artificial intelligence simulation).

**Project 8** Set up a TQM (total quality management) initiative.

Each project has been rated on five attributes: return on investment (ROI), cost, productivity improvement, worker requirements, and degree of technological risk. These ratings are given in Table 28.

TABLE 28

| Project   | 1     | 2   | 3   | 4   | 5   | 6   | 7     | 8   |
|-----------|-------|-----|-----|-----|-----|-----|-------|-----|
| ROI (\$)  | 2,070 | 456 | 670 | 350 | 495 | 380 | 1,500 | 480 |
| Cost (\$) | 900   | 240 | 335 | 700 | 410 | 190 | 500   | 160 |

|                          |    |    |    |    |    |   |    |    |
|--------------------------|----|----|----|----|----|---|----|----|
| Productivity improvement | 3  | 2  | 2  | 0  | 1  | 0 | 3  | 2  |
| Manpower needed          | 18 | 18 | 27 | 36 | 42 | 6 | 48 | 24 |
| Degree of risk           | 3  | 2  | 4  | 1  | 1  | 0 | 2  | 3  |

The company has set the following five goals (listed in order of priority):

**Goal 1** Achieve a return on investment of at least \$3,250.

**Goal 2** Limit cost to \$1,300.

**Goal 3** Achieve a productivity improvement of at least 6.

**Goal 4** Limit manpower use to 108.

**Goal 5** Limit technological risk to a total of 4.

Use preemptive goal programming to determine which projects should be undertaken.

## 5.8. Linear Programming with *SCILAB*

### EXAMPLE 7.

Max:  $z = 10x_1 + 6x_2 + 4x_3$  subject to:

$$x_1 + x_2 + x_3 \leq 100$$

$$10x_1 + 4x_2 + 4x_3 \leq 600$$

$$2x_1 + 2x_2 + 6x_3 \leq 300$$

$$x_1, x_2, x_3 \geq 0$$

### *SCILAB* Program

```
c = [10, 6, 4]'; #the objective function
a = [ 1, 1, 1;      #constraints
      10, 4, 5;
      2, 2, 6];
b = [100, 600, 300]'; %rhs
lb = [0, 0, 0]'; #lower bound
ub = [];
```

```
[xopt,lagr,fopt]=linpro(c,a,b,lb,ub,3,x0)
fopt =
    733.33333
lagr =
    0.
    0.
    0.
- 4.6666667
- 0.6666667
  0.6666667
xopt =
    33.33333
   66.66667
 - 2.776D-16
```

### EXAMPLE 8. Old McDonald had a Farm

Farmer McDonald grows wheat and corn on his 45-acre farm. He can sell at most 140 bushels of wheat and 120 bushels of corn. Each acre planted with wheat yields 5 bushels, and each acre planted with corn yields 4 bushels. Wheat sells for \$30 per bushel, and corn sells for \$50 per bushel. To harvest an acre of wheat requires 6 hours of labor; 10 hours are needed to harvest an acre of corn. Up to 350 hours of labor can be purchased at \$10 per hour. Let  $A_1$  = acres planted with wheat;  $A_2$  = acres planted with corn; and  $L$  = hours of labor that are purchased. To maximize profits, McDonald should solve the following LP:

$$\max z = 150A_1 + 200A_2 - 10L$$

s.t.

$$\begin{aligned} A_1 + A_2 - L &\leq 45 \\ 6A_1 + 10A_2 - L &\leq 0 \\ L &\leq 350 \\ 5A_1 &\leq 140 \\ 4A_2 &\leq 120 \\ A_1, A_2, L &\geq 0 \end{aligned}$$

- What is McDonald's profit and how much of each crop is optimal for planting?

- b. If only 40 acres of land were available, what would McDonald's profit be and how much of each crop should be planted?
- c. Determine the marginal prices for increasing the acreage for wheat and corn.

### SCILAB Program

We want to solve the linear program:

```
Maximize 150.x1+200.x2-10.x3
such as:
    x1+    x2      <= 45
6.x1+10.x2 - x3 <=0
5.x1            <= 140
        4.x2      <= 120
                  x3 <= 350
x >= 0
```

The solution is:

```
xstar = [25,20,350]
```

The following script solves the problem with the karmarkar function.:

```
c = [-150 -200 +10]';
A = [
    1  1  0
    6 10 -1
    5  0  0
    0  4  0
    0  0  1
];
b = [45 0 140 120 350]';
lb = [0;0;0];
```

The previous script produces the following output.

```
-->[xopt,fopt,exitflag,iter,yopt]
=karmarkar([],[],c,[],[],[],[],A,b,lb)
yopt =
ineqlin: [5x1 constant]
```

```

eqlin: [0x0 constant]
lower: [3x1 constant]
upper: [3x1 constant]
iter =
    89.
exitflag =
    1.
fopt =
 - 4249.9602
xopt =
 25.00115
 19.998673
 349.99469

```

So maximum profit for McDonald's Farm is \$4250. The report also shows that McDonald can maximize profit by planting 25 acres of wheat and 20 acres of corn at a labor cost of \$350.

The following script allows to solve the problem with the linpro function.

```

c = [-150 -200 +10]';
A = [
 1 1 0
 6 10 -1
 5 0 0
 0 4 0
 0 0 1
];
b = [45 0 140 120 350]';
ci=[0 0 0]';
cs=[%inf %inf %inf]';
[xopt,lagr,fopt]=linpro(c,A,b,ci,cs)

```

This produces :

```

-->[xopt,lagr,fopt]=linpro(c,A,b,ci,cs)
fopt =
 - 4250.
lagr =
 0.
 0.
 0.
 75.
 12.5
 0.

```

```
0.  
2.5  
xopt =  
25.  
20.  
350.
```

The marginal price (shadow price) of increasing the acreage for wheat is \$75 per acre and for corn is \$12.50 per acre. This is found in the Lagrange multipliers (lagr).

If only 40 acres can be used, we have:

```
-->b = [40 0 140 120 350]';  
-->[xopt,lagr,fopt]=linpro(c,A,b,ci,cs)  
fopt =  
- 3875.  
lagr =  
0.  
0.  
0.  
75.  
12.5  
0.  
0.  
2.5  
xopt =  
12.5  
27.5  
350.
```

So maximum profit for McDonald's Farm is \$3,875. The report also shows that McDonald can maximize profit by planting 12.5 acres of wheat and 27.5 acres of corn at a labor cost of \$350.

#### EXAMPLE 9.

Minimize       $600x_1 + 600x_2 + 600x_3 + 600x_4 + 600x_5 + 600x_6 + 600x_7 + 200x_8 + 200x_9 + 200x_{10} + 200x_{11} + 200x_{12} + 200x_{13} + 200x_{14}$

such that

$$8x_1 + 8x_4 + 8x_5 + 8x_6 + 8x_7 + 4x_8 + 4x_{11} + 4x_{12} + 4x_{13} + 4x_{14} \geq 88$$

$$8x_1 + 8x_2 + 8x_5 + 8x_6 + 8x_7 + 4x_8 + 4x_9 + 4x_{12} + 4x_{13} + 4x_{14} \geq 136$$

$$8x_1 + 8x_2 + 8x_3 + 8x_6 + 8x_7 + 4x_8 + 4x_9 + 4x_{10} + 4x_{13} + 4x_{14} \geq 104$$

$$8x_1 + 8x_2 + 8x_3 + 8x_4 + 8x_7 + 4x_8 + 4x_9 + 4x_{10} + 4x_{11} + 4x_{14} \geq 120$$

$$8x_1 + 8x_2 + 8x_3 + 8x_4 + 8x_5 + 4x_8 + 4x_9 + 4x_{10} + 4x_{11} + 4x_{12} \geq 152$$

$$8x_2 + 8x_3 + 8x_4 + 8x_5 + 8x_6 + 4x_9 + 4x_{10} + 4x_{11} + 4x_{12} + 4x_{13} \geq 112$$

$$8x_3 + 8x_4 + 8x_5 + 8x_6 + 8x_7 + 4x_{10} + 4x_{11} + 4x_{12} + 4x_{13} + 4x_{14} \geq 128$$

$$-20x_8 - 20x_9 - 20x_{10} - 20x_{11} - 20x_{12} - 20x_{13} - 20x_{14} \geq -840$$

$$x_i \geq 0 \quad (i = 1, \dots, 14)$$

## 5.9. EXCEL Example: Radio Barn

The Radio Barn produces two products: (1) the Shader Walkman, a portable CD/DVD player, and (2) the Shader Watch-TV, a wristwatch-size internet-connected color television. The production process for each product is similar in that both require a certain number of hours of electronic work and a certain number of labor-hours in the assembly department. Each Walkman takes 4 hours of electronic work and 2 hours in the assembly shop. Each Watch-TV requires 3 hours in electronics and 1 hour in assembly. During the current production period, 240 hours of electronic time are available, and 100 hours of assembly department time are available. Each Walkman sold yields a profit of \$7; each Watch-TV produced may be sold for a \$5 profit.

Shader's problem is to determine the best possible combination of Walkmans and Watch-TVs to manufacture to reach the maximum profit. This product-mix situation can be formulated as a linear programming problem.

#### HOURS REQUIRED TO PRODUCE 1 UNIT

| DEPARTMENT      | WALKMANS<br>(X1) | WATCH-TVS<br>(X2) | AVAILABLE<br>HOURS<br>THIS WEEK |
|-----------------|------------------|-------------------|---------------------------------|
| Electronic      | 4                | 3                 | 240                             |
| Assembly        | 2                | 1                 | 100                             |
| Profit per unit | \$7              | \$5               |                                 |

#### 5.9.1. Complete Solution.

The solution for the Excel Example will walk you through the entire process of LP formulation, graphical solution, numerical solution (in Excel) and sensitivity analysis.

We begin by summarizing the information needed to formulate and solve this problem (see Table B.1). Further, let's introduce some simple notation for use in the objective function and constraints. Let

Now we can create the LP *objective function* in terms of  $x_1$  and  $x_2$ :  
 Maximize profit =  $\$7x_1 + \$5x_2$

Our next step is to develop mathematical relationships to describe the two constraints in this problem. One general relationship is that the amount of a resource used is to be less than or equal to ( $\leq$ ) the amount of resource *available*.

*First constraint:* Electronic time used is  $\leq$  Electronic time available.

$$4x_1 + 3x_2 \leq 240 \text{ (hours of electronic time)}$$

*Second constraint:* Assembly time used is  $\leq$  Assembly time available.

$$2x_1 + 1x_2 \leq 100 \text{ (hours of Assembly time)}$$

Both these constraints represent production capacity restrictions and, of course, affect the total profit. For example, Radio Barn cannot produce 70 Walkmans during the production period because if  $x_1 = 70$ , both constraints will be violated. It also cannot make  $x_1 = 50$  Walkmans and  $x_2 = 10$  Watch-TVs. This constraint brings out another important aspect of linear programming; that is, certain interactions will exist between variables. The more units of one product that a firm produces, the fewer it can make of other products.

## 5.10. Graphical Solution to a Linear Programming Problem

The easiest way to solve a small LP problem such as that of the Radio Barn is the graphical solution approach. The graphical procedure can be used only when there are two decision variables (such as number of Walkmans to produce,  $x_1$ , and number of Watch-TVs to produce,  $x_2$ ). When there are more than two variables, it is *not* possible to plot the solution on a two-dimensional graph; we then must turn to more complex approaches described later in this module.

### Graphical Representation of Constraints

To find the optimal solution to a linear programming problem, we must first identify a set, or region, of feasible solutions. The first step in doing so is to plot the problem's constraints on a graph.

The variable  $x_1$  (Walkmans, in our example) is usually plotted as the horizontal axis of the graph, and the variable  $x_2$  (Watch-TVs) is plotted as the vertical axis. The complete problem may be restated as:

$$\text{Maximize } \text{profit} = \$7x_1 + \$5x_2$$

Subject to the constraints

$$4x_1 + 3x_2 \leq 240 \text{ (electronics constraint)}$$

$$2x_1 + 1x_2 \leq 100 \text{ (assembly constraint)}$$

$$x_1 \geq 0$$

(number of Walkmans produced is greater than or equal to 0)

$$x_2 \geq 0$$

(number of Watch-TVs produced is greater than or equal to 0)

The first step in graphing the constraints of the problem is to convert the constraint *inequalities* into *equalsities* (or equations).

**Constraint 1:**  $4x_1 + 3x_2 = 240$

**Constraint 2:**  $2x_1 + 1x_2 = 100$

The equation for constraint A is plotted in Figure 1 and for constraint B in Figure 2.

To plot the line in Figure B.1, all we need to do is to find the points at which the line  $4x_1 + 3x_2 = 240$  intersects the  $x_1$  and  $x_2$  axes. When  $x_1 = 0$  (the location where the line touches the  $x_2$ -axis), it implies that  $3x_2 = 240$  and that  $x_2 = 80$ . Likewise, when  $x_2 = 0$ , we see that  $4x_1 = 240$  and that  $x_1 = 60$ . Thus, constraint A is bounded by the line running from  $(x_1 = 0, x_2 = 80)$  to  $(x_1 = 60, x_2 = 0)$ . The shaded area represents all points that satisfy the original *inequality*.

Constraint B is illustrated similarly in Figure 2. When  $x_1 = 0$ , then  $x_2 = 100$ ; and when  $x_2 = 0$ , then  $x_1 = 50$ . Constraint B, then, is bounded by the line between  $(x_1 = 0, x_2 = 100)$  and  $(x_1 = 50, x_2 = 0)$ . The shaded area represents the original inequality.

Figure B.3 shows both constraints together. The shaded region is the part that satisfies both restrictions. The shaded region in Figure 3 is called the *area of feasible solutions*, or simply the feasible region. This region must satisfy *all* conditions specified by the program's constraints and is thus the region where all constraints overlap. Any point in the region would be a *feasible solution* to the Radio Barn problem. Any point outside the shaded area would represent an *infeasible solution*. Hence, it would be feasible to manufacture 30 Walkmans and 20 Watch-TVs ( $x_1 = 30, x_2 = 20$ ), but it would violate the constraints to produce 70 Walkmans and 40 Watch-TVs. This can be seen by plotting these points on the graph of Figure 3.

### 5.10.1. Iso-Profit Line Solution Method

Now that the feasible region has been graphed, we can proceed to find the *optimal* solution to the problem. The optimal solution is the point lying in the feasible region that produces the highest profit.

Once the feasible region has been established, several approaches can be taken in solving for the optimal solution. The speediest one to apply is called the **iso-profit line** method.

We start by letting profits equal some arbitrary but small dollar amount. For the Radio Barn problem, we may choose a profit of \$210. This is a profit level that can easily be obtained without violating either of the two constraints. The objective function can be written as  $\$210 = 7x_1 + 5x_2$ .

This expression is just the equation of a line; we call it an *iso-profit line*. It represents all combinations (of  $x_1$ ,  $x_2$ ) that will yield a total profit of \$210. To plot the profit line, we proceed exactly as we did to plot a constraint line. First, let  $x_1 = 0$  and solve for the point at which the line crosses the  $x_2$  axis:

$$\$280 = \$7(0) + \$5x_2$$

$$x_2 = 42 \text{ Watch-TVs}$$

Then let  $x_2 = 0$  and solve for  $x_1$ :

$$\$210 = \$7x_1 + \$5(0)$$

$$x_1 = 30 \text{ Walkmans}$$

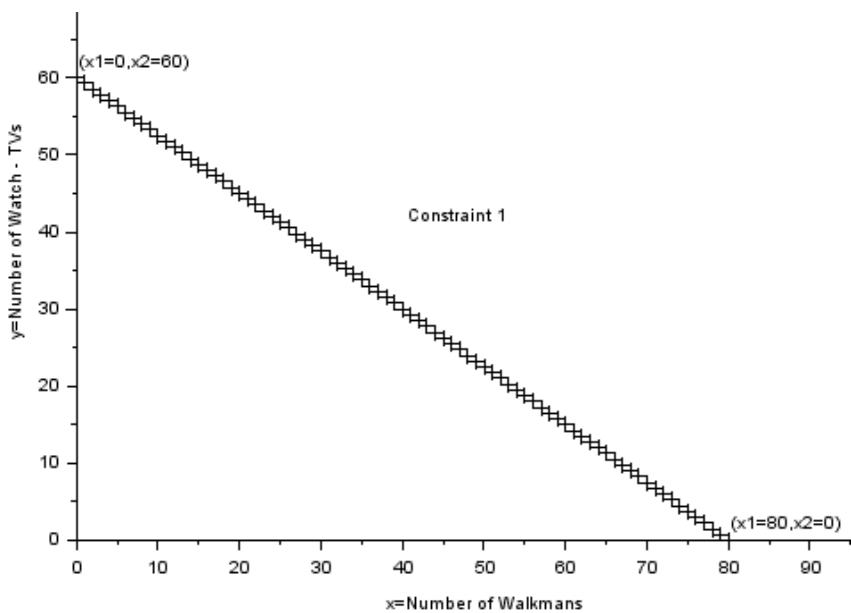
We can now connect these two points with a straight line. This profit line is illustrated in Figure 4.

All points on the line represent feasible solutions that produce a profit of \$210.

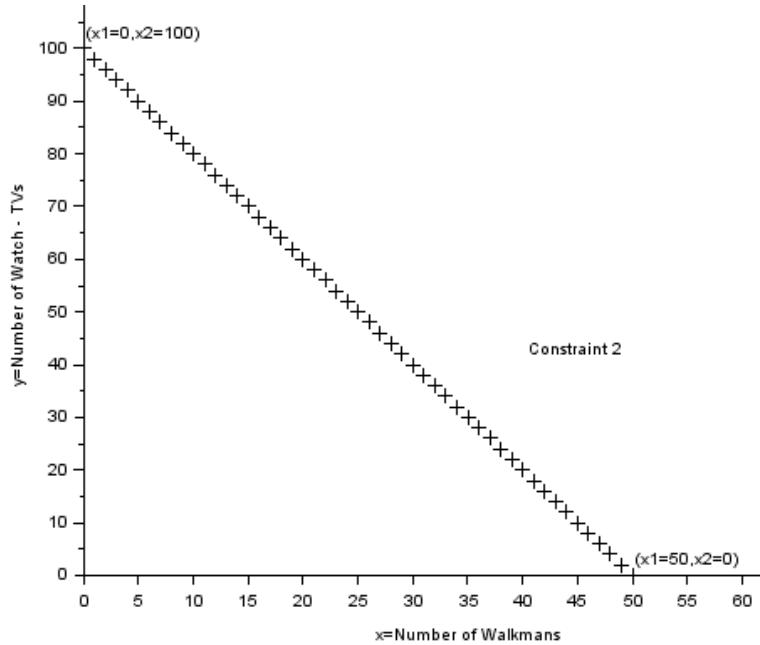
We see, however, that the isoprofit line for \$210 does not produce the highest possible profit to the firm. In Figure 5, we try graphing two more lines, each yielding a higher profit. The middle equation,  $\$280 = \$7x_1 + \$5x_2$ , was plotted in the same fashion as the lower line. When  $x_1 = 0$ ,

$$\$280 = \$7(0) + \$x_2$$

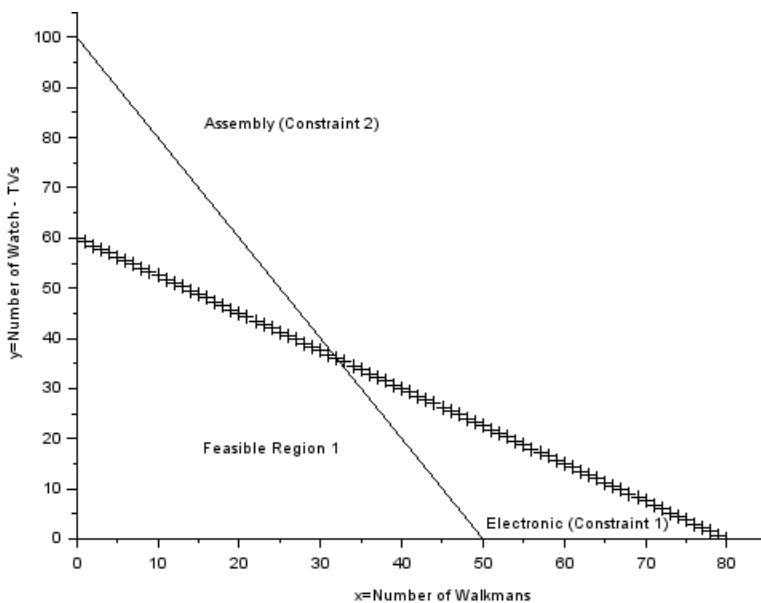
$$x_2 = 56 \text{ Watch-TVs}$$



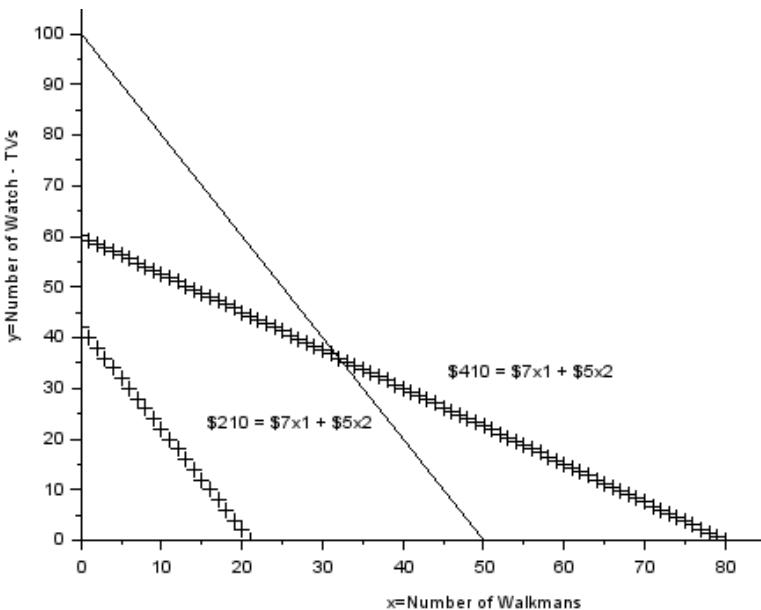
**FIGURE 1.** Constraint 1 (made with SCILAB)



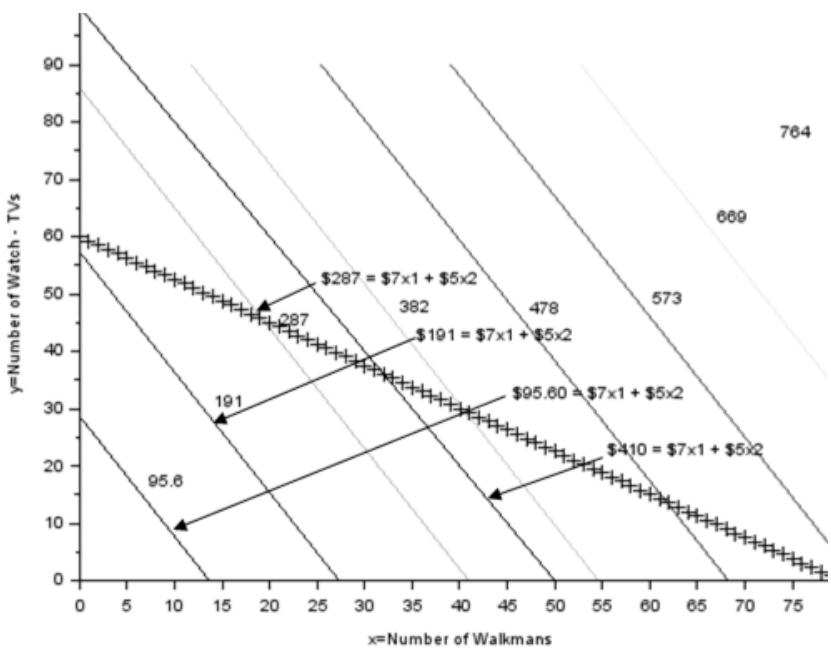
**FIGURE 2.** Constraint 2 (made with SCILAB)



**FIGURE 3.** Feasible Solution Region for the Radio Barn Problem (made with *SCILAB*)



**FIGURE 4.** A Profit Line of \$210 Plotted for the Radio Barn Problem (made with *SCILAB*)



**FIGURE 5.** Four Iso-Profit Lines Plotted for the Radio Barn (made with SCILAB)

When  $x_2 = 0$ ,

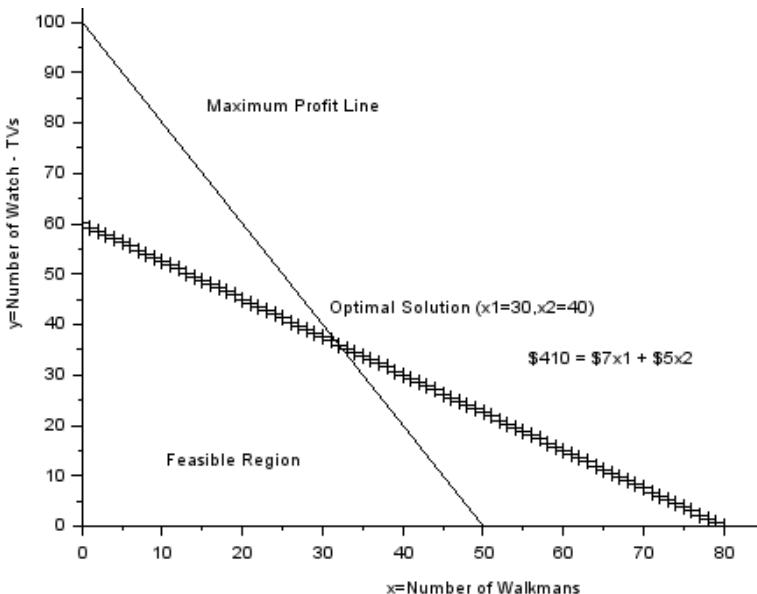
$$\$280 = \$7x_1 + \$5(0)$$

$$x_1 = 40 \text{ Walkmans}$$

Again, any combination of Walkmans ( $x_1$ ) and Watch-TVs ( $x_2$ ) on this iso-profit line will produce a total profit of \$280.

Note that the third line generates a profit of \$350, even more of an improvement. The farther we move from the 0 origin, the higher our profit will be. Another important point to note is that these iso-profit lines are parallel. We now have two clues as to how to find the optimal solution to the original problem. We can draw a series of parallel profit lines (by carefully moving our ruler in a plane parallel to the first profit line). The highest profit line that still touches some point of the feasible region will pinpoint the optimal solution. Notice that the fourth line (\$420) is too high to count because it does not touch the feasible region.

The highest possible iso-profit line is illustrated in Figure 6. It touches the tip of the feasible region at the corner point ( $x_1 = 30$ ,  $x_2 = 40$ ) and yields a profit of \$410.



**FIGURE 6.** Optimal Solution for the Radio Barn Problem (made with SCILAB)

#### 5.10.1.1 Corner-Point Solution Method

A second approach to solving linear programming problems employs the corner-point method. This technique is simpler in concept than the iso-profit line approach, but it involves looking at the profit at every corner point of the feasible region.

The mathematical theory behind linear programming states that an optimal solution to any problem (that is, the values of  $x_1$ ,  $x_2$  that yield the maximum profit) will lie at a *corner point*, or *extreme point*, of the feasible region. Hence, it is necessary to find only the values of the variables at each corner; the maximum profit or optimal solution will lie at one (or more) of them.

Once again we can see (in Figure 7) that the feasible region for the Radio Barn problem is a four-sided polygon with four corner, or extreme, points. These points are labeled ①, ②, ③ and ④ on the graph. To find the  $(x_1, x_2)$  values producing the maximum profit, we find out what the

coordinates of each corner point are, then determine and compare their profit levels.

Point ①:  $(x_1 = 0, x_2 = 0)$  Profit  $\$7(0) + \$5(0) = \$0$

Point ②:  $(x_1 = 0, x_2 = 80)$  Profit  $\$7(0) + \$5(80) = \$400$

Point ③:  $(x_1 = 50, x_2 = 0)$  Profit  $\$7(50) + \$5(0) = \$350$

We skipped corner point ③ momentarily because to find its coordinates *accurately*, we will have to solve for the intersection of the two constraint lines. As you may recall from algebra, we can apply the method of *simultaneous equations* to the two constraint equations:

$$4x_1 + 3x_2 = 240 \text{ (electronics time)}$$

$$2x_1 + 1x_2 = 100 \text{ (assembly time)}$$

To solve these equations simultaneously, we multiply the second equation by  $-2$ :

$$-2(2x_1 + 1x_2 = 100) = -4x_1 - 2x_2 = -200$$

and then add it to the first equation:

$$\begin{array}{r} +4x_1 + 3x_2 = 240 \\ -4x_1 - 2x_2 = -200 \\ \hline +1x_2 = 40 \end{array}$$

or

$$x_2 = 40$$

Doing this has enabled us to eliminate one variable,  $x_1$ , and to solve for  $x_2$ . We can now substitute 40 for  $x_2$  in either of the original equations and solve for  $x_1$ . Let us use the first equation. When  $x_2 = 40$ , then

$$4x_1 + 3(40) = 240$$

$$4x_1 + 120 = 240$$

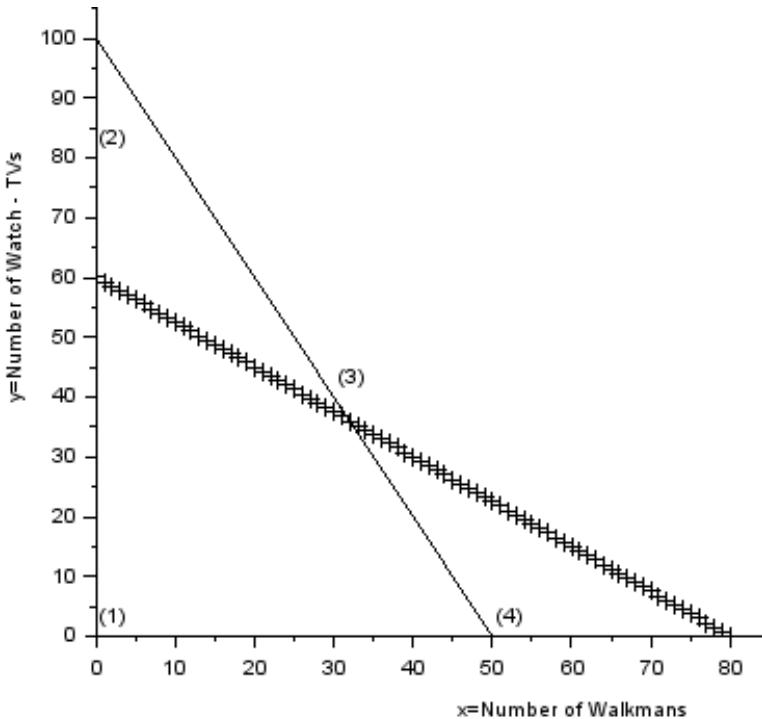
or

$$4x_1 = 120$$

$$x_1 = 30$$

Thus, point ③ has the coordinates ( $x_1 = 30$ ,  $x_2 = 40$ ). We can compute its profit level to complete the analysis:

Point ③: ( $x_1 = 30$ ,  $x_2 = 40$ ) Profit = \$7(30) + \$5(40) = \$410



**FIGURE 7.** The Four Corner Points of the Feasible Region (made with SCILAB)

Because point ③ produces the highest profit of any corner point, the product mix of  $x_1 = 30$  Walkmans and  $x_2 = 40$  Watch-TVs is the optimal solution to the Radio Barn problem. This solution will yield a profit of \$410 per production period; it is the same solution we obtained using the iso-profit line method.

## 5.11. Introduction to Sensitivity Analysis

Operations managers are usually interested in more than the optimal solution to an LP problem. In addition to knowing the value of each decision variable (the  $x_i$ 's) and the value of the objective function, they want to know how sensitive these answers are to input parameter changes. For example, what happens if the coefficients of the objective

function are not exact, or if they change by 10% or 15%? What happens if right-hand-side values of the constraints change? Because solutions are based on the assumption that input parameters are constant, the subject of sensitivity analysis comes into play. Sensitivity analysis, or postoptimality analysis, is the study of how sensitive solutions are to parameter changes.

There are two approaches to determining just how sensitive an optimal solution is to changes. The first is simply a trial-and-error approach. This approach usually involves resolving the entire problem, preferably by computer, each time one input data item or parameter is changed. It can take a long time to test a series of possible changes in this way.

The approach we prefer is the analytic postoptimality method. After an LP problem has been solved, we determine a range of changes in problem parameters that will not affect the optimal solution or change the variables in the solution. This is done without resolving the whole problem. LP software, such as Excel's Solver or POM for Windows, has this capability. Let us examine several scenarios relating to the Radio Barn example.

Figure 12 is part of the Excel Solver computer-generated output available to help a decision maker know whether a solution is relatively insensitive to reasonable changes in one or more of the parameters of the problem. (The complete computer run for these data, including input and full output, is illustrated in Figures 9 through 12 later in this chapter.)

### 5.11.1. Sensitivity Report

The Excel *Sensitivity Report* for the Radio Barn example in Figure 8 has two distinct components: (1) a table titled Adjustable Cells and (2) a table titled Constraints. These tables permit us to answer several what-if questions regarding the problem solution.

It is important to note that while using the information in the sensitivity report to answer what-if questions, we assume that we are considering a change to only a *single* input data value. That is, the sensitivity information does not always apply to simultaneous changes in several input data values.

The *Adjustable Cells* table presents information regarding the impact of changes to the objective function coefficients (i.e., the unit profits of \$7 and \$5) on the optimal solution. The *Constraints* table presents information related to the impact of changes in constraint right-hand-side (RHS) values (i.e., the 240 hours and 100 hours) on the optimal solution. Although different LP software packages may format and present these tables differently, the programs all provide essentially the same information.

All LP problems can also be solved with the simplex method, using software such as POM for Windows or Excel. This approach produces valuable economic information such as the shadow price, or dual, and provides complete sensitivity analysis on other inputs to the problems. Excel uses Solver, which requires that you enter your own constraints. Excel OM does not have an LP module. POM for Windows requires only that demand data, supply data, and shipping costs be entered. In the following section we illustrate how to create an Excel spreadsheet for LP problems.

## 5.12. Using Excel Spreadsheets

Excel offers the ability to analyze linear programming problems using built-in problem-solving tools. Excel's tool is named Solver. Solver is limited to 200 changing cells (variables), each with 2 boundary constraints and up to 100 additional constraints. These capabilities make Solver suitable for the solution of complex, real-world problems.

We use Excel to set up the Radio Barn problem in Program 2. The objective and constraints are repeated here:

Objective function: Maximize profit =

$$\$7(\text{No. of Walkmans}) + \$5(\text{No. of Watch-TVs})$$

Subject to:

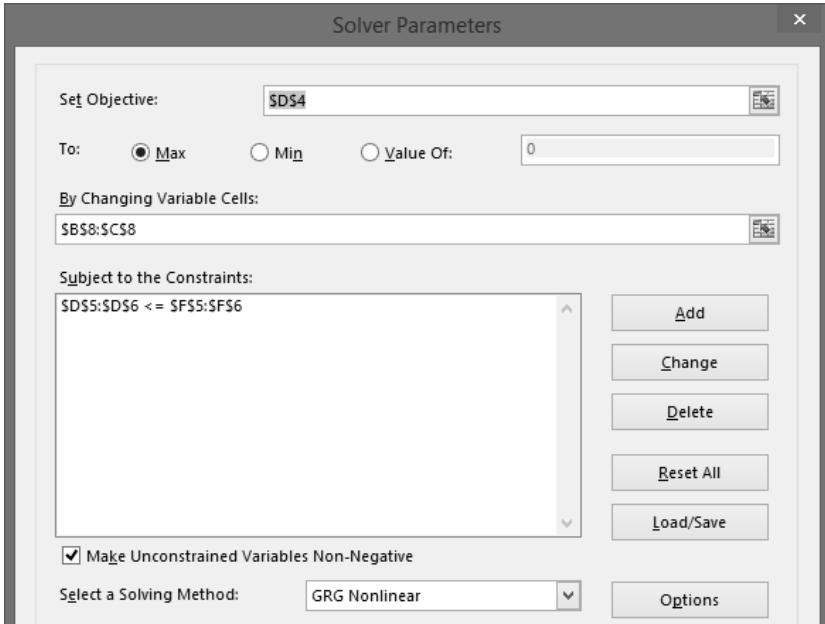
$$\begin{aligned}4(\text{Walkmans}) + 3(\text{Watch-TVs}) &\leq 240 \\2(\text{Walkmans}) + 1(\text{Watch-TV}) &\leq 100\end{aligned}$$

|   | A                         | B        | C         | D   | E    | F     | G   | H |
|---|---------------------------|----------|-----------|-----|------|-------|-----|---|
| 1 | <b>Shader Electronics</b> |          |           |     |      |       |     |   |
| 2 |                           |          |           |     |      |       |     |   |
| 3 |                           | Walkmans | Watch-TVs | LHS | RHS  | Slack |     |   |
| 4 | Objective Function        |          | 7         | 5   | 0    |       |     |   |
| 5 | Electronics               |          | 4         | 3   | 0 <= | 240   | 240 |   |
| 6 | Assembly                  |          | 2         | 1   | 0 <= | 100   | 100 |   |
| 7 |                           |          |           |     |      |       |     |   |
| 8 | Solution Values           |          | 0         | 0   |      |       |     |   |

**FIGURE 8.** Using Excel to Formulate the Radio Barn Problem

|   | A                         | B        | C         | D              | E   | F      | G |
|---|---------------------------|----------|-----------|----------------|-----|--------|---|
| 1 | <b>Shader Electronics</b> |          |           |                |     |        |   |
| 2 |                           |          |           |                |     |        |   |
| 3 |                           | Walkmans | Watch-TVs | LHS            | RHS | Slack  |   |
| 4 | Objective Function        | 7        | 5         | =SUMPRODUCT    |     |        |   |
| 5 | Electronics               | 4        | 3         | =SUMPRODUCT <= | 240 | =F5-D5 |   |
| 6 | Assembly                  | 2        | 1         | =SUMPRODUCT <= | 100 | =F6-D6 |   |
| 7 |                           |          |           |                |     |        |   |
| 8 | Solution Values           | 0        | 0         |                |     |        |   |

**FIGURE 9.** Excel cell formulas shown



**FIGURE 10.** Excel Solver settings

|   | A                         | B        | C         | D   | E      | F   | G     | H |
|---|---------------------------|----------|-----------|-----|--------|-----|-------|---|
| 1 | <b>Shader Electronics</b> |          |           |     |        |     |       |   |
| 2 |                           |          |           |     |        |     |       |   |
| 3 |                           | Walkmans | Watch-TVs | LHS |        | RHS | Slack |   |
| 4 | Objective Function        |          | 7         | 5   | 410    |     |       |   |
| 5 | Electronics               |          | 4         | 3   | 240 <= | 240 | 0     |   |
| 6 | Assembly                  |          | 2         | 1   | 100 <= | 100 | 0     |   |
| 7 |                           |          |           |     |        |     |       |   |
| 8 | Solution Values           |          | 30        | 40  |        |     |       |   |

**FIGURE 11.** Excel Solution to Radio Barn LP Problem

**Microsoft Excel 15.0 Sensitivity Report**  
**Worksheet: [Shader]LP\_formulaton**  
**Report Created: 2/2/2015 10:26:34 PM**

**Strickland Jeffrey:**  
The solution values for the variables appear.  
We should make 30 Walkmans and 40 Watch-TVs

**Variable Cells**

| Cell   | Name                      | Final Value | Reduced Gradient | Objective Coefficient | Allowable Increase | Allowable Decrease |
|--------|---------------------------|-------------|------------------|-----------------------|--------------------|--------------------|
| \$B\$8 | Solution Values Walkmans  | 30          | 0                | 7                     | 0.3                | 0.3333333          |
| \$C\$8 | Solution Values Watch-TVs | 40          | 0                | 5                     | 0.25               | 1.5                |

**Constraints**

| Cell   | Name            | Final Value | Lagrange Gradient | Constraint RHS | Allowable Increase | Allowable Decrease |
|--------|-----------------|-------------|-------------------|----------------|--------------------|--------------------|
| \$D\$5 | Electronics LHS | 240         | 1.5               | 240            | 60                 | 40                 |
| \$D\$6 | Assembly LHS    | 100         | 0.5               | 100            | 20                 | 20                 |

**Strickland Jeffrey:**  
We will use 240 hours and 100 hours of Electronics and Assembly time, respectively

**Strickland Jeffrey:**  
If we use one more Electronics hour, our profit will increase by \$1.50. This is true for up to 60 more hours. The profit will fall by \$1.50 for each Electronics hour less than 240 hours, down as low as 200 hours.

**FIGURE 12.** Sensitivity Analysis for Radio Barn Using Excel's Solver



# 6. What is Sensitivity Analysis?

## 6.1. What and Why is Sensitivity Analysis

A significant problem with LPs (or with any other models, for that matter) is the assumption that the parameters  $c$ ,  $A$ , and  $b$  in

$$\max z = c'x$$

s.t.

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

are known without any margin of error. In practice we only have a (good or bad) guess about the true values of the parameters. Sensitivity analysis is a systematic study of how, well, sensitive, the solutions of the LP are to small changes in the data. The key questions are

1. If the objective function  $c$  changes in its parameter  $c_i$ , how does the solution change?
2. If the resources available change, i.e., the constraint vector  $b$  change in its parameter  $b_i$ , how does the solution change?

Question 1 is related to the concept of **reduced cost** and question 2 is related to the concept of **shadow price**.

The brute force approach to these questions is to solve lots and lots of LPs: One LP for each change in the parameters. For example, in Gepetto's problem (Example 2) there might be uncertainty in what is the actual market demand for soldiers. It was assumed to be 40, but it could be anything between, say, 30 and 50. We could then solve the Gepetto's LP separately for market demands 30, 31, ..., 49, 50. So, we would solve 20 different LPs (21, actually, but who's counting). If it is also assumed that the profit for soldiers might not be exactly \$3 but could be anything between \$2.5 and \$3.5, then we could also solve the LP separately for profits \$2.5, \$2.6, ..., \$3.4, \$3.5. Combining this with the different LPs we got from the uncertainty in the market demand we would then have  $20 \times 10 = 200$  different LPs to solve (well,  $21 \times 11 = 231$  if you count

correctly). This “checking the scenarios” method works, and it is indeed widely used in practice.

The brute force method explained above has at least three problems: (1) It is inelegant, (2) it would involve a large amount of calculations, and (3) it is hard to see what happens when more than two parameters change. These problems are, however, not critical. Indeed, (3) understanding high-dimensional spaces is always difficult, (2) solving hundreds of LPs is not that time-consuming with modern computers and efficient algorithms like the simplex, and (1) who cares about elegance these days? Nevertheless, we shall try to be at least a little bit elegant in this chapter.

**DEFINITION.** Shadow Prices: The shadow price  $\pi_i$  of a constraint  $b_i$  is the amount that the objective function’s value  $z$  at the optimum would change if the constraint  $b_i$  is changed by one unit — given that the optimal BV does not change.

Note the clause “given that the optimal BV does not change”. This means that the shadow price is valid for small changes in the constraints. If the optimal corner changes when a constraint is changed, then the interpretation of the shadow price is no longer valid.

Shadow prices are sometimes called **dual variables** or **marginal prices**. The name “marginal price” actually a much more informative name than the nebulous shadow price (or dual variable). Indeed, suppose you have a constraint that limits, say, the amount of labor available to 40 hours per week. Then the shadow price will tell you how much you would be willing to pay for an additional hour of labor. If your shadow price is \$10 for the labor constraint, for instance, you should pay no more than \$10 an hour for additional labor. Labor costs of less than \$10 per hour will increase the objective value; labor costs of more than \$10 per hour will decrease the objective value. Labor costs of exactly \$10 will cause the objective function value to remain the same.

If you like mathematical formulas — and even if you don’t — the shadow prices can be defined as follows: Consider the LP

$$\max z = c'x$$

s.t.

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

The optimal solution  $z^*$  of this LP is a function of the objective vector  $\mathbf{c}$ , the technology matrix  $A$ , and the constraint vector  $\mathbf{b}$ :

$$z^* = z^*(\mathbf{c}; A; \mathbf{b})$$

Then the shadow price  $\pi_i$  associated to the constraint  $b_i$  is the partial derivative

$$\pi_i = \frac{\partial z^*}{\partial b_i},$$

or, in vector form,

$$\pi = \frac{\partial z^*}{\partial \mathbf{b}},$$

Where

$$\pi = \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_m \end{bmatrix} \text{ and } \frac{\partial z^*}{\partial \mathbf{b}} = \begin{bmatrix} \frac{\partial z^*}{\partial b_1} \\ \vdots \\ \frac{\partial z^*}{\partial b_m} \end{bmatrix}$$

Suppose now that  $\epsilon = [\epsilon_1 \quad \cdots \quad \epsilon_m]'$  is a small vector, and

$$z_\epsilon^* = z^*(\mathbf{c}, A, \mathbf{b} + \epsilon)$$

is the optimal value, when the constraints  $\mathbf{b}$  are changed by  $\epsilon$ . Then the first order approximation for the new optimal value is

$$\begin{aligned} z_\epsilon^* &= z^* + \epsilon' \frac{\partial z^*}{\partial \mathbf{b}} \\ &= z^* + \epsilon' \pi \end{aligned}$$

(6.1)

The equality (6.1) is valid as long as the elements  $\epsilon_i$  of  $\epsilon$  are small enough in absolute value. If some of the elements of  $\epsilon$  are too big, then the equality (6.1) may fail to be true.

Let us see now how to use formula (6.1) in sensitivity analysis.

**EXAMPLE 1.** Consider the LP

$$\max z = 4x_1 + 3x_2$$

s.t.

$$2x_1 + 3x_2 \leq 6 \quad (\text{Constraint 1})$$

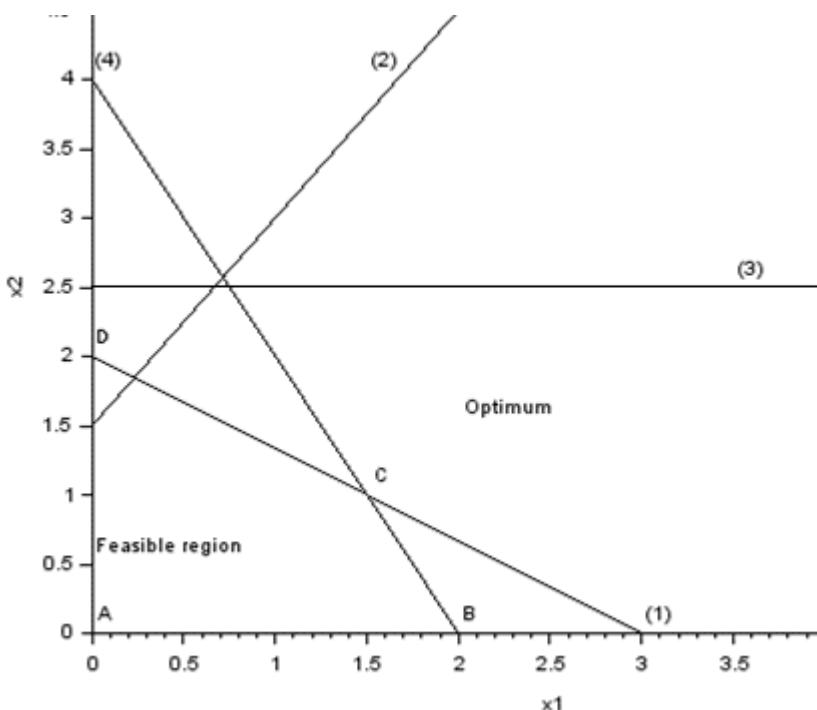
$$-3x_1 + 2x_2 \leq 3 \quad (\text{Constraint 2})$$

$$2x_2 \leq 5 \quad (\text{Constraint 3})$$

$$2x_1 + x_2 \leq 4 \quad (\text{Constraint 4})$$

$$x_1, x_2 \geq 0$$

A picture representing the LP is shown in Figure 1:



**Figure 1.** Example 1 LP (made with SCILAB)

From the picture we read — by moving the isoprofit line (the dashed lines) away from the origin — that the optimal decision is at the point  $C = (1.5, 1)$ . Therefore, the optimal value is  $z = 4 \times 1.5 + 3 \times 1 = 9$ . We also see that the constraints (1) and (4) are active at the optimum. So, changing them should change the optimal value. Indeed, they should have positive shadow prices. In contrast, the constraints (2) and (3) are not active at the optimum. So, changing them — slightly — should have no effect on the optimum. So, both of them should have 0 as their shadow price.

Let us calculate the shadow prices of Example 1 with Excel.

|    | A                  | B  | C   | D   | E       | F   | G         |
|----|--------------------|----|-----|-----|---------|-----|-----------|
| 1  | Example 1          |    |     |     |         |     |           |
| 2  |                    |    |     |     |         |     |           |
| 3  |                    | x1 | x2  | LHS |         | RHS | Slack     |
| 4  | Objective Function |    | 4   | 3   | 9       |     |           |
| 5  | Constraint 1       |    | 2   | 3   | 6 <=    | 6   | -7.46E-14 |
| 6  | Constraint 2       |    | -3  | 2   | -2.5 <= | 3   | 5.5       |
| 7  | Constraint 3       |    | 0   | 2   | 2 <=    | 5   | 3         |
| 8  | Constraint 4       |    | 2   | 1   | 4 <=    | 4   | -7.46E-14 |
| 9  |                    |    |     |     |         |     |           |
| 10 | Solution Values    |    | 1.5 | 1   |         |     |           |

Microsoft Excel 15.0 Answer Report

Worksheet: [LP.xlsx]Sheet1

Report Created: 2/24/2015 8:46:18 AM

Result: Solver found a solution. All Constraints and optimality conditions are satisfied.

Solver Engine

Engine: GRG Nonlinear

Solution Time: 0.062 Seconds.

Iterations: 3 Subproblems: 0

Solver Options

Max Time Unlimited, Iterations Unlimited, Precision 0.000001  
 Convergence 0.0001, Population Size 100, Random Seed 0,  
 Derivatives Central  
 Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer  
 Tolerance 1%, Assume Nonnegative

#### Objective Cell (Max)

| Cell               | Name | Original Value | Final Value |
|--------------------|------|----------------|-------------|
| Objective Function |      |                |             |
| \$D\$4             | LHS  | 7              | 9           |

#### Variable Cells

| Cell    | Name               | Original Value | Final Value | Integer |
|---------|--------------------|----------------|-------------|---------|
| \$B\$10 | Solution Values x1 | 1              | 1.5         | Contin  |
| \$C\$10 | Solution Values x2 | 1              | 1           | Contin  |

#### Constraints

| Cell   | Name             | Cell Value | Formula        | Status      | Slack |
|--------|------------------|------------|----------------|-------------|-------|
| \$D\$5 | Constraint 1 LHS | 6          | \$D\$5<=\$F\$5 | Binding     | 0     |
| \$D\$6 | Constraint 2 LHS | -2.5       | \$D\$6<=\$F\$6 | Not Binding | 5.5   |
| \$D\$7 | Constraint 3 LHS | 2          | \$D\$7<=\$F\$7 | Not Binding | 3     |
| \$D\$8 | Constraint 4 LHS | 4          | \$D\$8<=\$F\$8 | Binding     | 0     |

#### Sensitivity Report

##### Variable Cells

| Cell    | Name               | Final | Reduced  |
|---------|--------------------|-------|----------|
|         |                    | Value | Gradient |
| \$B\$10 | Solution Values x1 | 1.5   | 0        |
| \$C\$10 | Solution Values x2 | 1     | 0        |

##### Constraints

| Cell   | Name             | Final Value | Lagrange Multiplier |
|--------|------------------|-------------|---------------------|
| \$D\$5 | Constraint 1 LHS | 6           | 0.5                 |
| \$D\$6 | Constraint 2 LHS | -2.5        | 0                   |
| \$D\$7 | Constraint 3 LHS | 2           | 0                   |
| \$D\$8 | Constraint 4 LHS | 4           | 1.5                 |

From the Excel reports, we see that  $x^*=[1.5, 5]$  and  $f^*(x)=9$  is the optimum solution. The marginal values (shadow values) are 0.5 and 1.5 for constraints (1) and (4), respectively.

Using the information from the Excel answer report—that constraints (2) and (3) are not binding, we can write  $A=[2,3;2,1]$  and  $b=[6,4]$ , to obtain the solution with *SCILAB*.

```

A=[2,3;2,1; b=[6,4];
[xopt,lagr,fopt]=linpro(c,A,b,ci,cs,2)
fopt =
9
lagr =
0.
0.
-0.5
-1.5
xopt =
1.5
1.

```

So, the marginal (shadow) prices for the constraints (1) and (4) are 0.5 and 1.5 , respectively. All other shadow prices are 0 . So, the shadow price vector is

$$\pi = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 1.5 \end{bmatrix}$$

Let us then try to use formula (6.1) to see what happens if the constraints (1) – (4) change. Suppose each constraint is relaxed by 0.5. That means that in formula (6.1) we have  $c = [0.5 \ 0.5 \ 0.5 \ 0.5]'$ . So, the new optimum should be

$$\begin{aligned} z_\epsilon^* &= z^* + \epsilon' \pi \\ &= 9 + 0.5 \times 0.5 + 0.5 \times 0 + 0.5 \times 0 + 0.5 \times 1.5 \\ &= 10 \end{aligned}$$

Is this correct? Let us see:

```
bb=b+0.5;
[xopt,lagr,fopt]=linpro(c,A,bb,ci,cs,2)
fopt =
10.
lagr =
0.
0.
- 0.5
- 1.5
xopt =
1.75
1.
```

So, we see that formula (6.1) is correct, when  $E \leq [0.5 \ 0.5 \ 0.5 \ 0.5]'$ .

Let us then consider a big change. Let  $E = [10 \ 10 \ 10 \ 0]'$ . Then formula (6.1) would give us

$$\begin{aligned} z_\epsilon^* &= z^* + \epsilon' \pi \\ &= 9 + 10 \times 0.5 + 10 \times 0 + 10 \times 0 + 10 \times 1.5 \\ &= 14 \end{aligned}$$

Let's see what really happens:

```
c=[4;3]; A=[2 3;-3 2;0 2;2 1];
b=[6;3;5;4]+[10;10;10;0];
[xopt,lagr,fopt]=linpro(c,A,b,ci,cs)
fopt =
12.
lagr =
```

```

0.
0.
0.
3.
-2.
0.
xopt  =
0.
4.

```

We see that `linpro` gives 12 as the optimum. The formula (6.1) gave us the optimum 14. So, the change  $[10 \ 10 \ 10 \ 0]'$  is not small enough for the formula (6.1) to be valid. What actually happened here was that the optimal point jumped corners.

## 6.2. Reduced Costs

Let us then consider the reduced costs. Remember that the shadow prices were associated to the constraints, or — if you like simplex language — to the slacks. The reduced costs are associated to the decision variables.

The reduced cost  $u_i$  for a (NBV) decision variable  $x_i$  is the amount the objective value  $z$  at the optimum would decrease if  $x_i$  would be forced to be 1, and thus a BV — given that the change from  $x_i = 0$  to  $x_i = 1$  is small.

Here are some interpretations and remarks of reduced costs that should help you to understand the concept:

- The clause “given that the change from  $x_i = 0$  to  $x_i = 1$  is small” is a similar clause that the clause “given that the optimal BVs don’t change” was in the definition of the shadow price. Indeed, it may be, e.g., that forcing  $x_i \geq 1$  will make the LP infeasible. Remember: In sensitivity analysis we are talking about **small changes** — whatever that means. The analysis may, and most often will, fail for big changes.
- Decision variables that are BV have zero reduced costs.

- The reduced cost is also known as **opportunity cost**. Indeed, suppose we are given the **forced** opportunity (there are no problems — only opportunities) to produce one unit of  $x_i$  that we would not otherwise manufacture at all. This opportunity would cost us, since our optimized objective would decrease to a suboptimal value. Indeed, we have now one more constraint — the forced opportunity — in our optimization problem. So, the optimal solution can only get worse. The decrease of the objective value is the opportunity cost.
- The reduced cost  $u_i$  of  $x_i$  is the amount by which the objective coefficient  $c_i$  for  $x_i$  needs to change before  $x_i$  will become non-zero at the optimum.
- As an example of the point above, consider that you are producing  $x_1, \dots, x_n$  that will give you profits  $c_1, \dots, c_n$ . You have some constraints, but the actual form of them does not matter here. Now, you form the LP to optimize your profit, and you solve it. You get optimal solution for productions:  $x_1^*, x_2^*, \dots, x_n^*$ , and you get your optimal profit  $z^*$ . You notice that, say,  $x_2^* = 0$ . So, obviously the profit  $c_2$  for  $x_2$  is not big enough. Then you ask yourself: How big should the profit  $c_2$  for  $x_2$  be so that it becomes more profitable to produce  $x_2$ , at least a little, rather than not to produce  $x_2$  at all? The answer is  $c_2 - u_2$  (or  $c_2 + u_2$ , depending on how you interpret the sign). This means that the profit must increase at least by the reduced cost before it becomes more profitable to produce a product you would not produce otherwise.

Let us now consider the reduced cost with glpk with:

**EXAMPLE 2.**

$$\max z = 4x_1 + 3x_2$$

s.t.

$$2x_1 + 3x_2 \leq 16 \quad (1)$$

$$-3x_1 + 2x_2 \leq 13 \quad (2)$$

$$2x_2 \leq 5 \quad (3)$$

$$2x_1 + x_2 \leq 4 \quad (4)$$

$$x_1; x_2 \geq 0$$

glpk returns the reduced costs in the output structure extra in its field redcosts:

```
c=[4 3]'; A=[2 3; -3 2; 0 2; 2 1]; b=[16 13 15 4]';
[xopt,fopt,exitflag,iter,yopt]=karmarkar([],[],c,x0,[],[],[],[],A,b,lb,ub)
Rank deficient. rank = 5

yopt =
ineqlin: [4x1 constant]
eqlin: [0x0 constant]
lower: [2x1 constant]
upper: [2x1 constant]
iter =
1.
exitflag =
1.
fopt =
12.
xopt =
0.
4.

[xopt,lagr,fopt]=linpro(c,A,b,lb,ub,1,x0)
```

So, we see that the redcosts are  $u_1 = -2$  for the NBV decision  $x_1$  and  $u_2 = 0$  for the BV decision  $x_2$ .

Let us then test the interpretation:

“reduced cost is the decrease in the value of the objective if we are forced to produce one unit where we otherwise would produce none”.

We test the interpretation with the following LP:

$$\begin{aligned}
 \max z &= 4x_1 + 3x_2 \\
 2x_1 + 3x_2 &\leq 16 & (1) \\
 -3x_1 + 2x_2 &\leq 13 & (2) \\
 2x_2 &\leq 15 & (3) \\
 2x_1 + x_2 &\leq 4 & (4) \\
 x_1 &\geq 1 & (5) \\
 x_1, x_2 &\geq 0
 \end{aligned}$$

So, we have added to the LP of Example 2 the requirement that we must have at least one  $x_1$  in the solution. This is the constraint (5). Remember that without this requirement we would have zero  $x_1$ 's in the solution.

So, here is the *SCILAB* code for this problem:

```

c=[4 3]'; A=[2 3; -3 2; 0 2; 2 1; 0 1]; b=[16 13 15 4 1]';
[xopt,lagr,fopt]=linpro(c,A,b,lb,ub,1)
fopt =
10.
lagr =
0.
0.
- 3.
2.
xopt =
1.
2.

```

We see that the interpretation is indeed correct: The previous optimal value 12 dropped by 2 into 10.

### 6.2.1. Problems

- Carco manufactures cars and trucks. Each car contributes \$300 to profit and each truck, \$400. The resources required to manufacture a car and a truck are shown in Table 1. Each day, Carco can rent up to 98 Type 1 machines at a cost of \$50 per machine. The company now has 73 Type 2 machines and 260 tons of steel available. Marketing considerations dictate that at least 88 cars and at least 26 trucks be produced. Let

$x_1$  = number of cars produced daily

$x_2$  = number of trucks produced daily

$M_1$  = type 1 machines rented daily

**TABLE 1**

| Vehicle | Days on Type Machine | Days on Type Machine | 2Tons of Steel |
|---------|----------------------|----------------------|----------------|
| Car     | 0.8                  | 0.6                  | 2              |
| Truck   | 1.8                  | 0.7                  | 3              |

To maximize profit, Carco should solve the LP given in Figure 11. Use the Figure 11, LPSolve or Excel output to answer the following questions:

- a. If cars contributed \$310 to profit, what would be the new optimal solution to the problem?
- b. What is the most that Carco should be willing to pay to rent an additional Type 1 machine for 1 day?
- c. What is the most that Carco should be willing to pay for an extra ton of steel?
- d. If Carco were required to produce at least 86 cars, what would Carco's profit become?
- e. Carco is considering producing jeeps. A jeep contributes \$600 to profit and requires 1.2 days on machine 1, 2 days on machine 2, and 4 tons of steel. Should Carco produce any jeeps?

|            |                       |
|------------|-----------------------|
| MAX        | 300 X1 + 400 X2 - 50  |
|            | M1                    |
| SUBJECT TO |                       |
| 2)         | 0.8 X1 + X2 - M1 <= 0 |
| 3)         | M1 <= 98              |
| 4)         | 0.6 X1 + 0.7 X2 <= 73 |
| 5)         | 2 X1 + 3 X2 <= 260    |
| 6)         | X1 >= 88              |
| 7)         | X2 >= 26              |

**FIGURE 11.** Computer Output for Carco (Problem 1)

2. The following LP has the optimal tableau shown in Table 2.

**Table 2**

| <b><math>z</math></b> | <b><math>x_1</math></b> | <b><math>x_2</math></b> | <b><math>e_1</math></b> | <b><math>e_2</math></b> | <b><math>a_1</math></b> | <b><math>a_2</math></b> | <b><math>a_3</math></b> | <b><math>rhs</math></b> |
|-----------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| 1                     | 0                       | 3                       | 0                       | 0                       | M                       | M                       | $M + 4$                 | 12                      |
| 0                     | 1                       | 1                       | 0                       | 0                       | -0                      | -0                      | 1                       | 13                      |
| 0                     | 0                       | 2                       | 1                       | 0                       | -1                      | -0                      | 3                       | 13                      |
| 0                     | 0                       | 1                       | 0                       | 1                       | -0                      | -1                      | 2                       | 12                      |

- a. Find the dual of this LP and its optimal solution.
- b. Find the range of values of the objective function coefficient of  $x_2$  for which the current basis remains optimal.
- c. Find the range of values of the objective function coefficient of  $x_1$  for which the current basis remains optimal.

## 6.3. Dual Problem

### 6.3.1. Finding Dual

Associated with any LP there is another LP, called the **dual** — and then the original LP is called the **primal**. In general, if the primal LP is a maximization problem, the dual is a minimization problem — and vice versa. Also, the constraints of the primal LP are the coefficients of the objective of the dual problem — and vice versa. If the constraints of the primal LP are of type  $\leq$  then the constraints of the dual LP are of type  $\geq$  — and vice versa.

Let us now give the formal definition of the dual. We assume that the primal LP is in standard form. Since all LPs can be transformed into a standard form this assumption does not restrict the generality of the duality. The assumption is made only for the sake of convenience.

The (linear) **dual** of the LP

$$\max z = c'x$$

s.t.

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

is

$$\min w = b'y$$

s.t.

$$\begin{aligned} A'y &\geq c \\ y &\geq 0 \end{aligned}$$

**EXAMPLE 3.** Consider the LP

$$\max z = [1 \quad 2 \quad 3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

s.t.

$$\begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \leq \begin{bmatrix} 10 \\ 11 \end{bmatrix}$$

The dual LP is

$$\min w = [10 \quad 11] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

s.t.

$$\begin{bmatrix} 4 & 5 \\ 6 & 7 \\ 8 & 9 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Let us discuss briefly concept of duality in general and the linear duality in particular.

- In general, dual is a transformation with the following property: **transforming twice you get back**. This is the abstract definition of duality. In mathematics a function  $f$  is called **involution** if it is its own inverse, i.e.,  $f^{-1} = f$ . So, duality is a **meta-mathematical involution**.

- Looking at the definition of the linear dual one sees the dual is LP itself. So, it can be transformed into a standard form, and the one can construct the dual of the dual. When one does so one gets back to the original primal LP, i.e., **the dual of the dual is primal**. So, the linear dual deserves its name.
- We have already seen one duality between LPs before: A minimization problem is in duality with a maximization problem with the transform where the objective function is multiplied by  $-1$ . The usefulness of this simple duality was that we only need to consider maximization problems, and the solution of the minimization problem is  $-1$  times the solution of the corresponding maximization problem. Also, the optimal decisions in the maximization and minimization problems are the same.
- The linear duality is more complicated than the simple “multiply by  $-1$  duality” of the previous point. This makes the linear duality in some sense more useful. Indeed, since the transformation is more complicated, our change of perspective is more radical, and thus this transformation gives us better intuition of the original problem.
- The linear duality is very useful because of the following theorems: The weak duality theorem states that the objective function value  $w$  of the dual at any feasible solution  $y$  is always greater than or equal to the objective function value  $z$  of the primal at any feasible solution  $x$ :

$$w = b'y \geq c'x = z:$$

The weak duality theorem can be used to get upper bounds to the primal LP. The strong duality theorem states that if the primal has an optimal solution,  $x^*$ , then the dual also has an optimal solution,  $y^*$ , such that

$$Z^* = c'x^* = b'y = w^*$$

The strong duality theorem can be used to solve the primal LP. Finally, the complementary slackness theorem states that if a constraint in either the primal or the dual is non-active, then the corresponding variable in the other — complementary — problem must be zero.

Let us find a dual of an LP that is not in standard form.

**EXAMPLE 4.** Consider the LP

$$\min -z = 50x_1 + 20x_2 + 30x_3$$

s.t.

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 &\geq 11 \\ 12x_1 + 13x_2 + 14x_3 &\leq 111 \\ x_1 + x_2 + x_3 &\leq 1 \\ x_1 + x_2 + x_3 &= 1 \\ X_1, x_2, x_3 &\geq 0 \end{aligned}$$

The LP of Example 5.2.2 is not in standard form. So, before constructing its dual, we transform it into standard form. This is not necessary. Sometimes we can be clever, and find the dual without first transforming the primal into standard form. But we don't feel clever now. So, here is the standard form:

$$\max -z = -50x_1 - 20x_2 - 30x_3$$

s.t.

$$\begin{aligned} -2x_1 - 3x_2 - 4x_3 &\leq -11 \\ 12x_1 + 13x_2 + 14x_3 &\leq 111 \\ x_1 + x_2 + x_3 &\leq 1 \\ -x_1 - x_2 - x_3 &\leq -1 \\ X_1, x_2, x_3 &\geq 0 \end{aligned}$$

Now we are ready to present the dual:

$$\min -w = -11y_1 + 111y_2 + y_3 - y_4$$

s.t.

$$\begin{aligned} -2y_1 + 12y_2 + y_3 - y_4 &\geq -50 \\ -3y_1 + 13y_2 + y_3 - y_4 &\geq -20 \\ -4y_1 + 14y_2 + y_3 - y_4 &\geq -30 \\ Y_1, y_2, y_3, y_4 &\geq 0 \end{aligned}$$

(6.2)

(we used variable  $-w$  in the dual because there was variable  $-z$  in the standard form primal). Note now that the dual LP (5.2.3) is in “dual standard form”: It is a minimization problem with only inequalities of type  $\geq$ . The original primal LP was a minimization problem. So, it is natural to express the dual LP as a maximization problem. Also, inequalities of type  $\leq$  are more natural to maximization problems than the opposite type inequalities  $\geq$ . So, let us transform the dual LP (5.2.3) into a maximization problem with  $\leq$  type inequalities. In fact, let us transform the dual LP (5.2.3) into a standard form. We obtain

$$\max x = 11y_1 - 11y_2 - y_3 + y_4$$

s.t.

$$\begin{aligned} 2y_1 - 12y_2 - y_3 + y_4 &\leq 50 \\ 3y_1 - 13y_2 - y_3 + y_4 &\leq 20 \\ 4y_1 - 14y_2 - y_3 + y_4 &\leq 30 \\ y_1, y_2, y_3, y_4 &\geq 0 \end{aligned}$$

### EXAMPLE 5. The Dual Problem using SCILAB

Consider an LP with the following matrix format.

```
c = [ 1; -2; -3; -1 ]
a = [      1, -1, -2, -1;
       2,   0,   1, -4;
      -2,   1,   0,   1]
b = [ 4; 2; 1 ]
```

The SCILAB program which solves this LP is

```
# matrix notation of problem
c = [1; -2; -3; -1]
a = [1, -1, -2, -1; 2, 0, 1, -4; -2, 1, 0, 1]
b = [4; 2; 1]
[xopt,lagr,fopt]=linpro(c,Aeq,beq,lb,ub,2, x0)
fopt =
4.
lagr =
0.5
0.
- 1.25
0.
- 2.
```

```

0.25
0.
xopt =
7.
0.
0.
3.

```

Optimal value is 4 (fopt)

Optimal solution is [ 7; 0; 0; 3 ] (xopt)

We now want to construct and solve the dual problem. The  $4 \times 3$  matrix  $A$ , will become a  $3 \times 4$  matrix  $A'$ , and we will swap the vectors  $\mathbf{b}$  and  $\mathbf{c}$ . This gives us the matrix form of the dual problem:

```

b = [ 1; -2; -3; -1 ]
a = [
      1,   2, -2;
      -1,  0,  1;
      -2,  1,  0;
      -1. -4,  1]
c = [ 4; 2; 1 ]
The original problem was named Example_1, so we save and
name the dual problem as Example_1_dual. Here is the Octave
code.
# matrix notation of problem
b = [ 1; -2; -3; -1 ]
a = [
      1,   2, -2;
      -1,  0,  1;
      -2,  1,  0;
      -1. -4,  1]
c = [ 4; 2; 1 ]

#empty vector because the default lower bound on the
variables is zero
lb = [0;0;0];

#empty vector because there is no upper bound on the
variables
ub = [];

#indicates that each variable is continuous type
vartype = "CCCC";

#indicates the sense of each constraint as having an lower
bound
ctype = "LLLL";

```

```

#indicates that it is a minimization problem
s = 1;

# the command below displays:
# xopt = the optimal variables
# fopt = the optimal value
# status = 180 if solution is optimal
# status = 181 if solution is feasible, but, not optimal
# status = 182 if solution is not feasible
# status = 183 if there are no feasible solutions

[x_min,z_min,status,extra]=glpk(c,A,b,[0;0;0],[],"LLLL","CC
C",1)

```

*SCI/LAB* solution.

```

[xopt,lagr,fopt]=linpro(c,a,b,lb,ub,2, x0)
fopt =
4.
lagr =
0.
0.
- 12.
- 7.
0.
0.
- 3.
xopt =
1.
0.
0.

```

The shadow price for constraints (1) and (4) are 7 and 3 respectively. The dual problem has an optimal solution, indicated by status=180, and the minimum z-value is 4.

### 6.3.2. Economic Interpretation of Dual

#### **EXAMPLE 6.** Colorado's fractional furniture

The Colorado Furniture Company manufactures desks, tables, and chairs. The manufacture of each type of furniture requires lumber and two types of skilled labor: finishing labor and carpentry labor. The

amount of each resource needed to make each type of furniture is given in the table below:

**Table 1**

| Resource  | Product |         |           |
|-----------|---------|---------|-----------|
|           | Desk    | Table   | Chair     |
| Lumber    | 8 units | 6 units | 1 unit    |
| Finishing | 4 hours | 2 hours | 1.5 hours |

At present, 48 units of lumber, 20 finishing hours, and 8 carpentry hours are available. A desk sells for \$60, a table for \$30, and a chair for \$20.

Since the available resources have already been purchased, Colorado wants to maximize total revenue.

Colorado's fractional furniture problem is, given that it is okay to produce fractional amount of desks, tables, and chairs, a typical product selection problem. So, following Algorithm 3.1 in the same way as in the Gepetto's problem (Example 2), we find that the LP for Colorado's fractional furniture example 3 is

$$\max z = 60x_1 + 30x_2 + 20x_3$$

s.t.

$$\begin{aligned} 8x_1 + 6x_2 + x_3 &\leq 48 \quad (\text{lumber}) \\ 4x_1 + 2x_2 + 1.5x_3 &\leq 20 \quad (\text{finishing}) \\ 2x_1 + 1.5x_2 + 0.5x_3 &\leq 8 \quad (\text{carpentry}) \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

where

- $x_1$  = number of desks manufactured
- $x_2$  = number of tables manufactured
- $x_3$  = number of chairs manufactured

Now, the dual of this problem is

$$\min w = 48y_1 + 20y_2 + 8y_3$$

s.t.

$$\begin{aligned}
8y_1 + 4y_2 + 3y_3 &\geq 60 \text{ (desk)} \\
6y_1 + 2y_2 + 1.5y_3 &\geq 30 \text{ (table)} \\
y_1 + 1.5y_2 + 0.5y_3 &\geq 20 \text{ (chair)} \\
y_1, y_2, y_3 &\geq 0
\end{aligned} \tag{6.3}$$

We have given the constraints the names (desk), (table), and (chair). Those were the decision variables  $x_1$ ,  $x_2$  and  $x_3$  in the primal LP. By symmetry, or duality, we could say that  $y_1$  is associated with lumber,  $y_2$  with finishing, and  $y_3$  with carpentry. What is going on here? It is instructive to represent all the data of the Colorado's problem in a single table:

**Table 2**

| Resource  | Product |           |           | Availability |
|-----------|---------|-----------|-----------|--------------|
|           | Desk    | Table     | Chair     |              |
| Lumber    | 8 units | 6 units   | 1 unit    | 48 units     |
| Finishing | 4 hours | 2 hours   | 1.5 hours | 20 hours     |
| Carpentry | 2 hours | 1.5 hours | 0.5 hours | 8 hours      |
| Price     | \$60    | \$30      | \$20      |              |

Now, the table above can be read either horizontally or vertically. You should already know how to read the table above horizontally. That is the Colorado's point of view. But what does it mean to read the table vertically? Here is the explanation, which is also the **economic interpretation** of the dual LP:

Suppose you are an entrepreneur who wants to purchase all of Colorado's resources — maybe you are a competing furniture manufacturer, or maybe you need the resources to produce soldiers and trains like Gepetto. Then you must determine the price you are willing to pay for a unit of each of Colorado's resources. But what are the Colorado's resources? Well they are lumber, finishing hours, and carpentry hours, that Colorado uses to make its products. So, the **decision variables** for the entrepreneur who wants to buy Colorado's resources are:

$y_1$  = price to pay for one unit of lumber

$y_2$  = price to pay for one hour of finishing labor

$y_3$  = price to pay for one hour of carpentry labor

Now we argue that the resource prices  $y_1, y_2, y_3$  should be determined by solving the Colorado's dual (5.2.5).

First note that you are buying **all** of Colorado's resources. Also, note that this is a minimization problem: You want to pay as little as possible. So, the **objective function** is

$$\min w = 48y_1 + 20y_2 + 8y_3.$$

Indeed, Colorado has 48 units of lumber, 20 hours of finishing labor, and 8 hours of carpentry labor.

Now we have the decision variables and the objective. How about the **constraints**? In setting the resource prices  $y_1, y_2$ , and  $y_3$ , what kind of constraints do you face? You must set the resource prices high enough so that Colorado would sell them to you. Now Colorado can either use the resources itself, or sell them to you. How is Colorado using its resources? Colorado manufactures desks, tables, and chair. Take desks first. With 8 units of lumber, 4 hours of finishing labor, and 2 hours of carpentry labor Colorado can make a desk that will sell for \$60. So, you have to offer more than \$60 for this particular combination of resources. So, you have the constraint

$$8y_1 + 4y_2 + 2y_3 \geq 60.$$

But this is just the first constraint in the Colorado's dual, denoted by (desk). Similar reasoning shows that you must pay at least \$30 for the resources Colorado uses to produce one table. So, you get the second constraint, denoted by (table), of the Colorado's dual:

$$6y_1 + 2y_2 + 1.5y_3 \geq 30.$$

Similarly, you must offer more than \$20 for the resources Colorado can use itself to produce one chair. That way you get the last constraint, labeled as (chair), of the Colorado's dual:

$$y_1 + 1.5y_2 + 0.5y_3 \geq 20.$$

We have just interpreted economically the dual of a maximization problem. Let us then change our point of view to the opposite and interpret economically the dual of a minimization problem.

## 6.4. Importance of Sensitivity Analysis

Sensitivity analysis is important for several reasons. In many applications, the values of an LP's parameters may change. For example, the prices at which soldiers and trains are sold or the availability of carpentry and finishing hours may change. If a parameter changes, then sensitivity analysis often makes it unnecessary to solve the problem again. For example, if the profit contribution of a soldier increased to \$3.50, we would not have to solve the Gepetto problem again, because the current solution remains optimal. Of course, solving the Gepetto problem again would not be much work, but solving an LP with thousands of variables and constraints again would be a chore. A knowledge of sensitivity analysis often enables the analyst to determine from the original solution how changes in an LP's parameters change its optimal solution.

Recall that we may be uncertain about the values of parameters in an LP. For example, we might be uncertain about the weekly demand for soldiers. With the graphical method, it can be shown that if the weekly demand for soldiers is at least 20, then the optimal solution to the Gepetto problem is still (20, 60) (see Problem 3 at the end of this section). Thus, even if Gepetto is uncertain about the demand for soldiers, the company can be fairly confident that it is still optimal to produce 20 soldiers and 60 trains.

## 6.5. SUMMARY. Graphical Sensitivity Analysis

To determine whether the current basis remains optimal after changing an objective function coefficient, note that changing the objective function coefficient of a variable changes the slope of the isoprofit line. The current basis remains optimal as long as the current optimal solution is the last point in the feasible region to make contact with isoprofit lines as we move in the direction of increasing  $z$  (for a max problem). If the current basis remains optimal, the values of the decision variables remain unchanged, but the optimal  $z$ -value may change.

To determine if the current basis remains optimal after changing the right-hand side of a constraint, begin by finding the constraints (possibly including sign restrictions) that are binding for the current optimal solution. As we change the right-hand side of a constraint, the current basis remains optimal as long as the point where the constraints are binding remains feasible. Even if the current basis remains optimal, the values of the decision variables and the optimal  $z$ -value may change.

### 6.5.1. Shadow Prices

The **shadow price** of the  $i$ th constraint of a linear programming problem is the amount by which the optimal  $z$ -value is improved if the right-hand side of the  $i$ th constraint is increased by 1 (assuming that the current basis remains optimal). The shadow price of the  $i$ th constraint is the dual price for row  $i + 1$  given on the *SCILAB* output.

If the right-hand side of the  $i$ th constraint is increased by  $.1b_i$ , then (assuming the current basis remains optimal) the new optimal  $z$ -value for a maximization problem may be found as follows:

$$(\text{New optimal } z\text{-value}) = (\text{old optimal } z\text{-value})$$

$$+ (\text{Constraint } i\text{'s shadow price}) \cdot .1b_i \quad (1)$$

For a minimization problem, the new optimal  $z$ -value may be found from

$$(\text{New optimal } z\text{-value}) = (\text{old optimal } z\text{-value})$$

$$- (\text{Constraint } i\text{'s shadow price}) \cdot .1b_i \quad (2)$$

### 6.5.2. Objective Function Coefficient Range

The OBJ COEFFICIENT RANGE portion of the *SCILAB* output gives the range of values for an objective function coefficient for which the current basis remains optimal. Within this range, the values of the decision variables remain unchanged, but the optimal  $z$ -value may or may not change.

### 6.5.3. Reduced Cost

For any nonbasic variable, the reduced cost for the variable is the amount by which the nonbasic variable's objective function coefficient

must be improved before that variable will become a basic variable in some optimal solution to the LP.

#### 6.5.4. Right-Hand Side Range

If the right-hand side of a constraint remains within the Righthand Side Ranges value given on the *SCILAB* results, then the current basis remains optimal, and the dual price may be used to determine how a change in the right-hand side changes the optimal  $z$ -value. Even if the right-hand side of a constraint remains within the Righthand Side Ranges value on the *SCILAB* output, then the values of the decision variables will probably change.

#### 6.5.5. Signs of Shadow Prices

A 2: constraint will have a nonpositive shadow price; a  $<$  constraint will have a nonnegative shadow price; and an equality constraint may have a positive, negative, or zero shadow price.

#### 6.5.6. Optimal $z$ -Value as a Function of a Constraint's Right-Hand Side

In all cases, the optimal  $z$ -value will be a piecewise linear function of a constraint's right-hand side. The exact form of the function is as shown in Table 3.

#### 6.5.7. Finding the Dual of an LP

For a normal (all  $<$  constraints and all variables nonnegative) max problem or a normal min (all 2: constraints and all variables nonnegative) problem, we find the dual as follows:

If we read the primal across in Table 14, we read the dual down. If we read the primal down in Table 14, we read the dual across. We use  $x_i$ 's and  $z$  as variables for a maximization problem and  $y_j$ 's and  $w$  as variables for a minimization problem.

To find the dual of a nonnormal max problem:

**Step 1.** Fill in Table 14 so that the primal can be read across.

**Step 2.** After making the following changes, the dual can be read down in the usual fashion: (a) If the  $i$ th primal constraint is a 2: constraint, the

corresponding dual variable  $y_i$  must satisfy  $y_i < 0$ . (b) If the  $i$ th primal constraint is an equality, then the dual variable  $y_i$  is now unrestricted in sign (**urs**). (c) If the  $i$ th primal variable is urs, then the  $i$ th dual constraint will be an equality.

To find the dual of a nonnormal min problem:

**Step 1.** Write out the primal so it can be read down in Table 14.

**Step 2.** Except for the following changes, the dual can be read across the table: (a) If the  $i$ th primal constraint is a  $<$  constraint, then the corresponding dual variable  $x_i$  must satisfy  $x_i < 0$ . (b) If the  $i$ th primal constraint is an equality, then the corresponding dual variable  $x_i$  will be urs. (c) If the  $i$ th primal variable  $y_i$  is urs, then the  $i$ th dual constraint is an equality.

### 6.5.8. Optimal z-Value as a Function of an Objective Function Coefficient

In a maximization problem, the optimal  $z$ -value will be a nondecreasing, piecewise linear function of an objective function coefficient. The slope will be a nondecreasing function of the objective function coefficient.

TABLE 3

| Type of LP   | Type of Constraint | Slopes of Segments Are                 | Piecewise Linear |
|--------------|--------------------|--|------------------|
| Maximization | $\leq$             | Nonnegative and nonincreasing          |                  |
| Maximization | $\geq$             | Nonpositive and nonincreasing          |                  |
| Maximization | $=$                | Unrestricted in sign and nonincreasing |                  |
| Minimization | $\leq$             | Nonpositive and nondecreasing          |                  |
| Minimization | $\geq$             | Nonnegative and nondecreasing          |                  |
| Minimization | $=$                | Unrestricted in sign and nondecreasing |                  |

In a minimization problem, the optimal z-value will be a nondecreasing, piecewise linear function of an objective function coefficient. The slope will be a nonincreasing function of the objective function coefficient.

## 6.6. Review Problems

1. In the Colorado problem, show that the current basis remains optimal if  $c_3$ , the price of chairs, satisfies  $15 < c_3 < 22.5$ . If  $c_3 = 21$ , find the new optimal solution. Also,  $c_3 = 25$ , find the new optimal solution.
2. If  $c_1 = 55$  in the Colorado problem, show that the new optimal solution does not produce any desks.
3. Holiday Meal Turkey Ranch is considering buying two different types of turkey feed. Each feed contains, in varying proportions, some or all of the three nutritional ingredients essential for fattening turkeys. Brand Y feed costs the ranch \$.02 per pound. Brand Z costs \$.03 per pound. The rancher would like to determine the lowest-cost diet that

**TABLE 4. COMPOSITION OF EACH POUND OF FEED**

| INGREDIENT | BRAND Y<br>FEED | BRAND Z<br>FEED | MINIMUM<br>REQUIREMENT | MONTHLY |
|------------|-----------------|-----------------|------------------------|---------|
| A          | 5 oz.           | 10 oz.          | 90 oz.                 |         |
| B          | 4 oz.           | 3 oz.           | 48 oz.                 |         |
| C          | .5 oz.          | 0               | 1.5 oz.                |         |
| Cost/lb    | \$.02           | \$.03           |                        |         |

4. Ed Silver Dog Food Company wishes to introduce a new brand of dog biscuits composed of chicken and liver flavored biscuits that meet certain nutritional requirements. The liver-flavored biscuits contain 1 unit of nutrient A and 2 units of nutrient B; the chicken-flavored biscuits contain 1 unit of nutrient A and 4 units of nutrient B. According to federal requirements, there must be at least 40 units

of nutrient A and 60 units of nutrient B in a package of the new mix. In addition, the company has decided that there can be no more than 15 liver-flavored biscuits in a package. If it costs 1¢ to make 1 liver-flavored biscuit and 2¢ to make 1 chicken-flavored, what is the optimal product mix for a package of the biscuits to minimize the firm's cost?

- a) Formulate this as a linear programming problem.
  - b) Solve this problem graphically, giving the optimal values of all variables.
  - c) Solve this problem using *SCILAB*.
  - d) What is the total cost of a package of dog biscuits using the optimal mix?
  - e) What are the shadow costs?
5. A fertilizer manufacturer has to fulfill supply contracts to its two main customers (650 tons to Customer A and 800 tons to Customer B). It can meet this demand by shipping existing inventory from any of its three warehouses. Warehouse 1 (W1) has 400 tons of inventory on-hand, Warehouse 2 (W2) has 500 tons, and Warehouse 3 (W3) has 600 tons. The company would like to arrange the shipping for the lowest cost possible, where the per-ton transit costs are as follows:

|            | W1     | W3     | W3     |
|------------|--------|--------|--------|
| Customer A | \$7.50 | \$6.25 | \$6.50 |
| Customer B | 6.75   | 7.00   | 8.00   |

- a) Explain what each of the six decision variables (V) is: (*Hint: Look at the Solver report below.*)

V1: \_\_\_\_\_

V2: \_\_\_\_\_

V3: \_\_\_\_\_

V4: \_\_\_\_\_

V5: \_\_\_\_\_

V6: \_\_\_\_\_

b) Write out the objective function in terms of the variables (V1, V2, etc.) and the objective coefficients.

c) Aside from nonnegativity of the variables, what are the five constraints? Write a short description for each constraint, and write out the formula (and circle the type of equality/inequality).

| Description | Variables and Coefficients          | RHS |
|-------------|-------------------------------------|-----|
| C1:         | Formula: $(\Rightarrow   =   \leq)$ |     |
| C2:         | Formula: $(\Rightarrow   =   \leq)$ |     |
| C3:         | Formula: $(\Rightarrow   =   \leq)$ |     |
| C4:         | Formula: $(\Rightarrow   =   \leq)$ |     |
| C5:         | Formula: $(\Rightarrow   =   \leq)$ |     |

After you formulate and enter the linear program in Excel, the Solver gives you the following sensitivity report:

#### Adjustable Cells

| CELL   | NAME | FINAL VALUE | REDUCED COST | OBJECTIVE COEFFICIENT | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
|--------|------|-------------|--------------|-----------------------|--------------------|--------------------|
| \$B\$6 | V1   | 0           | 1.5          | 7.5                   | 1E+30              | 1.5                |
| \$C\$6 | V2   | 100         | 0            | 6.25                  | 0.25               | 0.75               |
| \$D\$6 | V3   | 550         | 0            | 6.5                   | 0.75               | 0.25               |
| \$E\$6 | V4   | 400         | 0            | 6.75                  | 0.5                | 1E+30              |
| \$F\$6 | V5   | 400         | 0            | 7                     | 0.75               | 0.5                |
| \$G\$6 | V6   | 0           | 0.75         | 8                     | 1E+30              | 0.75               |

#### Constraints

| CELL    | NAME | FINAL<br>VALUE | SHADOW<br>PRICE | CONSTRAINT<br>R.H. SIDE | ALLOWABLE<br>INCREASE | ALLOWABLE<br>DECREASE |
|---------|------|----------------|-----------------|-------------------------|-----------------------|-----------------------|
| \$H\$7  | C1   | 650            | 6.5             | 650                     | 50                    | 550                   |
| \$H\$8  | C2   | 800            | 7.25            | 800                     | 50                    | 400                   |
| \$H\$9  | C3   | 400            | -0.5            | 400                     | 400                   | 50                    |
| \$H\$10 | C4   | 500            | -0.25           | 500                     | 550                   | 50                    |
| \$H\$11 | C5   | 550            | 0               | 600                     | 1E+30                 | 50                    |

- d) How many of the constraints are binding?
- e) How much slack/surplus is there with the nonbinding constraint(s)?
- f) What is the range of optimality on variable V3? \_\_\_\_\_ to \_\_\_\_\_
- g) If we could ship 10 tons less to Customer A, how much money *might* we be able to save? \_\_\_\_\_ If we could chose to short *either* Customer A or Customer B by 10 tons, which would we prefer to short? Why?

6. **Silvery Landscaping and Plants, Inc.** Kenneth and Patricia Silvery spent a career as a husband-and-wife real estate investment partnership in Washington, DC. When they finally retired to a 25-acre farm in northern Virginia's Fairfax County, they became ardent amateur gardeners. Kenneth planted shrubs and fruit trees, and Patricia spent her hours potting all sizes of plants. When the volume of shrubs and plants reached the point that the Silverys began to think of their hobby in a serious vein, they built a greenhouse adjacent to their home and installed heating and watering systems.

By 2005, the Silverys realized their retirement from real estate had really only led to a second career—in the plant and shrub business—and they filed for a Virginia business license. Within a matter of months, they asked their attorney to file incorporation documents and formed the firm Silvery Landscaping and Plants, Inc.

Early in the new business's existence, Kenneth Silvery recognized the need for a high-quality commercial fertilizer that he could blend himself, both for sale and for his own nursery. His goal was to keep his costs to a

minimum while producing a top-notch product that was especially suited to the northern Virginia climate.

Working with chemists at George Mason University, Silvery blended "Silvery-Grow." It consists of four chemical compounds, C-30, C-92, D-21, and E-11. The cost per pound for each compound is indicated in Table 1.

**Table 1.** Silvery-Grow Compounds

| CHEMICAL COMPOUND | COST PER POUND (\$) |
|-------------------|---------------------|
| C-30              | 0.12                |
| C-92              | 0.09                |
| D-21              | 0.11                |
| E-11              | 0.04                |

The specifications for Silvery-Grow are established as:

- Chemical E-11 must constitute at least 15% of the blend.
- C-92 and C-30 must together constitute at least 45% of the blend.
- D-21 and C-92 can together constitute no more than 30% of the blend.
- Silvery-Grow is packaged and sold in 50-lb bags.

### Questions

- a. Formulate an LP problem to determine what blend of the four chemicals will allow Silvery to minimize the cost of a 50-lb bag of the fertilizer.
  - b. Solve to find the best solution.
7. The Denver advertising agency promoting the new Breem dishwashing detergent wants to get the best exposure possible for the product within the \$100,000 advertising budget ceiling placed on it. To do so, the agency needs to decide how much of the budget to spend on each of its two most effective media: (1) television spots

during the afternoon hours and (2) large ads in the city's Sunday newspaper. Each television spot costs \$3,000; each Sunday newspaper ad costs \$1,250. The expected exposure, based on industry ratings, is 35,000 viewers for each TV commercial and 20,000 readers for each newspaper advertisement. The agency director, Deborah Kellogg, knows from experience that it is important to use both media in order to reach the broadest spectrum of potential Breem customers. She decides that at least 5 but no more than 25 television spots should be ordered, and that at least 10 newspaper ads should be contracted. How many times should each of the two media be used to obtain maximum exposure while staying within the budget? Use the graphical method to solve.

8. Graph the following LP problem:

$$\text{max profit} = \$3x_1 + \$2x_2$$

s.t.:

$$\begin{aligned} 2x_1 + 1x_2 &\leq 150 \\ 2x_1 + 3x_2 &\leq 300 \end{aligned}$$

9. Boston's famous Limoges Restaurant is open 24 hours a day. Servers report for duty at 3 A.M., 7 A.M., 11 A.M., 3 P.M., 7 P.M., or 11 P.M., and each works an 8-hour shift. The following table shows the minimum number of workers needed during the 6 periods into which the day is divided.

| PERIOD | TIME            | NUMBER OF SERVERS REQUIRED |
|--------|-----------------|----------------------------|
| 1      | 3 A.M.– 7 A.M.  | 3                          |
| 2      | 7 A.M.–11 A.M.  | 12                         |
| 3      | 11 A.M.– 3 P.M. | 16                         |
| 4      | 3 P.M.– 7 P.M.  | 9                          |
| 5      | 7 P.M.–11 P.M.  | 11                         |
| 6      | 11 P.M.– 3 A.M. | 4                          |

Owner Michelle Limoges' scheduling problem is to determine how many servers should report for work at the start of each time period in order to minimize the total staff required for one day's operation. (*Hint:* Let  $x_i$  equal the number of servers beginning work in time period  $i$ , where  $i = 1, 2, 3, 4, 5, 6.$ )

10. The National Credit Union has \$250,000 available to invest in a 12-month commitment. The money can be placed in Treasury notes yielding an 8% return or in municipal bonds at an average rate of return of 9%. Credit union regulations require diversification to the extent that at least 50% of the investment be placed in Treasury notes. Because of defaults in such municipalities as Cleveland and New York, it is decided that no more than 40% of the investment be placed in bonds. How much should the National Credit Union invest in each security so as to maximize its return on investment?
11. Kalyan Singhal Corp. makes three products, and it has three machines available as resources as given in the following LP problem:

$$\text{max contribution} = 4x_1 + 4x_2 + 7x_3$$

s.t.:

$$\begin{aligned}1x_1 + 7x_2 + 4x_3 &\leq 100 \quad (\text{hours on machine 1}) \\2x_1 + 1x_2 + 7x_3 &\leq 110 \quad (\text{hours on machine 2}) \\8x_1 + 4x_2 + 1x_3 &\leq 100 \quad (\text{hours on machine 3})\end{aligned}$$

- a. Determine the optimal solution using LP software.
- b. Is there unused time available on any of the machines with the optimal solution?
- c. What would it be worth to the firm to make an additional hour of time available on the third machine?
- d. How much would the firm's profit increase if an extra 10 hours of time were made available on the second machine at no extra cost?

12. Consider the following linear programming problem:

$$\max z = 30x_1 + 10x_2$$

s.t.:

$$\begin{aligned}3x_1 + x_2 &\leq 300 \\x_1 + x_2 &\leq 200 \\x_1 &\leq 100 \\x_2 &\geq 50 \\x_1 - x_2 &\leq 0 \\x_1, x_2 &\geq 0\end{aligned}$$

- a. Solve the problem graphically.
- b. Is there more than one optimal solution? Explain.



## 7. What is CPM and PERT?

Network models can be used as an aid in scheduling large complex projects that consist of many activities. If the duration of each activity is known with certainty, then the **critical path method (CPM)** can be used to determine the length of time required to complete a project. CPM also can be used to determine how long each activity in the project can be delayed without delaying the completion of the project. CPM was developed in the late 1950s by researchers at DuPont and Sperry Rand.

If the duration of the activities is not known with certainty, the **Program Evaluation and Review Technique (PERT)** can be used to estimate the probability that the project will be completed by a given deadline. PERT was developed in the late 1950s by consultants working on the development of the Polaris missile. CPM and PERT were given a major share of the credit for the fact that the Polaris missile was operational two years ahead of schedule.

CPM and PERT have been successfully used in many applications, including:

1. Scheduling construction projects such as office buildings, highways, and swimming pools
2. Scheduling the movement of a 400-bed hospital from Portland, Oregon, to a suburban location
3. Developing a countdown and “hold” procedure for the launching of space flights
4. Installing a new computer system
5. Designing and marketing a new product
6. Completing a corporate merger
7. Building a ship

To apply CPM and PERT, we need a list of the activities that make up the project. The project is considered to be completed when all the activities have been completed. For each activity, there is a set of activities (called the **predecessors** of the activity) that must be completed before the activity begins. A project network is used to represent the precedence relationships between activities. In our discussion, activities will be

represented by directed arcs, and nodes will be used to represent the completion of a set of activities. (For this reason, we often refer to the nodes in our project network as **events**.) This type of project network is called an **AOA (activity on arc)** network.

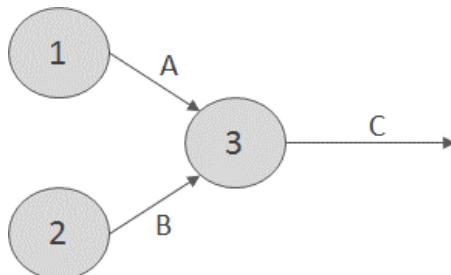
To understand how an AOA network represents precedence relationships, suppose that activity A is a predecessor of activity B. Each node in an AOA network represents the completion of one or more activities. Thus, node 2 in Figure 1 represents the completion of activity A and the beginning of activity B. Suppose activities A and B must be completed before activity C can begin. In Figure 2, node 3 represents the event that activities A and B are completed. Figure 3 shows activity A as a predecessor of both activities B and C.

Given a list of activities and predecessors, an AOA representation of a project (called a **project network** or **project diagram**) can be constructed by using the following rules:

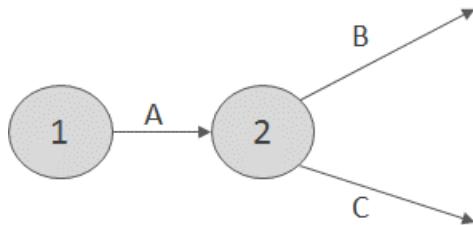
1. Node 1 represents the start of the project. An arc should lead from node 1 to represent each activity that has no predecessors.



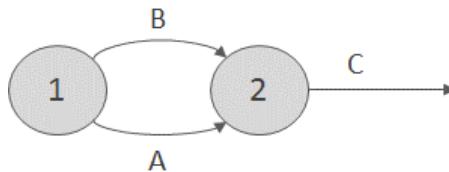
**Figure 1.** Activity A Must Be Completed Before Activity B Can Begin



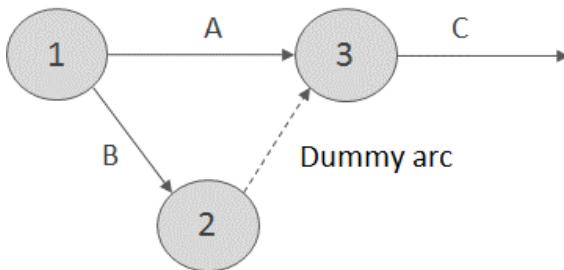
**Figure 2.** Activities A and B Must Be Completed Before Activity C Can Begin



**Figure 3.** Activity A Must Be Completed Before Activities B and C Can Begin



**Figure 4.** Violation of Rule 5



**Figure 5.** Use of Dummy Activity

In an AON (activity on node) project network, the nodes of the network are used to represent activities. See Wiest and Levy (1977) for details.

2. A node (called the finish node) representing the completion of the project should be included in the network.
3. Number the nodes in the network so that the node representing the completion of an activity always has a larger number than the node representing the beginning of an activity (there may be more than one numbering scheme that satisfies rule 3).
4. An activity should not be represented by more than one arc in the network.
5. Two nodes can be connected by at most one arc.

To avoid violating rules 4 and 5, it is sometimes necessary to utilize a dummy activity that takes zero time. For example, suppose activities A and B are both predecessors of activity C and can begin at the same time. In the absence of rule 5, we could represent this by Figure 4. However, because nodes 1 and 2 are connected by more than one arc, Figure 4 violates rule 5. By using a dummy activity (indicated by a dotted arc), as in Figure 5, we may represent the fact that A and B are both predecessors of C. Figure 5 ensures that activity C cannot begin until both A and B are completed, but it does not violate rule 5. Problem 10 at the end of this section illustrates how dummy activities may be needed to avoid violating rule 4. Example 6 illustrates a project network.

### **EXAMPLE 1.** Drawing a Project Network

Gadgetco is about to introduce a new product (product 3). One unit of product 3 is produced by assembling 1 unit of product 1 and 1 unit of product 2. Before production begins on either product 1 or 2, raw materials must be purchased and workers must be trained. Before products 1 and 2 can be assembled into product 3, the finished product 2 must be inspected. A list of activities and their predecessors and of the duration of each activity is given in Table 1. Draw a project diagram for this project.

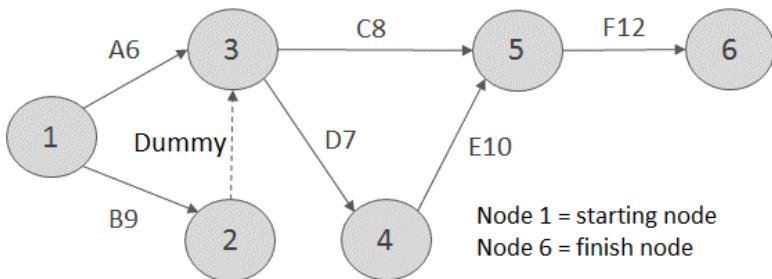
**TABLE 1.** Duration of Activities and Predecessor Relationships for Gadgetco

| Activity                      | Predecessors | Duration (Days) |
|-------------------------------|--------------|-----------------|
| A = train workers             | —            | 6               |
| B = purchase raw materials    | —            | 9               |
| C = produce product 1         | A, B         | 8               |
| D = produce product 2         | A, B         | 7               |
| E = test product 2            | D, B         | 10              |
| F = assemble products 1 and 2 | C, E         | 12              |

**Solution.** Observe that although we list only C and E as predecessors of F, it is actually true that activities A, B, and D must also be completed before F begins. C cannot begin until A and B are completed, and E cannot begin until D is completed, however, so it is redundant to state

that  $A$ ,  $B$ , and  $D$  are predecessors of  $F$ . Thus, in drawing the project network, we need only be concerned with the immediate predecessors of each activity.

The AOA network for this project is given in Figure 6 (the number above each arc represents activity duration in days). Node 1 is the beginning of the project, and node 6 is the finish node representing completion of the project. The dummy arc (2, 3) is needed to ensure that rule 5 is not violated.



**Figure 6.** Diagram for Gadgetco

The two key building blocks in CPM are the concepts of early event time (ET) and late event time (LT) for an event

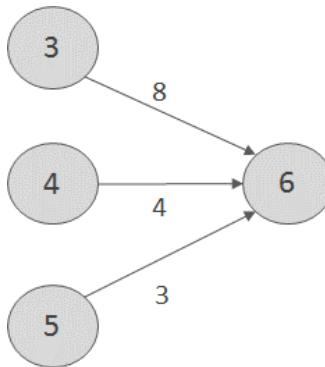
**DEFINITION** ■ The early event time for node  $i$ , represented by  $ET(i)$ , is the earliest time at which the event corresponding to node  $i$  can occur. ■

The late event time for node  $i$ , represented by  $LT(i)$ , is the latest time at which the event corresponding to node  $i$  can occur without delaying the completion of the project. ■

## 7.1. Computation of Early Event Time

To find the early event time for each node in the project network, we begin by noting that because node 1 represents the start of the project,  $ET(1) = 0$ . We then compute  $ET(2)$ ,  $ET(3)$ , and so on, stopping when  $ET(\text{finish node})$  has been calculated. To illustrate how  $ET(i)$  is calculated, suppose that for the segment of a project network in Figure 7, we have already determined that  $ET(3) = 6$ ,  $ET(4) = 8$ , and  $ET(5) = 10$ . To determine  $ET(6)$ , observe that the earliest time that

node 6 can occur is when the activities corresponding to arc (3, 6), (4, 6), and (5, 6) have all been completed.



**Figure 7.** Determination of  $ET(6)$

$$ET(6) = \max \begin{cases} ET(3) + 8 = 14 \\ ET(4) + 4 = 12 \\ ET(5) + 3 = 13 \end{cases}$$

Thus, the earliest time that node 6 can occur is 14, and  $ET(6) = 14$ .

From this example, it is clear that computation of  $ET(i)$  requires (for  $j < i$ ) knowledge of one or more of the  $ET(j)$ 's. This explains why we begin by computing the predecessor ETs. In general, if  $ET(1), ET(2), \dots, ET(i-1)$  have been determined, then we compute  $ET(i)$  as follows:

**Step 1.** Find each prior event to node  $i$  that is connected by an arc to node  $i$ . These events are the immediate predecessors of node  $i$ .

**Step 2.** To the  $ET$  for each immediate predecessor of the node  $i$  add the duration of the activity connecting the immediate predecessor to node  $i$ .

**Step 3.**  $ET(i)$  equals the maximum of the sums computed in step 2.

We now compute the  $ET(i)$ 's for Example 1. We begin by observing that  $ET(1) = 0$ . Node 1 is the only immediate predecessor of node 2, so

$ET(2) = ET(1) + 9 = 9$ . The immediate predecessors of node 3 are nodes 1 and 2. Thus,

$$ET(3) = \max \begin{cases} ET(1) + 6 = 6 \\ ET(2) + 0 = 9 \end{cases} = 9$$

Node 4's only immediate predecessor is node 3. Thus,  $ET(4) = ET(3) + 7 = 16$ . Node 5's immediate predecessors are nodes 3 and 4. Thus,

$$ET(5) = \max \begin{cases} ET(3) + 8 = 17 \\ ET(4) + 10 = 26 \end{cases} = 26$$

Finally, node 5 is the only immediate predecessor of node 6. Thus,  $ET(6) = ET(5) + 12 = 38$ . Because node 6 represents the completion of the project, we see that the earliest time that product 3 can be assembled is 38 days from now.

It can be shown that  $ET(i)$  is the length of the longest path in the project network from node 1 to node  $i$ .

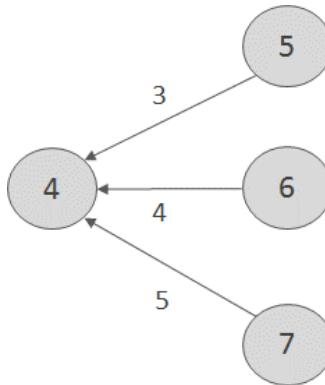
## 7.2. Computation of Late Event Time

To compute the late even times,  $LT(i)$ 's, we begin with the finish node and work backward (in descending numerical order) until we determine  $LT(1)$ . The project in Example 1 can be completed in 38 days, so we know that  $LT(6)=38$ . To illustrate how  $LT(i)$  is computed for nodes other than the finish node, suppose we are working with a network (Figure 8) for which we have already determined that  $LT(5) = 24$ ,  $LT(6) = 26$ , and  $LT(7) = 28$ . In this situation, how can we compute  $LT(4)$ ? If the event corresponding to node 4 occurs after  $LT(5) - 3$ , node 5 will occur after  $LT(5)$ , and the completion of the project will be delayed.

Similarly, if node 4 occurs after  $LT(6) - 4$  or if node 4 occurs after  $LT(7) - 5$ , the completion of the project will be delayed. Thus,

$$LT(4) = \min \begin{cases} LT(5) - 3 = 21 \\ LT(6) - 4 = 22 = 21 \\ LT(7) - 5 = 23 \end{cases}$$

In general, if  $LT(j)$  is known for  $j > i$ , we can find  $LT(i)$  as follows:



**Figure 8.** Computation of  $LT(4)$

**Step 1.** Find each node that occurs after node  $i$  and is connected to node  $i$  by an arc. These events are the immediate successors of node  $i$ .

**Step 2.** From the  $LT$  for each immediate successor to node  $i$ , subtract the duration of the activity joining the successor the node  $i$ .

**Step 3.**  $LT(i)$  is the smallest of the differences determined in step 2.

We now compute the  $LT(i)$ 's for Example 1. Recall that  $LT(6) = 38$ . Because node 6 is the only immediate successor of node 5,  $LT(5) = LT(6) - 12 = 26$ . Node 4's only im- mediate successor is node 5. Thus,  $LT(4) = LT(5) - 10 = 16$ . Nodes 4 and 5 are imme- diate successors of node 3. Thus,

$$LT(3) = \min \begin{cases} LT(4) - 7 = 9 \\ LT(5) - 8 = 18 \end{cases}$$

Node 3 is the only immediate successor of node 2. Thus,  $LT(2) = LT(3) - 0 = 9$ . Finally, node 1 has nodes 2 and 3 as immediate successors. Thus,

$$LT(1) = \min \begin{cases} LT(3) - 6 = 3 \\ LT(2) - 9 = 0 \end{cases}$$

Table 2 summarizes our computations for Example 1. If  $LT(i) = ET(i)$ , any delay in the occurrence of node  $i$  will delay the completion of the project. For example, because  $LT(4) = ET(4)$ , any delay in the occurrence of node 4 will delay the completion of the project.

**TABLE 2.** ET and LT for Gadgetco

| Node | $ET(i)$ | $LT(i)$ |
|------|---------|---------|
| 1    | 0       | 0       |
| 2    | 9       | 9       |
| 3    | 9       | 9       |
| 4    | 16      | 16      |
| 5    | 26      | 26      |
| 6    | 38      | 38      |

### 7.3. Total Float

Before the project is begun, the duration of an activity is unknown, and the duration of each activity used to construct the project network is just an estimate of the activity's actual completion time. The concept of total float of an activity can be used as a measure of how important it is to keep each activity's duration from greatly exceeding our estimate of its completion time.

**DEFINITION ■** For an arbitrary arc representing activity  $(i, j)$ , the total float, represented by  $TF(i, j)$ , of the activity represented by  $(i, j)$  is the amount by which the starting time of activity  $(i, j)$  could be delayed beyond its earliest possible starting time without delaying the completion of the project (assuming no other activities are delayed). ■

Equivalently, the total float of an activity is the amount by which the duration of the activity can be increased without delaying the completion of the project.

If we define  $t_{ij}$  to be the duration of activity  $(i, j)$ , then  $TF(i, j)$  can easily be expressed in terms of  $LT(j)$  and  $ET(i)$ . Activity  $(i, j)$  begins at node  $i$ . If the occurrence of node  $i$ , or the duration of activity  $(i, j)$ , is delayed by  $k$  time units, then activity  $(i, j)$  will be completed at time  $ET(i) + k + t_{ij}$ . Thus, the completion of the project will not be delayed if

$$ET(i) + k + t_{ij} < LT(j) \text{ or } k < LT(j) - ET(i) - t_{ij}$$

Therefore,

(7.1)

$$TF(i,j) = LT(j) - ET(i) - t_{ij} \quad (7.2)$$

For Example 1, the  $TF(i,j)$  are as follows:

$$\text{Activity B: } TF(1,2) = LT(2) - ET(1) - 9 = 0$$

$$\text{Activity A: } TF(1,3) = LT(3) - ET(1) - 6 = 3$$

$$\text{Activity D: } TF(3,4) = LT(4) - ET(3) - 7 = 0$$

$$\text{Activity C: } TF(3,5) = LT(5) - ET(3) - 8 = 9$$

$$\text{Activity E: } TF(4,5) = LT(5) - ET(4) - 10 = 0$$

$$\text{Activity F: } TF(5,6) = LT(6) - ET(5) - 12 = 0$$

$$\text{Dummy activity: } TF(2,3) = LT(3) - ET(2) - 0 = 0$$

## 7.4. Finding a Critical Path

If an activity has a total float of zero, then any delay in the start of the activity (or the duration of the activity) will delay the completion of the project. In fact, increasing the duration of an activity by  $Ll$  days will increase the length of the project by  $Ll$  days. Such an activity is critical to the completion of the project on time.

**DEFINITION ■** Any activity with a total float of zero is a critical activity. ■

A path from node 1 to the finish node that consists entirely of critical activities is called a critical path. ■

In Figure 31, activities  $B, D, E, F$ , and the dummy activity are critical activities and the path 1–2–3–4–5–6 is the critical path (it is possible for a network to have more than one critical path). A critical path in any project network is the longest path from the start node to the finish node.

Any delay in the duration of a critical activity will delay the completion of the project, so it is advisable to monitor closely the completion of critical activities.

## 7.5. Free Float

As we have seen, the total float of an activity can be used as a measure of the flexibility in the duration of an activity. For example, activity A can take up to 3 days longer than its scheduled duration of 6 days without delaying the completion of the project. Another measure of the flexibility available in the duration of an activity is free float.

**DEFINITION ■** The free float of the activity corresponding to arc  $(i, j)$ , denoted by  $FF(i, j)$ , is the amount by which the starting time of the activity corresponding to arc  $(i, j)$  (or the duration of the activity) can be delayed without delaying the start of any later activity beyond its earliest possible starting time. ■

Suppose the occurrence of node  $i$ , or the duration of activity  $(i, j)$ , is delayed by  $k$  units. Then the earliest that node  $j$  can occur is  $ET(i) + t_{ij} + k$ . Thus, if  $ET(i) + t_{ij} + k \leq ET(j)$ , or  $k \leq ET(j) - ET(i) - t_{ij}$ , then node  $j$  will not be delayed. If node  $j$  is not delayed, then no other activities will be delayed beyond their earliest possible starting times. Therefore,

$$FF(i, j) = ET(j) - ET(i) - t_{ij} \quad (7.3)$$

For Example 1, the  $FF(i, j)$  are as follows:

|             |                              |
|-------------|------------------------------|
| Activity B: | $FF(1,2) = 9 - 0 - 9 = 0$    |
| Activity A: | $FF(1,3) = 9 - 0 - 6 = 3$    |
| Activity D: | $FF(3,4) = 16 - 9 - 7 = 0$   |
| Activity C: | $FF(3,5) = 26 - 9 - 8 = 9$   |
| Activity E: | $FF(4,5) = 26 - 16 - 10 = 0$ |
| Activity F: | $FF(5,6) = 38 - 26 - 12 = 0$ |

For example, because the free float for activity  $C$  is 9 days, a delay in the start of activity  $C$  (or in the occurrence of node 3) or a delay in the duration of activity  $C$  of more than 9 days will delay the start of some later activity (in this case, activity F).

## 7.6. Using Linear Programming to Find a Critical Path

Although the previously described method for finding a critical path in a project network is easily programmed on a computer, linear programming can also be used to determine the length of the critical path. Define

$x_j$  = the time that the event corresponding to node  $j$  occurs

For each activity  $(i, j)$ , we know that before node  $j$  occurs, node  $i$  must occur and activity  $(i, j)$  must be completed. This implies that for each arc  $(i, j)$  in the project network,  $x_j \geq x_i + t_{ij}$ . Let  $F$  be the node that represents completion of the project. Our goal is to minimize the time required to complete the project, so we use an objective function of  $z = x_F - x_1$ .

To illustrate how linear programming can be used to find the length of the critical path, we apply the preceding approach to Example 1. The appropriate LP is

$$\min z = x_6 - x_1$$

s.t.

$$x_3 \geq x_1 + 6 \quad (\text{Arc } (1, 3)\text{constraint})$$

$$x_2 \geq x_1 + 9 \quad (\text{Arc } (1, 2)\text{constraint})$$

$$x_5 \geq x_3 + 8 \quad (\text{Arc } (3, 5)\text{constraint})$$

$$x_4 \geq x_3 + 7 \quad (\text{Arc } (3, 4)\text{constraint})$$

$$x_5 \geq x_4 + 10 \quad (\text{Arc } (4, 5)\text{constraint})$$

$$x_6 \geq x_5 + 12 \quad (\text{Arc } (5, 6)\text{constraint})$$

$$x_3 \geq x_2 + 12 \quad (\text{Arc } (2, 3)\text{constraint})$$

All variables urs

An optimal solution to this LP is  $z = 38$ ,  $x_1 = 0$ ,  $x_2 = 9$ ,  $x_3 = 9$ ,  $x_4 = 16$ ,  $x_5 = 26$ , and  $x_6 = 38$ . This indicates that the project can be completed in 38 days.

This LP has many alternative optimal solutions. In general, the value of  $x_i$  in any optimal solution may assume any value between  $ET(i)$  and  $LT(i)$ . All optimal solutions to this LP, however, will indicate that the length of any critical path is 38 days.

A critical path for this project network consists of a path from the start of the project to the finish in which each arc in the path corresponds to a constraint having a dual price of  $-1$ . From the *SCILAB* output in Figure 34, we find, as before, that 1–2–3–4–5–6 is a critical path. For each constraint with a dual price of  $-1$ , increasing the duration of the activity corresponding to that constraint by  $Ll$  days will increase the duration of the project by  $Ll$  days. For example, an increase of  $Ll$  days in the duration of activity B will increase the duration of the project by  $Ll$  days. This assumes that the current basis remains optimal.

## 7.7. Crashing the Project

In many situations, the project manager must complete the project in a time that is less than the length of the critical path. For instance, suppose Gadgetco believes that to have any chance of being a success, product 3 must be available for sale before the competitor's product hits the market. Gadgetco knows that the competitor's product is scheduled to hit the market 26 days from now, so Gadgetco must introduce product 3 within 25 days. Because the critical path in Example 1 has a length of 38 days, Gadgetco will have to expend additional resources to meet the 25-day project deadline. In such a situation, linear programming can often be used to determine the allocation of resources that minimizes the cost of meeting the project deadline.

Suppose that by allocating additional resources to an activity, Gadgetco can reduce the duration of any activity by as many as 5 days. The cost per day of reducing the duration of an activity is shown in Table 14. To find the minimum cost of completing the project by the 25-day deadline, define variables  $A, B, C, D, E$ , and  $F$  as follows:

$A$  = number of days by which duration of activity  $A$  is reduced

$\vdots$   $\vdots$

$F$  = number of days by which duration of activity  $F$  is reduced

$x_j$  = time that the event corresponding to node  $j$  occurs

Then Gadgetco should solve the following LP:

$$\min z = 10A + 20B + 3C + 30D + 40E + 50F$$

$$A \leq 5$$

$$B \leq 5$$

$$C \leq 5$$

$$D \leq 5$$

$$E \leq 5$$

$$F \leq 5$$

```
c=[-1, 0, 0, 0, 0, 1];
A=[-1, 0, 1, 0, 0, 0;
   -1, 1, 0, 0, 0, 0;
   0, 0, -1, 0, 1, 0;
   0, 0, -1, 1, 0, 0;
   0, 0, 0, -1, 1, 0;
   0, 0, 0, 0, -1, 1;
   0, -1, 1, 0, 0, 0];
b=[6; 9; 8; 7; 10; 12; 0];
[[xopt,lagr,fopt]=linpro(c,A,b,lb,ub,0,x0)
fopt =
38.
lagr =
1.
0.
0.
0.
0.
- 1.
0.
0.
xopt =
0.
9.
9.
16.
26.
```

**TABLE 3**

| A    | B    | C   | D    | E    | F    |
|------|------|-----|------|------|------|
| \$10 | \$20 | \$3 | \$30 | \$40 | \$50 |

s.t.

$$\begin{aligned}
 x_2 &\geq x_1 + 9 - B && (\text{Arc } (1, 2)\text{constraint}) \\
 x_3 &\geq x_1 + 6 - A && (\text{Arc } (1, 3)\text{constraint}) \\
 x_5 &\geq x_3 + 8 - C && (\text{Arc } (3, 5)\text{constraint}) \\
 x_4 &\geq x_3 + 7 - D && (\text{Arc } (3, 4)\text{constraint}) \\
 x_5 &\geq x_4 + 10 - E && (\text{Arc } (4, 5)\text{constraint}) \\
 x_6 &\geq x_5 + 12 - F && (\text{Arc } (5, 6)\text{constraint}) \\
 x_3 &\geq x_2 + 0 && (\text{Arc } (2, 3)\text{constraint}) \\
 x_6 - x_1 &\leq 25, A, B, C, D, E, F \geq 0, x_j, \text{urs}
 \end{aligned}$$

The first six constraints stipulate that the duration of each activity can be reduced by at most 5 days. As before, the next seven constraints ensure that event  $j$  cannot occur until after node  $i$  occurs and activity  $(i, j)$  is completed. For example, activity  $B(\text{arc } (1,2))$  now has a duration of  $9 - B$ . Thus, we need the constraint  $x_2 \geq x_1 + (9 - B)$ . The constraint  $x_6 - x_1 \leq 25$  ensures that the project is completed within the 25-day deadline. The objective function is the total cost incurred in reducing the duration of the activities. An optimal solution to this LP is  $z = \$390$ ,  $x_1 = 0$ ,  $x_2 = 4$ ,  $x_3 = 4$ ,  $x_4 = 6$ ,  $x_5 = 13$ ,  $x_6 = 25$ ,  $A = 2$ ,  $B = 5$ ,  $C = 0$ ,  $D = 5$ ,  $E = 3$ ,  $F = 0$ . After reducing the durations of projects B, A, D, and E by the given amounts, we obtain the project network pictured in Figure 35. The reader should verify that A, B, D, E, and F are critical activities and that 1–2–3–4–5–6 and 1–3–4–5–6 are both critical paths (each having length 25). Thus, the project deadline of 25 days can be met for a cost of \$390.

## 7.8. PERT: Program Evaluation and Review Technique

CPM assumes that the duration of each activity is known with certainty. For many projects, this is clearly not applicable. PERT is an attempt to correct this shortcoming of CPM by modeling the duration of each

activity as a random variable. For each activity, PERT requires that the project manager estimate the following three quantities:

$a$  = estimate of the activity's duration under the most favorable conditions

$b$  = estimate of the activity's duration under the least favorable conditions

$m$  = most likely value for the activity's duration

Let  $T_{ij}$  (random variables are printed in boldface) be the duration of activity  $(i,j)$ . PERT requires the assumption that  $T_{ij}$  follows a beta distribution. The specific definition of a beta distribution need not concern us, but it is important to realize that it can approximate a wide range of random variables, including many positively skewed, negatively skewed, and symmetric random variables. If  $T_{ij}$  follows a beta distribution, then it can be shown that the mean and variance of  $T_{ij}$  may be approximated by

$$E(T_{ij}) = \frac{a + 4m + b}{6}$$

$$\text{var } T_{ij} = \frac{(b - a)^2}{36}$$

(7.4) and (7.5)

PERT requires the assumption that the durations of all activities are independent. Then for any path in the project network, the mean and variance of the time required to complete the activities on the path are given by

$$\sum_{(i,j) \in \text{path}} E(T_{ij}) = \text{expected duration of activities on any path}$$

$$\sum_{(i,j) \in \text{path}} \text{var } T_{ij} = \text{variance of duration of activities on any path}$$

(7.6) and (7.7)

Let  $\mathbf{CP}$  be the random variable denoting the total duration of the activities on a critical path found by CPM. PERT assumes that the critical path found by CPM contains enough activities to allow us to invoke the Central Limit Theorem and conclude that

$$\mathbf{CP} = (i, j) \in \text{critical path} \text{var } T_{ij}$$

is normally distributed. With this assumption, (7.4)–(7.7) can be used to answer questions concerning the probability that the project will be completed by a given date. For example, suppose that for Example 1,  $a$ ,  $b$ , and  $m$  for each activity are shown in Table 4. Now (7.4) and (7.5) yield

$$\begin{aligned} E(T_{12}) &= \frac{\{5 + 13 + 36\}}{6} = 9 & \text{var } T_{12} &= \frac{(13 - 5)^2}{36} = 1.78 \\ E(T_{13}) &= \frac{\{2 + 10 + 24\}}{6} = 6 & \text{var } T_{13} &= \frac{(10 - 2)^2}{36} = 1.78 \\ E(T_{35}) &= \frac{\{3 + 13 + 32\}}{6} = 8 & \text{var } T_{35} &= \frac{(13 - 3)^2}{36} = 2.78 \\ E(T_{34}) &= \frac{\{1 + 13 + 28\}}{6} = 7 & \text{var } T_{34} &= \frac{(13 - 1)^2}{36} = 4 \\ E(T_{45}) &= \frac{\{8 + 12 + 40\}}{6} = 10 & \text{var } T_{45} &= \frac{(12 - 8)^2}{36} = 0.44 \\ E(T_{56}) &= \frac{\{9 + 15 + 48\}}{6} = 12 & \text{var } T_{56} &= \frac{(15 - 9)^2}{36} = 1 \end{aligned}$$

Of course, the fact that arc  $(2, 3)$  is a dummy arc yields

$$E(T_{23}) = \text{var } T_{23} = 0$$

Recall that the critical path for Example 1 was 1–2–3–4–5–6. From Equations (7.6) and (7.7),

$$\begin{aligned} E(\mathbf{CP}) &= 9 + 0 + 7 + 10 + 12 = 38 & \text{var } \mathbf{CP} \\ &= 1.78 + 0 + 4 + 0.44 + 1 = 7.22 \end{aligned}$$

Then the standard deviation for  $\mathbf{CP}$  is  $(7.22)^{\frac{1}{2}} = 2.69$ .

**TABLE 4.**  $a$ ,  $b$ , and  $m$  for Activities in Widgeto

| Activity | $a$ | $b$ | $m$ |
|----------|-----|-----|-----|
| (1, 2)   | 5   | 13  | 9   |

|        |   |    |    |
|--------|---|----|----|
| (1, 3) | 2 | 10 | 6  |
| (3, 5) | 3 | 13 | 8  |
| (3, 4) | 1 | 13 | 7  |
| (4, 5) | 8 | 12 | 10 |
| (5, 6) | 9 | 15 | 12 |

Applying the assumption that **CP** is normally distributed, we can answer questions such as the following: What is the probability that the project will be completed within 35 days? To answer this question, we must also make the following assumption: No matter what the durations of the project's activities turn out to be, 1–2–3–4–5–6 will be a critical path. This assumption implies that the probability that the project will be completed within 35 days is just  $P(\text{CP} < 35)$ . Standardizing and applying the assumption that **CP** is normally distributed, we find that **Z** is a standardized normal random variable with mean 0 and variance 1. The cumulative distribution function for a normal random variable is tabulated in Table 16. For example,  $P(Z < -1) = 0.1587$  and  $P(Z < 2) = 0.9772$ . Thus,

$$P(\text{CP} \leq 35) = P\left(\frac{\text{CP} - 38}{2.69} \leq \frac{35 - 38}{2.69}\right) = P(Z \leq -1.12) = 0.13$$

where  $F(-1.12) = 0.13$  may be obtained using the NORMSDIST function in Excel. Entering the formula =NORMSDIST( $x$ ) returns the probability that a standard normal random variable with mean 0 and standard deviation 1 is less than or equal to  $x$ . For example =NORMDIST(-1.12) yields .1313.

## 7.9. Difficulties with PERT

There are several difficulties with PERT:

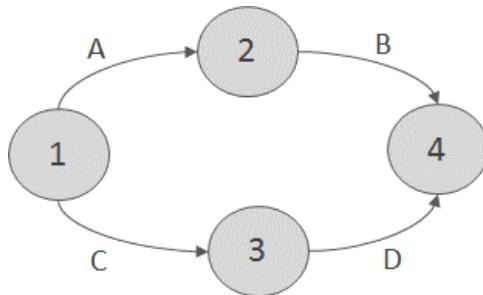
1. The assumption that the activity durations are independent is difficult to justify.
2. Activity durations may not follow a beta distribution.
3. The assumption that the critical path found by CPM will always be the critical path for the project may not be justified.

The last difficulty is the most serious. For example, in our analysis of Example 1, we assumed that 1–2–3–4–5–6 would always be the critical path. If, however, activity A were significantly delayed and activity B were completed ahead of schedule, then the critical path might be 1–3–4–5–6.

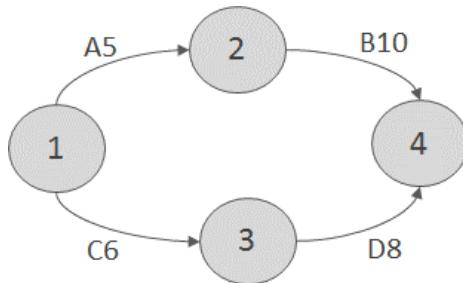
Here is a more concrete example of the fact that (because of the uncertain duration of activities) the critical path found by CPM may not actually be the path that determines the completion date of the project. Consider the simple project network in Figure 9. Assume that for each activity in Table 16,  $a$ ,  $b$ , and  $m$  each occur with probability  $\frac{1}{3}$ . If CPM were applied (using the expected duration of each activity as the duration of the activity), then we would obtain the network in Figure 10. For this network, the critical path is 1–2–4. In actuality, however, the critical path could be 1–3–4. For example, if the optimistic duration of B (6 days) occurred and all other activities had a duration  $m$ , then 1–3–4 would be the critical path in the network. If we assume that the durations of the four activities are independent random variables, then using elementary probability, it can be shown that there is a  $\frac{10}{27}$  probability that 1–3–4 is the critical path, a  $\frac{15}{27}$  chance that 1–2–4 is the critical path, and a  $\frac{2}{27}$  chance that 1–2–4 and 1–3–4 will both be critical paths. This example shows that one must be cautious in designating an activity as critical. In this situation, the probability that each activity is actually a critical activity is shown in Table 6.

**TABLE 5.**  $a$ ,  $b$ , and  $m$  for Figure 37

| Activity | $a$ | $b$ | $m$ |
|----------|-----|-----|-----|
| A        | 1   | 9   | 5   |
| B        | 6   | 14  | 10  |
| C        | 5   | 7   | 6   |
| D        | 7   | 9   | 8   |



**FIGURE 9.** Project Network to Illustrate Difficulties with PERT



**FIGURE 10.** Network to Determine Critical Path If Each Activity's Duration Equals  $m$

**TABLE 6.** Probability That Each Arc is on a Critical Path

| Activity | Probability |
|----------|-------------|
| A        | 17/27       |
| B        | 17/27       |
| C        | 12/27       |
| D        | 12/27       |

When the duration of activities is uncertain, the best way to analyze a project is to use a Monte Carlo simulation @Risk add-in for Excel. In Chapter 18, we will mention @Risk again. With @Risk, we can easily determine the probability that a project is completed on time and determine the probability that each activity is critical.

# 8. What is Integer Linear Programming?

In this chapter we lift the **divisibility assumption** of LPs. This means that we are looking at LPs where some (or all) of the decision variables are required to be integers.

## 8.1. Mixed Integer Linear Programming Terminology

Although the name integer program does not state it explicitly, it is assumed that Integer and mixed integer programs are LPs with the additional requirement that all (or some) of the decision variables are integers.

If the additional requirement that some of the decision variables are integers is lifted, then the resulting LP is called the **LP relaxation** of the program in question.

### 8.1.1. Pure Integer Program

An IP in which **all** the decision variables are required to be integers is called a **pure integer program**, or simply an **integer program (IP)**.

#### EXAMPLE 1. Pure Integer Program

$$\max z = 3x_1 + 2x_2 \quad \text{LP (8.1a)}$$

s.t.

$$\begin{aligned} x_1 + x_2 &\leq 6 \\ x_1, x_2 &\geq 0, \text{integer} \end{aligned}$$

is a pure integer program.

### 8.1.2. Binary Integer Program

A **binary integer program (BIP)** is a IP where all the decision variables are required to be either 0 or 1.

$$\max z = 3.2x_1 + 2.4x_2 + 1.5x_3 \quad \text{LP (8.1b)}$$

s.t.

$$0.1x_1 + 0.9x_2 + 0.5x_3 \leq 2.5$$

$$-1.7x_2 - 9.5x_3 \geq 1.5$$

$$x_1, x_2, x_3 = 0 \text{ or } 1.$$

### EXAMPLE 2. Assignment

Assignment problems are BIPs, although they can be solved as LPs.

Any BIP can be transformed into an IP by simply noting that the constraint

$$x_i = 0 \text{ or } 1$$

is the same as the constraints

$$\begin{aligned} x_i &\leq 1 \\ x_i &\geq 0, \text{ integer} \end{aligned}$$

### 8.1.3. Mixed Integer Program

An LP in which **some but not all** of the decision variables are required to be integers is called a **mixed integer linear program** (MILP). Sometimes the name mixed integer program (MIP) is also used.

### EXAMPLE 3. Mixed Integer Program

$$\max z = 3x_1 + 2x_2 \quad \text{LP (8.1c)}$$

s.t.

$$\begin{aligned} x_1 + x_2 &\leq 6 \\ x_1, x_2 &\geq 0, x_1 \text{ integer} \end{aligned}$$

is a MILP, since  $x_1$  is required to be an integer, but  $x_2$  is not.

**DEFINITION** ■ The LP obtained by omitting all integer or 0–1 constraints on variables is called the **LP relaxation** of the IP. ■

### 8.2. Branch-And-Bound Method

In this section we provide a relatively fast algorithm for solving IPs and MILPs. The general idea of the algorithm is to solve LP relaxations of the

IP or MILP and to look for an integer solution by branching and bounding on the decision variables provided by the LP relaxations.

#### **EXAMPLE 4.** Branch-And-Bound (Furniture with Honesty)

The Honesty Furniture Corporation manufactures tables and chairs. A table requires 1 hour of labor and 9 units of wood. A chair requires 1 hour of labor and 5 units of wood. Currently 6 hours of labor and 45 units of wood are available. Each table contributes \$8 to profit, and each chair contributes \$5 to profit.

Honesty Furniture's want to maximize its profit.

This problem would be a classical product selection problem, but we insist that only integral furniture is produced. So, the problem is no longer an LP but an IP.

Let us formulate and solve the (pure) IP for Example 4. Let

$x_1$  = number of tables manufactured;

$x_2$  = number of chairs manufactured:

Since  $x_1$  and  $x_2$  must be integers, Honesty Furniture wishes to solve the following (pure) IP:

$$\max z = 8x_1 + 5x_2 \quad \text{IP (8.2)}$$

s.t.

$$x_1 + x_2 \leq 6 \quad (\text{labor})$$

$$9x_1 + 5x_2 \leq 45 \quad (\text{wood})$$

$$x_1, x_2 \geq 0; \text{ integer}$$

The first step in the branch-and-bound method is to solve the LP relaxation of the IP:

$$\max z = 8x_1 + 5x_2 \quad \text{LP Relaxation (8.2)}$$

s.t.

$$x_1 + x_2 \leq 6 \quad (\text{labor})$$

$$9x_1 + 5x_2 \leq 45 \quad (\text{wood})$$

$$x_1; x_2 \geq 0$$

If we are lucky, all the decision variables ( $x_1$  for tables and  $x_2$  for chairs in the example we are considering) in the LP relaxation's optimum turn out to be integers. In this lucky case the optimal solution of the LP relaxation is also the optimal solution to the original IP.

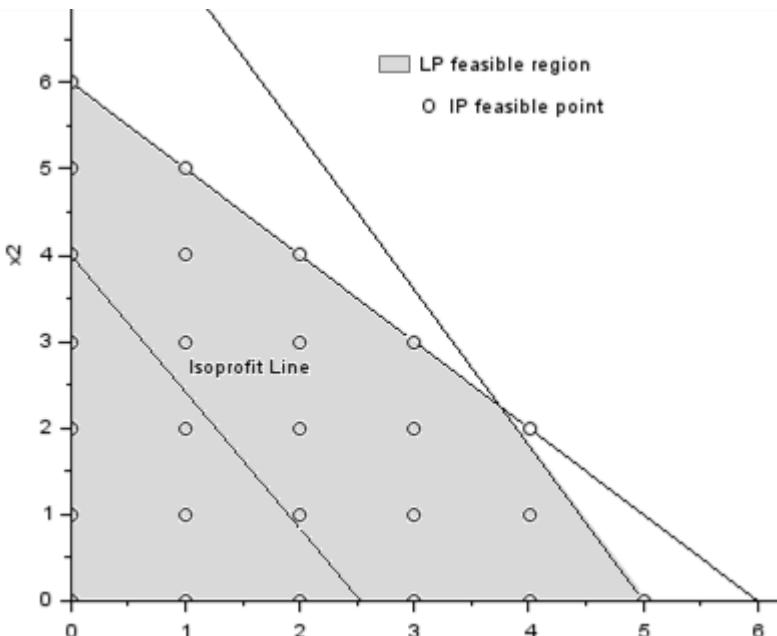
In the branch-and-bound algorithm we use next, what we call the LP relaxation of the IP (8.2) or sub-problem 1 (SP 1). After solving SP 1 we find the solution to be

$$z = 165 = 4$$

$$x_1 = 15 = 4$$

$$x_2 = 9 = 4$$

This means that we were not lucky: the decision variables turned out to be fractional. So, the LP relaxation SP 1 has (possibly) a better optimum than the original IP (8.2). In any case, we have found an upper bound to the original IP: Honesty Furniture Corporation of Example 4 cannot possibly have better profit than \$165=4.



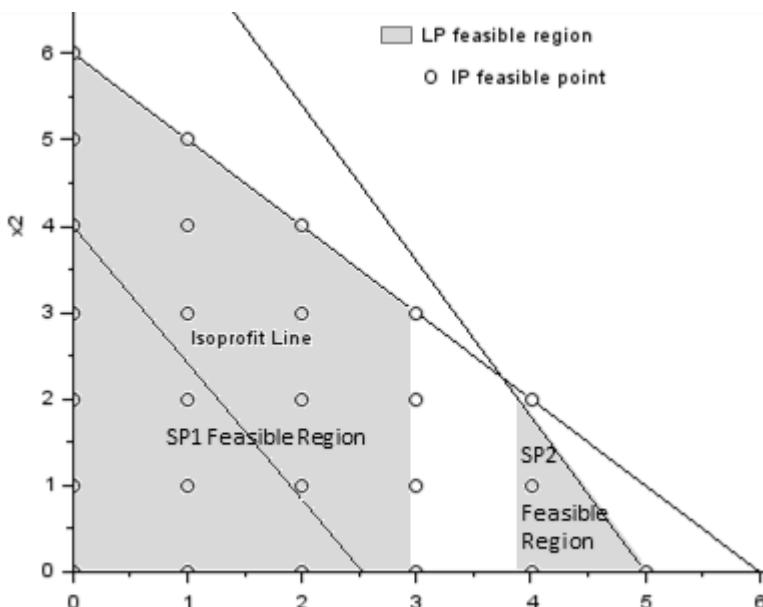
**FIGURE 1.** LP Relaxation (8.2) Feasible Region (made with SCILAB)

Next we split the feasible region (painted light grey in Figure 1) of the LP relaxation (8.2) in hope to find a solution that is an integer one. We arbitrarily choose a variable that is fractional at the optimal solution of the LP SP 1 (the first LP relaxation). We choose  $x_1$ . Now,  $x_1$  was  $15/4=3.75$ . Obviously, at the optimal solution to the IP we have either  $x_1 \geq 4$  or  $x_1 \leq 3$ , since the third alternative  $3 < x_1 < 4$  is out of the question for IPs. So, we consider the two possible cases  $x_1 \geq 4$  and  $x_1 \leq 3$  as separate sub-problems. We denote these sub-problems as SP 2 and SP 3. So,

$$SP\ 2 = SP\ 1 + "x_1 \geq 4"$$

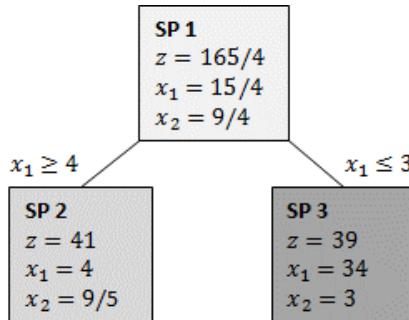
$$SP\ 3 = SP\ 1 + "x_1 \leq 3"$$

In Figure 2 we see that every possible feasible solution of the Honesty Furniture's IP (8.2) (the bullet points) is included in the feasible region of either SP 2 or SP 3. Since SP 2 and SP 3 were created by adding constraints involving the fractional solution  $x_1$ , we say that SP 2 and SP 3 were created by **branching** on  $x_1$ .



**FIGURE 2.** SP 2 and SP 3 Feasible Regions (made with SCILAB)

Below in Figure 3, we have a tree-representation of the Figure 2 above, and of our branching-and-bounding so far. The color coding refer to the two previous figures.



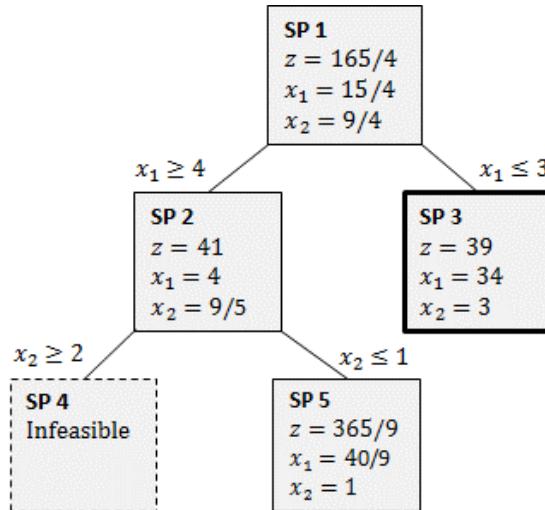
**FIGURE 3.** First Branch-and-Bound

We see that SP 3 has an integer solution. So, we do not have to branch SP 2 anymore. Unfortunately, the integer solution of SP 3,  $z = 39$ , is suboptimal when compared to the non-integer solution,  $z = 41$ , of SP 2. So, it may turn out that the SP 2 has a further sub-problem that has better integer solution than SP 3. So, we have to branch SP 2 to find out what happens. Since  $x_1 = 4$  is an integer, we branch on  $x_2 = 9 = 5 = 1.8$ . So, we have the new sub-problems of SP 2:

$$SP\ 4 = SP\ 2 + x_2 \geq 2;$$

$$SP\ 5 = SP\ 2 + x_2 \leq 1;$$

When the solutions to these sub-problems are added to the tree above we get the following tree in Figure 4. There is no color coding in the boxes in the tree. There is border coding, however. A thick borderline expresses a feasible IP solution, and a dashing borderline expresses an infeasible case (the color coding is dropped):



**FIGURE 4.** Second Branch-and-Bound

We see that SP 4 is infeasible. So, the optimal solution is not there. However, SP 5 gives us a non-integer solution that is better than the integer solution of SP 3. So, we have to branch on SP 5. Since  $x_2 = 1$  is already an integer, we branch on  $x_1 = 40 = 9 = 4: 444$ . So, we get two new sub-problems of SP 5:

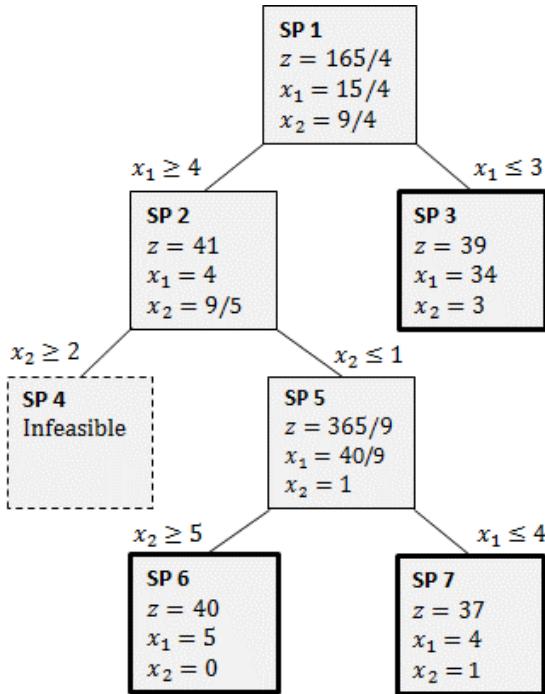
$$SP\ 6 = SP\ 5 + x_1 \geq 5,$$

$$SP\ 7 = SP\ 5 + x_1 \leq 4.$$

After this branching we finally arrive at the final solution where all the sub-problems are either infeasible, have integer solutions, or are suboptimal to some sub-problem that have integer solution, i.e. we have integer bounds for the sub-problems. This is the bound part of the branch-and-bound.

From the tree in Figure 5 above, we can read the solution to the IP. The SP 6 is the optimal sub-problem with integer solution. So, the solution to the IP (8.2.2) is

$$\begin{aligned} z &= 40 \\ x_1 &= 5 \\ x_2 &= 0 \end{aligned}$$



**FIGURE 5.** Third Branch-and-Bound

### 8.2.1. General Branch-And-Bound Algorithm

We have solved a pure IP with branch and bound. To solve MILP with branch-and-bound one follows the same steps as in the pure IP case except one only branches on decision variables that are required to be integers. So, solving MILPs is actually somewhat easier than solving pure IPs!

**Algorithm 8.1.** (Branch-and-bound algorithm).

**Step 1.** Solve the LP relaxation of the problem. If the solution is integer where required, then we are done. Otherwise create two new sub-problems by branching any fractional variable that is required to be integer.

**Step 2.** A sub-problem is not active when any of the following occurs:

- You used the sub-problem to branch on, i.e., the sub-problem is not a leaf in the tree.

- a. All variables in the solution that are required to be integers, are integers.
- b. The sub-problem is infeasible.
- c. You can fathom the sub-problem by a bounding argument.

Choose an active sub-problem and branch on a fractional variable that should be integer in the final solution. Repeat until there are no active sub-problems.

**Step 3.** Solution to the IP/MILP is the best IP/MILP solution of the sub-problems you have created. It is found in one of the leaves of the tree representing the sub-problems.

### 8.2.2. Problems

1. Solve the optimization problems (8.1a), (8.1b), and (8.1c) and their corresponding LP relaxations.
2. How can MILP be used to ensure that a variable,  $x_1$ , say, can assume only values 1, 2, 3, 4?
3. How can MILP be used to ensure that a variable,  $x_1$ , say, can assume only values from a given finite set of values?
4. Make an *SCILAB* function

```
[zmax, xmax, status] = rLPSolver(c,A,b, rvar, rvarsets)
```

(restricted LP solver) that solves a given standard form LP, given by the input variables  $c$ ,  $A$ , and  $b$ , with the additional restrictions that the decision variables indicated by the input variable  $rvar$  belong to the sets defined by the input variable  $rvarsets$ .

5. \* Implement the branch-and-bound algorithm with *SCILAB*.

### 8.3. Traveling Salesman Problem

MILPs can be used to model and solve many combinatorial problems. Arguably, the most famous, or notorious, combinatorial problem is the **traveling salesman problem**. Many ways of solving it are known. Here we present the MILP way. Unfortunately, no-one knows any fast ways to solve the traveling salesman problem.

In the **traveling salesman problem** one has to find a tour around  $N$  points so that the total distance traveled is as short as possible.

An example of the traveling salesman problem is:

**EXAMPLE 5.** (Laurie travels).

Laurie the Traveler wants to take a Kyiv–Paris–Rome–Vaasa tour (in some order). The distances (in kilometers) between the cities are

| City  | Kyiv  | Paris | Rome  | Vaasa |
|-------|-------|-------|-------|-------|
| City  | 0     | 2,023 | 1,674 | 1,497 |
| Kyiv  |       | 0     | 1,105 | 1,967 |
| Paris | 2,023 | 0     | 1,105 | 1,967 |
| Rome  | 1,674 | 1,105 | 0     | 2,429 |
| Vaasa | 1,497 | 1,967 | 2,429 | 0     |

In which order should Laurie take her tour?

An obvious brute force method to solve the traveling salesman problem is to check all the possible routes and pick the shortest one. The obvious problem with this obvious method is that with  $N$  cities we have to check  $N!$  routes. For example, with 20 cities we would have  $20! = 2,432,902,008,176,640,000$  routes to check. So we should check over 2 quintillion routes if we are Americans. The Europeans have to check over 2 trillion routes G. This is our old Nemesis, the **combinatorial curse**.

Here is a MILP formulation of the traveling salesman problem. It will work much faster than the brute force method. (I wouldn't call it fast, though.)

First we consider the **data** associated with the problem. The data is the cities and their distances. Let  $c_{ij}$  be the distance between cities  $i$  and  $j$ . We can generalize the problem right away by **not** assuming that  $c_{ij} = c_{ji}$ . So, can we allow that, for some strange reason, the distance from the city  $i$  to the city  $j$  can be different than the distance from the city  $j$  to the city  $i$ . However, the distance from the city  $i$  to itself is set to be a very, very, very, very big number  $M_i$ :  $c_{ii} = M_i$ . This may look silly

(shouldn't it be 0 ?), but this is to ensure that the traveling salesman does not go to the city  $i$  immediately after visiting the same city  $i$  (that would be silly).

Now the data of the problem is formulated as the matrix  $C = [c_{ij}]$ . Then we can start the LP, or MILP, modeling by using Algorithm 1.1.2.

First, we have to find the **decision variables**. We use indicator decisions:

$$x_{ij} = \begin{cases} 1, & \text{if the salesman goes from the city } i \text{ to the city } j \\ 0, & \text{otherwise} \end{cases}$$

Second, we have to find the **objective function**. But this is to minimize the total traveling distance:

$$\min z = \sum_i \sum_j c_{ij} x_{ij}$$

Third, we have to find the **constraints**. Now, we have to arrive once at every city. So, for all cities  $j$  we must have

$$\sum_i x_{ij} = 1$$

Also, we must leave every city  $i$  once. So, for all cities  $i$  we must have

$$\sum_j x_{ij} = 1$$

Finally, there is a very tricky set of constraints. We must visit **all** the cities by a single continuous path going from one city to another. This is ensured by the constraints: For all  $i, j \geq 2$ ,  $i \neq j$ , and for some  $u_i, u_j \geq 0$ ,  $u_i - u_j + Nx_{ij} \leq N - 1$ . These constraints are a bit difficult to understand. What they do is that they prevent subtours, i.e., tours that form a loop that do not contain all the cities. Indeed, suppose that there is a subtour starting from the city 1, say, that does not visit all the cities. Then there must be another subtour that does not visit the city 1. Then the number of cities of this subtour is  $R < N$ . If we now add together the constraint above corresponding to the  $x_{ij}$ 's of the latter subtour we get  $NR \leq (N - 1)R$ , since all the  $u_i$ 's cancel. So, the constraint set is violated. On the other hand, interpreting the variables  $u_i$  as the first time

the city  $i$  is visited, we see (at least after some tedious case-checking) that a complete tour will satisfy the constraints above.

Putting what we have found out together we find that the traveling salesman problem with  $N$  cities is the MILP

$$\min z = \sum_i \sum_j c_{ij} x_{ij},$$

s.t. (IP 8.3)

$$\sum_i x_{ij} = 1, \text{ for all } j$$

$$\sum_j x_{ij} = 1, \text{ for all } i$$

$$u_i - u_j + Nx_{ij} \leq N - 1, \text{ for all } i \neq j \geq 2$$

$$x_{ij} = 0 \text{ or } 1, \text{ for all } i \text{ and } j$$

$$u_i \geq 0, \text{ for all } i \geq 2$$

Note that, in addition to the actual decision variables  $X_{ij}$  determining the optimal route, we have the auxiliary variables  $u_i$  also.

The following script file solves Laurie's problem (Example 5) with *SCILAB*'s karmarkar. It should also be useful in solving any traveling salesman problem with *SCILAB*. You might want to copy–paste it to your editor and save it as *Laurie\_travels.m*. To make the copy–pasting easier, we have not included line numbers here. Consequently, we do not explain the script in detail here. Anyway, the in-script comments should help to understand the code.

```
#####
// SCRIPT FILE: Laurie the Traveler: Kyiv, Paris, Rome,
// Vaasa.
#####
// The distance matrix C, and its vector form c augmented
// with u:
M = 10000;
c = [ M 2023 1674 1497;2023 M 1105 1967;1674 1105 M
2429;1497 1967 2429 M];
c=[c(1:4);c(5:8);c(9:12);c(13:16);1;1;1]
```

```

// The first 4*4 columns are for the tour matrix x in the
// vector form
// and the last 3 columns are for the auxiliary vector u:
A = [ 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0;
0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0;
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0;
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0;
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1 0;
0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 1 0 -1;
0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 1 0 -1;
0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 -1 1 0;
0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 1 -1;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 -1 0 1;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 -1 1];
// The bounds and other glpk parameters, and finally call
// glpk:
L = 50; // For u upper bounds.
b = [1; 1; 1; 1; 1; 1; 1; 1; 3; 3; 3; 3; 3; 3];
lb = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]';
ub = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 L L L]';
x0 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
[xopt,lagr,fopt]=linpro(c,A,b,lb,ub,4,x0)
fopt =

```

5204.

```

lagr =

```

- 8503.
- 918.
- 569.
- 0.
- 526.
- 8895.
- 0.
- 470.
- 177.
- 0.
- 8895.
- 932.
- 0.
- 862.
- 1324.
- 8503.
- 1.
- 1.
- 1.
- 1497.
- 1105.

Still, there is another way to solve LPs, using LPSolve (IDE fromSourceForge). LPSolve can be called form within SCILAB, but the initial setup is complicated. With the IDE, we can enter the LPs like we would write them, except each statement is followed by a semicolon. This can be done in an editor, like Notepad++, saved as ASCII with a .lp suffix, and read in the LPSolve IDE or written in the IDE Source (see Figure 6) or Matrix windows.

The screenshot shows the LPSolve IDE interface. The main window is titled "LPSolve IDE - 5.5.2.0 - C:\Users\Strickland\Documents\Operations Research\model4.lp". The menu bar includes File, Edit, Search, Action, View, Options, and Help. Below the menu is a toolbar with various icons. The main area has tabs for Source, Matrix, Options, and Result. The Source tab is active, displaying the following LP code:

```

1 /* model.lp */
2
3 min: 10000x1+2023x2+1674x3+1497x4+2023x5+10000x6+1105x7+1967x8+1674x9+11
4
5 x1+x5+x9+x13=1;
6 x2+x6+x10+x14=1;
7 x3+x7+x11+x15=1;
8 x4+x8+x12+x16=1;
9 x1+x2+x3+x4=1;
10 x5+x6+x7+x8=1;
11 x9+x10+x11+x12=1;
12 x13+x14+x15+x16=1;
13 4x7+x17-x18<=3;
14 4x8+x17-x19<=3;
15 4x10-x17+x18<=3;
16 4x12+x18-x19<=3;
17 4x14-x17+x19<=3;
18 4x15-x18+x19<=3;
19 0<=x1<=1;

```

Below the code is a Log window containing solver statistics:

```

MEMO: lp_solve version 5.5.2.0 for 32 bit OS, with 64 bit REAL variables.
In the total iteration count 22, 0 (0.0%) were bound flips.
There were 4 refactorizations, 0 triggered by time and 0 by density.
... on average 5.5 major pivots per refactorization.
The largest [LUSOL v2.2.1.0] fact(B) had 45 NZ entries, 1.2x largest basis.
The maximum B&B level was 5, 0.2x MIP order, 2 at the optimal solution.
The constraint matrix inf-norm is 4, with a dynamic range of 4.
Time to load data was 0.011 seconds, presolve used 0.009 seconds,
... 0.073 seconds in simplex solver, in total 0.093 seconds.

```

At the bottom, status bars show: 71:18, ITE: 21, INV: 10, NOD: 7, TME: 0.03.

**Figure 6.** LPSolve IDE showing Source window

Section 8.8 provides information on how to download and install LPSolve. LPSolve can be run from MATLAB, R, Octave and other programs.

The LP is entered as shown below. I wrote this in Notepad++, saved it as model.lp (ASCII), and read it in the LPSolve IDE, e.g., File | Open | drive/path/model.lp.

To start with I am assuming that we do not know which constraints are binding

```

/* model.lp */

min:
10000x1+2023x2+1674x3+1497x4+2023x5+10000x6+1105x7+1967x8+11

```

```

674x9+1105x10+10000x11+2429x12+1497x13+1967x14+2429x15+1000
0x16+0x17+0x18+0x19;

x1+x5+x9+x13=1;
x2+x6+x10+x14=1;
x3+x7+x11+x15=1;
x4+x8+x12+x16=1;
x1+x2+x3+x4=1;
x5+x6+x7+x8=1;
x9+x10+x11+x12=1;
x13+x14+x15+x16=1;
4x7+x17-x18<=3;
4x8+x17-x19<=3;
4x10-x17+x18<=3;
4x12+x18-x19<=3;
4x14-x17+x19<=3;
4x15-x18+x19<=3;
0<=x1<=1;
0<=x2<=1;
0<=x3<=1;
0<=x4<=1;
0<=x5<=1;
0<=x6<=1;
0<=x7<=1;
0<=x8<=1;
0<=x9<=1;
0<=x10<=1;
0<=x11<=1;
0<=x12<=1;
0<=x13<=1;
0<=x14<=1;
0<=x15<=1;
0<=x16<=1;
0<=x17<=50;
0<=x18<=50;
0<=x19<=50;
int x1;
int x2;
int x3;
int x4;
int x5;
int x6;
int x7;
int x8;
int x9;
int x10;
int x11;
int x12;
int x13;
int x14;
int x15;

```

```
int x16;
```

The solution is found in the Result tab, and then the Variables, MILP Feasible and results tabs. The variable results are shown in Figure 7, giving the total tour cost of 6243, as we found in *SCILAB*.

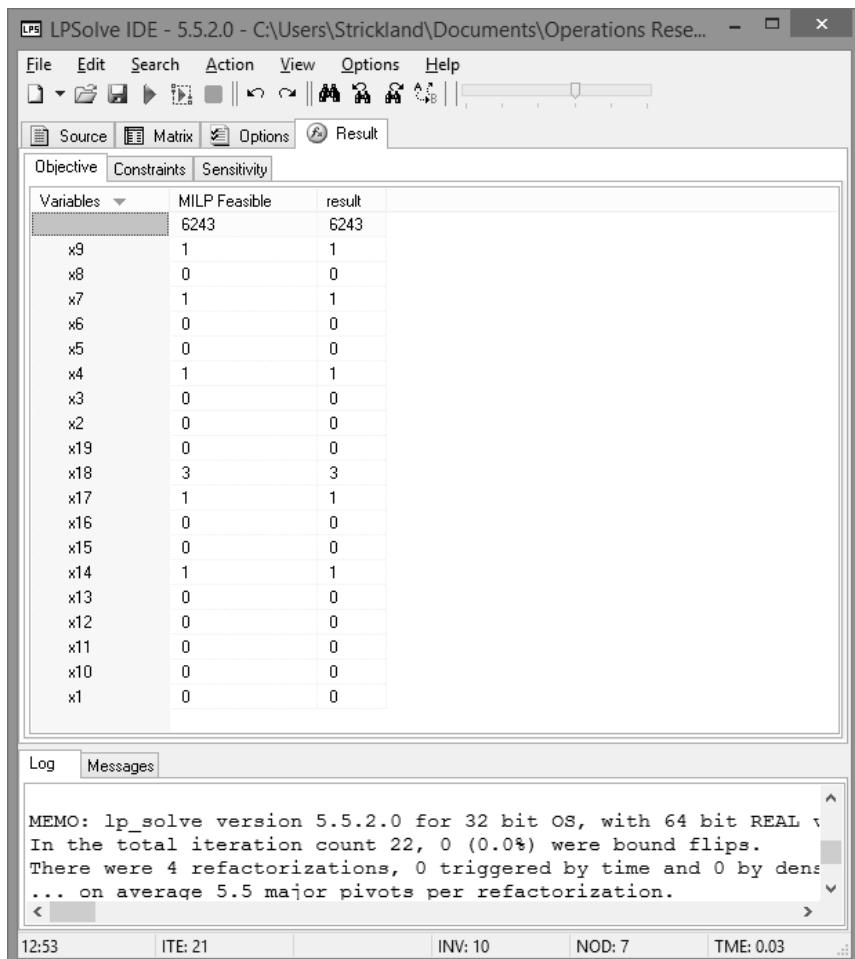


Figure 7.LPSolve Results window

| Variables | MILP | Feasible | result |
|-----------|------|----------|--------|
|           | 5204 |          | 5204   |
| x9        | 1    |          | 1      |
| x8        | 0    |          | 0      |

|     |   |   |
|-----|---|---|
| x7  | 1 | 1 |
| x6  | 0 | 0 |
| x5  | 0 | 0 |
| x4  | 1 | 1 |
| x3  | 0 | 0 |
| x2  | 0 | 0 |
| x19 | 0 | 0 |
| x18 | 3 | 3 |
| x17 | 1 | 1 |
| x16 | 0 | 0 |
| x15 | 0 | 0 |
| x14 | 1 | 1 |
| x13 | 0 | 0 |
| x12 | 0 | 0 |
| x11 | 0 | 0 |
| x10 | 0 | 0 |
| x1  | 0 | 0 |

## 8.4. Fixed-Charge Problems

Example 6 illustrates an important trick that can be used to formulate many location and production problems as IPs.

### EXAMPLE 6. Fixed-Charge IP

Mandella Cloth Company is capable of manufacturing three types of clothing: shirts, shorts, and pants. The manufacture of each type of clothing requires that Mandella have the appropriate type of machinery available. The machinery needed to manufacture each type of clothing must be rented at the following rates: shirt machinery, \$200 per week; shorts machinery, \$150 per week; pants machinery, \$100 per week. The manufacture of each type of clothing also requires the amounts of cloth and labor shown in Table 2. Each week,

150 hours of labor and 160 sq yd of cloth are available. The variable unit cost and selling price for each type of clothing are shown in Table 3. Formulate an IP whose solution will maximize Mandella's weekly profits.

**Solution.** As in LP formulations, we define a decision variable for each decision that Mandella must make. Clearly, Mandella must decide how many of each type of clothing should be manufactured each week, so we define

$x_1$  = number of shirts produced each week

$x_2$  = number of shorts produced each week  $x_3$  = number of pants produced each week

**TABLE 2.** Resource Requirements for Mandella

| Clothing Type | Labor<br>(Hours) | Cloth<br>(Square Yards) |
|---------------|------------------|-------------------------|
| Shirt         | 3                | 4                       |
| Shorts        | 2                | 3                       |
| Pants         | 6                | 4                       |

**TABLE 3.** Revenue and Cost Information for Mandella

| Clothing Type | Sales Price (\$) | Variable Cost (\$) |
|---------------|------------------|--------------------|
| Shirt         | 12               | 6                  |
| Shorts        | 18               | 4                  |
| Pants         | 15               | 8                  |

Note that the cost of renting machinery depends only on the types of clothing produced, not on the amount of each type of clothing. This enables us to express the cost of renting machinery by using the following variables:

$$y_1 = \begin{cases} 1, & \text{if any shirts are manufactured} \\ 0, & \text{otherwise} \end{cases}$$

$$y_2 = \begin{cases} 1, & \text{if any shorts are manufactured} \\ 0, & \text{otherwise} \end{cases}$$

$$y_3 = \begin{cases} 1, & \text{if any pants are manufactured} \\ 0, & \text{otherwise} \end{cases}$$

In short, if  $x_j > 0$ , then  $y_j = 1$ , and if  $x_j = 0$ , then  $y_j = 0$ . Thus, Mandella's weekly profits = (weekly sales revenue) - (weekly variable costs) - (weekly costs of renting machinery). Also,

$$\text{Weekly cost of renting machinery} = 200y_1 + 150y_2 + 100y_3 \quad (8.3)$$

To justify (8.3), note that it picks up the rental costs only for the machines needed to manufacture those products that Mandella is actually manufacturing. For example, suppose that shirts and pants are manufactured. Then  $y_1 = y_3 = 1$  and  $y_2 = 0$ , and the total weekly rental cost will be  $200 + 100 = \$300$ .

Because the cost of renting, say, shirt machinery does not depend on the number of shirts produced, the cost of renting each type of machinery is called a fixed charge. A fixed charge for an activity is a cost that is assessed whenever the activity is undertaken at a nonzero level. The presence of fixed charges will make the formulation of the Mandella problem much more difficult.

We can now express Mandella's weekly profits as

$$\begin{aligned}\text{Weekly profit} &= (12x_1 + 8x_2 + 15x_3) - (6x_1 + 4x_2 + 8x_3) \\ &\quad - (200y_1 + 150y_2 + 100y_3) \\ &= 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3\end{aligned}$$

Thus, Mandella wants to maximize

$$z = 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$$

Because its supply of labor and cloth is limited, Mandella faces the following two constraints:

- Constraint 1:** At most, 150 hours of labor can be used each week.  
**Constraint 2:** At most, 160 sq. yd. of cloth can be used each week.

Constraint 1 is expressed by

$$3x_1 + 2x_2 + 6x_3 \leq 150 \quad (\text{Labor constraint}) \quad (8.4)$$

Constraint 2 is expressed by

$$4x_1 + 3x_2 + 4x_3 \leq 160 \quad (\text{Cloth constraint}) \quad (8.5)$$

Observe that  $x_j > 0$  and  $x_j$  integer ( $j = 1, 2, 3$ ) must hold along with  $y_j = 0$  or  $1$  ( $j = 1, 2, 3$ ). Combining (8.4) and (8.5) with these restrictions and the objective function yields the following IP:

$$\max z = 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$$

s.t.

$$3x_1 + 2x_2 + 6x_3 < 150$$

$$4x_1 + 3x_2 + 4x_3 < 160 \quad (\text{IP 4})$$

$$3x_1 + x_2, x_3 > 0; x_1, x_2, x_3 \quad \text{integer}$$

$$3x_1 + y_1, y_2, y_3 = 0 \text{ or } 1$$

The optimal solution to this problem is found to be  $x_1 = 30$ ,  $x_3 = 10$ ,  $x_2 = y_1 = y_2 = y_3 = 0$ . This cannot be the optimal solution to Mandella's problem because it indicates that Mandella can manufacture shirts and pants without incurring the cost of renting the needed machinery. The current formulation is incorrect because the variables  $y_1$ ,  $y_2$ , and  $y_3$  are not present in the constraints. This means that there is nothing to stop us from setting  $y_1 = y_2 = y_3 = 0$ . Setting  $y_1 = 0$  is certainly less costly than setting  $y_i = 1$ , so a minimum-cost solution to (IP 1) will always set  $y_i = 0$ . Somehow we must modify (IP 1) so that whenever  $x_i > 0$ ,  $y_i = 1$  must hold. The following trick will accomplish this goal. Let  $M_1$ ,  $M_2$ , and  $M_3$  be three large positive numbers, and add the following constraints to (IP 1):

$$x_1 < M_1 y_1 \quad (8.6)$$

$$x_2 < M_2 y_2 \quad (8.7)$$

$$x_3 < M_3 y_3 \quad (8.8)$$

Adding (8.6)–(8.8) to IP 1 will ensure that if  $x_i > 0$ , then  $y_i = 1$ . To illustrate, let us show that (16) ensures that if  $x_1 > 0$ , then  $y_1 = 1$ . If  $x_1 > 0$ , then  $y_1$  cannot be 0. For if  $y_1 = 0$ , then (16) would imply  $x_1 < 0$  or  $x_1 = 0$ . Thus, if  $x_1 > 0$ ,  $y_1 = 1$  must hold. If any shirts are produced ( $x_1 > 0$ ), (8.6) ensures that  $y_1 = 1$ , and the objective function will include the cost of the machinery needed to manufacture shirts. Note that if  $y_1 = 1$ , then (8.6) becomes  $x_1 < M_1$ , which does not unnecessarily restrict the value of  $x_1$ . If  $M_1$  were not chosen large, however (say,  $M_1 = 10$ ), then (8.6) would unnecessarily restrict the value of  $x_1$ . In general,  $M_i$  should be set equal to the maximum value that  $x_i$  can attain. In the current problem, at most 40 shirts can be produced (if Mandella produced more than 40 shirts, the company

would run out of cloth), so we can safely choose  $M_1 = 40$ . The reader should verify that we can choose  $M_2 = 53$  and  $M_3 = 25$ .

If  $x_1 = 0$ , (8.6) becomes  $0 < M_1 y_1$ . This allows either  $y_1 = 0$  or  $y_1 = 1$ . Because  $y_1 = 0$  is less costly than  $y_1 = 1$ , the optimal solution will choose  $y_1 = 0$  if  $x_1 = 0$ . In summary, we have shown that if (8.6)–(8.8) are added to (IP 1), then  $x_i > 0$  will imply  $y_i = 1$ , and  $x_i = 0$  will imply  $y_i = 0$ .

The optimal solution to the Mandella problem is  $z = \$75$ ,  $x_3 = 25$ ,  $y_3 = 1$ . Thus, Mandella should produce 25 pants each week.

Now, the IP (IP 4) of Mandella's mass-production Example 6 can be solved, e.g., with the *SCILAB*'s function *kararkar*. In the code we set all the  $M_i$  s to be 100. Of course, 100 is not a very, very, very, very big number, but a very, very, very, very big number may cause problems rounding errors in *SCILAB*, Octave and others. In any case, 100 is big enough.

```

M = 100;
c = [6 4 7 -200 -150 -100]';
A =
 3 2 6 0 0 0;
 4 3 4 0 0 0;
 1 0 0 -M 0 0;
 0 1 0 0 -M 0;
 0 0 1 0 0 -M;
 0 0 0 1 0 0;
 0 0 0 0 1 0;
 0 0 0 0 0 1];
b = [150 160 0 0 0 1 1 1]';
[xopt,fopt,exitflag,iter,yopt]
=karmarkar([],[],c,x0,[],[],[],A,b,lb)
Rank deficient. rank = 8

yopt =
  ineqlin: [8x1 constant]
  eqlin: [0x0 constant]
  lower: [6x1 constant]
  upper: [6x1 constant]
iter =
  1.
exitflag =

```

```
1.  
fopt =  
  
75.  
xopt =  
  
0.  
0.  
25.  
0.  
0.  
1.
```

The same LP is formulated and solves with the same solution in LPSolve (see Figures 8 and 9). Note that in LP solve, we define the problem as an integer program with the int  $x_i$  constraints, much like we do in Excel (Figure 10). We will show you how to set this up in Section 8.8.

The Mandella problem is an example of a fixed-charge problem. In a fixed-charge problem, there is a cost associated with performing an activity at a nonzero level that does not depend on the level of the activity. Thus, in the Mandella problem, if we make any shirts at all (no matter how many we make), we must pay the fixed charge of \$200 to rent a shirt machine. Problems in which a decision maker must choose where to locate facilities are often fixed-charge problems. The decision maker must choose where to locate various facilities (such as plants, warehouses, or business offices), and a fixed charge is often associated with building or operating a facility. Example 7 is a typical location problem involving the idea of a fixed charge.

LPSolve IDE - 5.5.2.0 - C:\Users\Strickland\Documents\O...

File Edit Search Action View Options Help

Source Matrix Options Result

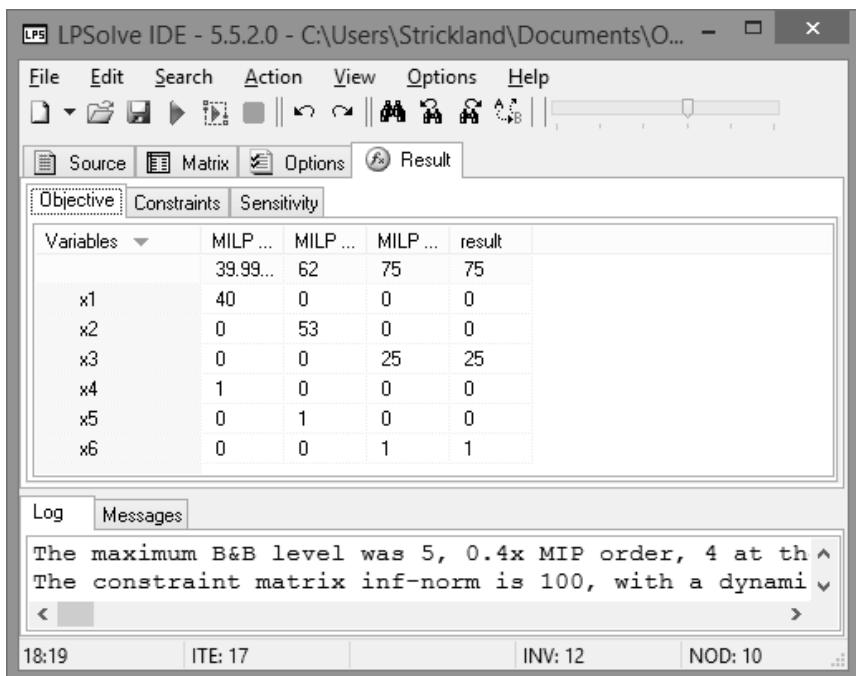
```
1 /* model16.lp */
2
3 max: 6x1+4x2+7x3-200x4-150x5-100x6;
4
5 3x1+2x2+6x3<=150;
6 4x1+3x2+4x3<=160;
7 x1+-100x4<=0;
8 0+x2-100x5<=0;
9 x3-100x6<=0;
10 x4<=1;
11 x5<=1;
12 x6<=1;
13
14 int x1;
15 int x2;
16 int x3;
17 int x4;
18 int x5;
19 int x6;
```

Log Messages

```
The maximum B&B level was 5, 0.4x MIP order, 4 at th^
The constraint matrix inf-norm is 100, with a dynami v
```

18:19 ITE: 17 INV: 12 NOD: 10

Figure 8. Mandella problem in formulated in LPSolve



**Figure 9.** Mandella LP solutionin LPSolve

Microsoft Excel 15.0 Answer Report

**Worksheet:** [model6.lp.csv]model6.lp

**Report Created:** 2/26/2015 1:25:49 AM

**Result:** Solver found a solution. All Constraints and optimality conditions are satisfied.

Solver Engine

Engine: GRG Nonlinear

Solution Time: 0.328 Seconds.

Iterations: 2 Subproblems: 6

Solver Options

Max Time Unlimited, Iterations Unlimited, Precision 0.000001, Use Automatic Scaling

Convergence 0.0001, Population Size 100, Random Seed 0, Derivatives Forward, Require Bounds

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume Nonnegative

#### Objective Cell (Max)

| Cell   | Name    | Original Value | Final Value |
|--------|---------|----------------|-------------|
| \$H\$2 | max LHS | 0              | 75          |

#### Variable Cells

| Cell    | Name        | Original Value | Final Value | Integer |
|---------|-------------|----------------|-------------|---------|
| \$B\$11 | Solution x1 | 0              | 0           | Integer |
| \$C\$11 | Solution x2 | 0              | 0           | Integer |
| \$D\$11 | Solution x3 | 0              | 25          | Integer |
| \$E\$11 | Solution x4 | 0              | 0           | Integer |
| \$F\$11 | Solution x5 | 0              | 0           | Integer |
| \$G\$11 | Solution x6 | 0              | 1           | Integer |

#### Constraints

| Cell    |        |       |                  |             |       |
|---------|--------|-------|------------------|-------------|-------|
| Cell    | Name   | Value | Formula          | Status      | Slack |
| \$H\$3  | R1 LHS | 150   | \$H\$3<=\$J\$3   | Binding     | 0     |
| \$H\$4  | R2 LHS | 100   | \$H\$4<=\$J\$4   | Not Binding | 60    |
| \$H\$5  | R3 LHS | 0     | \$H\$5<=\$J\$5   | Binding     | 0     |
| \$H\$6  | R4 LHS | 0     | \$H\$6<=\$J\$6   | Binding     | 0     |
| \$H\$7  | R5 LHS | -75   | \$H\$7<=\$J\$7   | Not Binding | 75    |
| \$H\$8  | R6 LHS | 0     | \$H\$8<=\$J\$8   | Not Binding | 1     |
| \$H\$9  | R7 LHS | 0     | \$H\$9<=\$J\$9   | Not Binding | 1     |
| \$H\$10 | R8 LHS | 1     | \$H\$10<=\$J\$10 | Binding     | 0     |

|                         |             |    |            |             |    |
|-------------------------|-------------|----|------------|-------------|----|
| \$B\$11                 | Solution x1 | 0  | \$B\$11>=0 | Binding     | 0  |
| \$C\$11                 | Solution x2 | 0  | \$C\$11>=0 | Binding     | 0  |
| \$D\$11                 | Solution x3 | 25 | \$D\$11>=0 | Not Binding | 25 |
| \$E\$11                 | Solution x4 | 0  | \$E\$11>=0 | Binding     | 0  |
| \$F\$11                 | Solution x5 | 0  | \$F\$11>=0 | Binding     | 0  |
| \$G\$11                 | Solution x6 | 1  | \$G\$11>=0 | Binding     | 0  |
| \$B\$11:\$G\$11=Integer |             |    |            |             |    |

Figure 10. Excel Ganghi LP results

#### 8.4.1. Problems

**Blending problem.** Consider the example of a manufacturer of animal feed who is producing feed mix for dairy cattle. In our simple example the feed mix contains two active ingredients and a filler to provide bulk. One kg of feed mix must contain a minimum quantity of each of four nutrients as below:

TABLE 4

| Nutrient | A  | B  | C  | D |
|----------|----|----|----|---|
| Gram     | 90 | 50 | 20 | 2 |

The ingredients have the following nutrient values and cost

TABLE 5

|                        | A   | B   | C  | D  | Cost/kg |
|------------------------|-----|-----|----|----|---------|
| Ingredient 1 (gram/kg) | 100 | 80  | 40 | 10 | 40      |
| Ingredient 2 (gram/kg) | 200 | 150 | 20 | -  | 60      |

- What should be the amounts of active ingredients and filler in one kg of feed mix?

Suppose now we have the additional conditions:

- if we use any of ingredient 2 we incur a fixed cost of 15
- we need not satisfy all four nutrient constraints but need only satisfy three of them (i.e. whereas before the optimal solution required all four nutrient constraints to be satisfied now the optimal solution could (if it is worthwhile to do so) only have three (any three) of these nutrient constraints satisfied and the

fourth violated.

2. Give the complete MIP formulation of the problem with these two new conditions added.

## 8.5. Set-Covering Problems

The following example is typical of an important class of IPs known as set-covering problems.

### EXAMPLE 7. Facility-Location Set-Covering Problem

There are six cities (cities 1–6) in El Paso County. The county must determine where to build fire stations. The county wants to build the minimum number of fire stations needed to ensure that at least one fire station is within 15 minutes (driving time) of each city. The times (in minutes) required to drive between the cities in El Paso County are shown in Table 6. Formulate an IP that will tell El Paso how many fire stations should be built and where they should be located.

**Solution.** For each city, El Paso must determine whether to build a fire station there. We define the 0–1 variables  $x_1, x_2, x_3, x_4, x_5$ , and  $x_6$  by

$$x_{-i} = \begin{cases} 1 & \text{if a fire station is built in city } i \\ 0 & \text{otherwise} \end{cases}$$

Then the total number of fire stations that are built is given by  $x_1 + x_2 + x_3 + x_4 + x_5 + x_6$ , and El Paso's objective function is to minimize

$$z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

What are El Paso's constraints? El Paso must ensure that there is a fire station within 15 minutes of each city. Table 7 indicates which locations can reach the city in 15 minutes or less. To ensure that at least one fire station is within 15 minutes of city 1, we add the constraint

$$x_1 + x_2 > 1 \quad (\text{City 1 constraint})$$

This constraint ensures that  $x_1 = x_2 = 0$  is impossible, so at least one fire station will be built within 15 minutes of city 1. Similarly the constraint

$$x_1 + x_2 + x_6 > 1 \quad (\text{City 2 constraint})$$

ensures that at least one fire station will be located within 15 minutes of city 2. In a similar fashion, we obtain constraints for cities 3–6. Combining these six constraints with the

**TABLE 6.** Time Required to Travel between Cities in El Paso County

| From   | To     |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|
|        | City 1 | City 2 | City 3 | City 4 | City 5 | City 6 |
| City 1 | 0      | 10     | 20     | 30     | 30     | 20     |
| City 2 | 10     | 0      | 25     | 35     | 20     | 10     |
| City 3 | 20     | 25     | 0      | 15     | 30     | 20     |
| City 4 | 30     | 35     | 15     | 0      | 15     | 25     |
| City 5 | 30     | 20     | 30     | 15     | 0      | 14     |
| City 6 | 20     | 10     | 20     | 25     | 14     | 0      |

**TABLE 7.** Cities within 15 Minutes of Given City

| City | Within 15 Minutes |
|------|-------------------|
| 1    | 1, 2              |
| 2    | 1, 2, 6           |
| 3    | 3, 4              |
| 4    | 3, 4, 5           |
| 5    | 4, 5, 6           |
| 6    | 2, 5, 6           |

objective function (and with the fact that each variable must equal 0 or 1), we obtain the following 0–1 IP:

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

s.t.

$$\begin{aligned}
 x_1 + x_2 &> 1 \quad (\text{City 1 constraint}) \\
 x_1 + x_2 + x_6 &> 1 \quad (\text{City 2 constraint}) \\
 x_3 + x_4 &> 1 \quad (\text{City 3 constraint}) \\
 x_3 + x_4 + x_5 &> 1 \quad (\text{City 4 constraint}) \\
 x_4 + x_5 + x_6 &> 1 \quad (\text{City 5 constraint}) \\
 x_2 + x_5 + x_6 &> 1 \quad (\text{City 6 constraint})
 \end{aligned}$$

$$x_i = 0 \text{ or } 1 \quad (i = 1, 2, 3, 4, 5, 6)$$

One optimal solution to this IP is  $z = 2$ ,  $x_2 = x_4 = 1$ ,  $x_1 = x_3 = x_5 = x_6 = 0$ . Thus, El Paso County can build two fire stations: one in city 2 and one in city 4.

```
//#####
// SCRIPT FILE: El Paso County Facility Location
//#####
c = [1; 1; 1; 1; 1; 1];
A = [1 1 0 0 0 0 ;
      1 1 0 0 0 1 ;
      0 0 1 1 0 0 ;
      0 0 1 1 1 0 ;
      0 0 0 1 1 1 ;
      0 1 0 0 1 1 ];
b = [1; 1; 1; 1; 1; 1];
lb = [0 0 0 0 0 0 ]';
ub = [1 1 1 1 1 1 ]';
[xopt, lagr, fopt]=linpro(c,A,b,lb,ub,3)
fopt =
2.
lagr =
0.
0.
0.
0.
- 1.
0.
0
- 1.
- 1.
0.
0.
0.
0.
xopt =
1.
0
```

1.  
0  
0.  
0

As noted, Example 7 represents a class of IPs known as set-covering problems. In a set-covering problem, each member of a given set (call it set 1) must be “covered” by an acceptable member of some set (call it set 2). The objective in a set-covering problem is to minimize the number of elements in set 2 that are required to cover all the elements in set 1. In Example 7, set 1 is the cities in El Paso County, and set 2 is the set of fire stations. The station in city 2 covers cities 1, 2, and 6, and the station in city 4 covers cities 3, 4, and 5. Set-covering problems have many applications in areas such as airline crew scheduling, political districting, airline scheduling, and truck routing.

### 8.5.1. Problems

1. The Holy Emperor is waging a desperate war against The Supreme Sultan. The count of Narnia, who is a vassal of The Holy Emperor, fathoms that the war will be lost in exactly 3 years, and after the war there will be no county, nor count, of Narnia. So, the count decides to capitalize his possessions. After the capitalization, the count intends to disappear into the New World. The count’s income comes from the 6 villages under his yoke. The average monthly income of the villages are (in Holy Imperial Denara):

TABLE 8

| Village | Taxes |
|---------|-------|
| 1       | 8     |
| 2       | 19    |
| 3       | 18    |
| 4       | 18    |
| 5       | 19    |
| 6       | 102   |

To collect the taxes, the count must hire imperial tax-collectors. The imperial tax-collectors are placed in the villages and their salary is 2 Holy

Imperial Denara per week. The time it takes for a tax-collector based in the village  $i$  to collect the taxes from the village  $j$ , and then to deliver them to the count are (in days):

TABE 9

| Tax-collectors village | Taxed village |     |     |     |     |     | Delivery |
|------------------------|---------------|-----|-----|-----|-----|-----|----------|
|                        | 1             | 2   | 3   | 4   | 5   | 6   |          |
| 1                      | 0             | 100 | 100 | 107 | 101 | 192 | 12       |
| 2                      | 100           | 0   | 128 | 135 | 197 | 197 | 71       |
| 3                      | 100           | 128 | 0   | 115 | 130 | 189 | 87       |
| 4                      | 107           | 135 | 115 | 0   | 192 | 185 | 88       |
| 5                      | 101           | 197 | 130 | 192 | 0   | 184 | 89       |
| 6                      | 192           | 197 | 189 | 185 | 184 | 0   | 97       |

Now, the Holy Emperor finances his war by diluting the gold content of the Holy Roman Denara by 1% each day.

How much should The Supreme Sultan offer to the count of Narnia to disappear immediately?

## 2. Solve the TSP

### 8.6. Using the Excel Solver to Solve IP Problems

It is easy to use the Excel Solver to solve integer programming problems. Create a file Mandella.xls that will contain a spreadsheet solution to Example 6. See Figure 6 for the optimal solution. Set up the spreadsheet using the information in Figure 7. In our spreadsheet, the changing cells **J4:J6** (the number of each product produced) must be integers. To tell the Solver that these changing cells must be integers, just select Add Constraint and point to the cells **J4:J6**. Then select **int** from the drop-down arrow in the middle.

The changing cells **K4:K6** are the binary fixed charge variables. To tell the Solver that these changing cells must be binary, select Add Constraint and point to cells **K4:K6**. Then select **bin** from the drop-down arrow. See Figure 8.

From Figure 9, we find that the optimal solution (as found with SCILAB) is to make 25 pairs of pants.

| Gandhi                          |             |                  |                     |                      |           |             |               |             |                 |  |
|---------------------------------|-------------|------------------|---------------------|----------------------|-----------|-------------|---------------|-------------|-----------------|--|
|                                 |             | Labor hours used | Cloth yards used    | Unit price           | Unit cost | Unit profit | Fixed Cost    | Number Made | Binary variable |  |
|                                 | Shirt       | 3                | 4                   | \$ 12.00             | \$ 6.00   | \$ 6.00     | \$ 200.00     | 0           | 0               |  |
|                                 | Shorts      | 2                | 3                   | \$ 8.00              | \$ 4.00   | \$ 4.00     | \$ 150.00     | 0           | 0               |  |
|                                 | Pants       | 6                | 4                   | \$ 15.00             | \$ 8.00   | \$ 7.00     | \$ 100.00     | 25          | 1               |  |
| <b>Resource Constraints</b>     |             |                  |                     |                      |           |             |               |             |                 |  |
|                                 |             | Used             | Available           |                      |           |             | Fixed charge  | \$ 100.00   |                 |  |
| Labor                           |             | 150              | <=                  | 150                  |           |             | Variable cost | \$ 200.00   |                 |  |
| Cloth                           |             | 100              | <=                  | 160                  |           |             | Revenue       | \$ 375.00   |                 |  |
|                                 |             |                  |                     |                      |           |             | Profit        | \$ 75.00    |                 |  |
| <b>Fixed Charge Constraints</b> |             |                  |                     |                      |           |             |               |             |                 |  |
|                                 | Number Made |                  | Logical Upper Bound | Max possible to make |           |             |               |             |                 |  |
| Shirts                          |             | 0                | <=                  | 0                    | 40        |             |               |             |                 |  |
| Shorts                          |             | 0                | <=                  | 0                    | 53.33333  |             |               |             |                 |  |
| Pants                           |             | 25               | <=                  | 25                   | 25        |             |               |             |                 |  |

FIGURE 6. Optimal solution to the Mandella problem

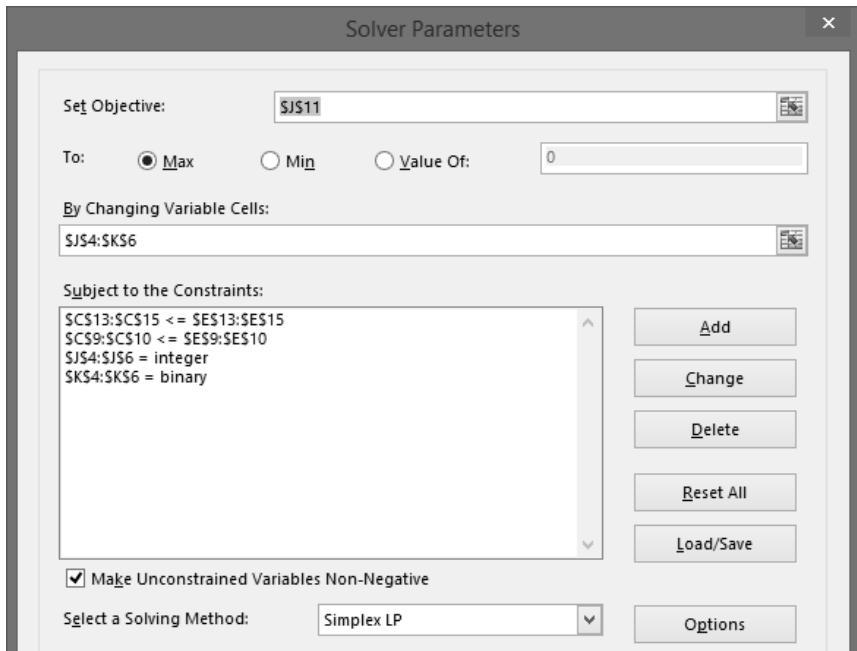


FIGURE 7. Solver setup

| Gandhi                   |        |                          |                     |                            |           |             |            |
|--------------------------|--------|--------------------------|---------------------|----------------------------|-----------|-------------|------------|
|                          |        | Labor hours used         | Cloth yards used    | Unit price                 | Unit cost | Unit profit | Fixed Cost |
|                          | Shirt  | 3                        | 4                   | 12                         | 6         | =F4-G4      | 200        |
|                          | Shorts | 2                        | 3                   | 8                          | 4         | =F5-G5      | 150        |
|                          | Pants  | 6                        | 4                   | 15                         | 8         | =F6-G6      | 100        |
| Resource Constraints     |        |                          |                     |                            |           |             |            |
|                          |        | Used                     | Available           |                            |           |             |            |
| Labor                    |        | =SUMPRODUCT(D4:D6,J4:J6) | <=                  | 150                        |           |             |            |
| Cloth                    |        | =SUMPRODUCT(E4:E6,J4:J6) | <=                  | 160                        |           |             |            |
|                          |        |                          |                     |                            |           |             |            |
| Fixed Charge Constraints |        |                          |                     |                            |           |             |            |
|                          |        | Number Made              | Logical Upper Bound | Max possible to make       |           |             |            |
| Shirts                   |        | =J4                      | =F13*K4             | =MIN(\$E\$9/D4,\$E\$10/E4) |           |             |            |
| Shorts                   |        | =J5                      | =F14*K5             | =MIN(\$E\$9/D5,\$E\$10/E5) |           |             |            |
| Pants                    |        | =J6                      | =F15*K6             | =MIN(\$E\$9/D6,\$E\$10/E6) |           |             |            |
|                          |        |                          |                     |                            |           |             |            |
| Variable Cost            |        |                          |                     |                            |           |             |            |
| Revenue                  |        |                          |                     |                            |           |             |            |
| Profit                   |        |                          |                     |                            |           |             |            |
|                          |        |                          |                     |                            |           |             |            |

**FIGURE 8.** Spreadsheet setup

Variable Cells

| Cell    | Name         | Original Value | Final Value | Integer |
|---------|--------------|----------------|-------------|---------|
| \$C\$38 | Solution x1  | 0              | 0           | Contin  |
| \$D\$38 | Solution x2  | 0              | 0           | Contin  |
| \$E\$38 | Solution x3  | 0              | 0           | Contin  |
| \$F\$38 | Solution x4  | 1              | 1           | Contin  |
| \$G\$38 | Solution x5  | 0              | 0           | Contin  |
| \$H\$38 | Solution x6  | 0              | 0           | Contin  |
| \$I\$38 | Solution x7  | 1              | 1           | Contin  |
| \$J\$38 | Solution x8  | 0              | 0           | Contin  |
| \$K\$38 | Solution x9  | 0              | 0           | Contin  |
| \$L\$38 | Solution x10 | 1              | 1           | Contin  |
| \$M\$38 | Solution x11 | 0              | 0           | Contin  |
| \$N\$38 | Solution x12 | 0              | 0           | Contin  |
| \$O\$38 | Solution x13 | 1              | 1           | Contin  |
| \$P\$38 | Solution x14 | 0              | 0           | Contin  |
| \$Q\$38 | Solution x15 | 0              | 0           | Contin  |
| \$R\$38 | Solution x16 | 0              | 0           | Contin  |
| \$S\$38 | Solution x17 | 0              | 0           | Contin  |
| \$T\$38 | Solution x18 | 0              | 0           | Contin  |
| \$U\$38 | Solution x19 | 0              | 0           | Contin  |

**Figure 9.** Excel Solver Solution

## 8.7. Review Problems

1. Coach Knight is trying to choose the starting lineup for the basketball team. The team consists of seven players who have been rated (on a scale of 1 = poor to 3 = excellent) according to their ball-handling, shooting, rebounding, and defensive abilities. The positions that each player is allowed to play and the player's abilities are listed in Table 10.

The five-player starting lineup must satisfy the following restrictions:

- (1) At least 4 members must be able to play guard, at least 2 members must be able to play forward, and at least 1 member must be able to play center.
- (2) The average ball-handling, shooting, and rebounding level of the starting lineup must be at least 2.
- (3) If player 3 starts, then player 6 cannot start.
- (4) If player 1 starts, then players 4 and 5 must both start.
- (5) Either player 2 or player 3 must start.

Given these constraints, Coach Night wants to maximize the total defensive ability of the starting team. Formulate an IP that will help him choose his starting team.

TABLE 10

| Player | Position | Ball-Handling | Shooting | Rebounding | Defense |
|--------|----------|---------------|----------|------------|---------|
| 1      | G        | 3             | 3        | 1          | 3       |
| 2      | C        | 2             | 1        | 3          | 2       |
| 3      | G-F      | 2             | 3        | 2          | 2       |
| 4      | F-C      | 1             | 3        | 3          | 1       |
| 5      | G-F      | 3             | 3        | 3          | 3       |
| 6      | F-C      | 3             | 1        | 2          | 3       |
| 7      | G-F      | 3             | 2        |            | 1       |

2. Stockcom is considering four investments. Investment 1 will yield a net present value (NPV) of \$16,000; investment 2, an NPV of \$22,000; investment 3, an NPV of \$12,000; and investment 4, an NPV of \$8,000. Each investment requires a certain cash outflow at the present time: investment 1, \$5,000; investment 2, \$7,000; investment 3, \$4,000; and investment 4, \$3,000. Currently, \$14,000 is available for investment. Formulate an IP whose solution will tell Stockcom how to maximize the NPV obtained from investments 1–4.
3. To graduate from Einstein University with a major in operations research, a student must complete at least two math courses, at

least two OR courses, and at least two computer courses. Some courses can be used to fulfill more than one requirement: Calculus can fulfill the math requirement; operations research, math and OR requirements; data structures, computer and math requirements; business statistics, math and OR requirements; computer simulation, OR and computer requirements; introduction to computer programming, computer requirement; and forecasting, OR and math requirements.

Some courses are prerequisites for others: Calculus is a prerequisite for business statistics; introduction to computer programming is a prerequisite for computer simulation and for data structures; and business statistics is a prerequisite for forecasting. Formulate an IP that minimizes the number of courses needed to satisfy the major requirements.

4. The manager of State University's DED computer wants to be able to access five different files. These files are scattered on 10 disks as shown in Table 11. The amount of storage required by each disk is as follows: disk 1, 3K; disk 2, 5K; disk 3, 1K; disk 4, 2K; disk 5, 1K; disk 6, 4K; disk 7, 3K; disk 8, 1K; disk 9, 2K; disk 10, 2K.
  - a. Formulate an IP that determines a set of disks requiring the minimum amount of storage such that each file is on at least one of the disks. For a given disk, we must either store the entire disk or store none of the disk; we cannot store part of a disk.
  - b. Modify your formulation so that if disk 3 or disk 5 is used, then disk 2 must also be used.

TABLE 11

| File | Disk |   |   |   |   |   |   |   |   |    |
|------|------|---|---|---|---|---|---|---|---|----|
|      | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1    | x    | x |   | x | x |   |   | x | x |    |
| 2    | x    |   | x |   |   |   |   |   |   |    |
| 3    |      | x |   | x |   | x |   |   |   | x  |
| 4    |      |   | x |   |   | x |   | x |   |    |
| 5    | x    | x |   | x |   | x | x |   | x | x  |

5. Use *SCILAB*, *LPSolve* or *Excel Solver* to find the optimal solution to the following IP: Book.com Publishers is considering publishing five textbooks. The maximum number of copies of each textbook that can be sold, the variable cost of producing each textbook, the sales price of each textbook, and the fixed cost of a production run for each book are given in Table 16. Thus, for example, producing 2,000 copies of book 1 brings in a revenue of  $2,000(50) = \$100,000$  but costs  $80,000 + 25(2,000) = \$130,000$ . Book.com can produce at most 10,000 books if it wants to maximize profit.
6. A company sells seven types of boxes, ranging in volume from 17 to 33 cubic feet. The demand and size of each box are given in Table 12. The variable cost (in dollars) of producing each box is equal to the box's volume. A fixed cost of \$1,000 is incurred to produce any of a particular box. If the company desires, demand for a box may be satisfied by a box of larger size. Formulate and solve (with *SCILAB*, *LPSolve* or *Excel Solver*) an IP whose solution will minimize the cost of meeting the demand for boxes.

TABLE 12

| Box    | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
|--------|-----|-----|-----|-----|-----|-----|-----|
| Size   | 433 | 330 | 326 | 324 | 319 | 318 | 317 |
| Demand | 400 | 300 | 500 | 700 | 200 | 400 | 200 |

7. A large drug company must determine how many sales representatives to assign to each of four sales districts. The cost of having  $n$  representatives in a district is  $(\$88,000 + \$80,000n)$  per year. If a rep is based in a given district, the time it takes to complete a call on a doctor is given in Table 48 (times are in hours).

Each sales rep can work up to 160 hours per month. Each month the number of calls given in Table 49 must be made in each district. A fractional number of representatives in a district is not permissible. Determine how many representatives should be assigned to each district.

TABLE 13

| District | Number of Calls |
|----------|-----------------|
| 1        | 50              |
| 2        | 80              |
| 3        | 100             |
| 4        | 60              |

8. Venture capital firm JD is trying to determine in which of 10 projects it should invest. It knows how much money is available for investment each of the next N years, the NPV of each project, and the cash required by each project during each of the next N years (see Table 14).
- Write a *SCILAB* Script to determine the projects in which JD should invest.
  - Use your *SCILAB* script to determine which of the 10 projects should be selected. Each project requires cash investment during the next three years. During year 1, \$80 million is available for investment. During year 2, \$60 million is available for investment. During year 3, \$70 million is available for investment. (All figures are in millions of dollars.)

TABLE 14

| Investment   | Project | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10  |
|--------------|---------|----|----|----|----|----|----|----|----|-----|-----|
| (\$ Million) |         | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10  |
| Year 1       |         | 6  | 9  | 12 | 15 | 18 | 21 | 24 | 27 | 30  | 35  |
| Year 2       |         | 3  | 5  | 7  | 9  | 11 | 13 | 15 | 17 | 19  | 21  |
| Year 3       |         | 5  | 7  | 9  | 12 | 12 | 14 | 16 | 11 | 20  | 24  |
| NPV          |         | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 130 |



# 9. Why Probability and Statistics?

## 9.1. Introduction

Before we proceed with the following chapters, we need to refresh our memory of some fundamental concepts that will be used in the stochastic processes that we will entertain. The deterministic models and processes we have discussed thus far are applicable to many situations, but certainly not all. Randomness and chance often come into play when analyzing the situations that you will face.

## 9.2. Basic Definitions

Some of the definitions provided in the subparagraphs below are given more explanation in Chapter 10.

### 9.2.1. Population

A population is any entire collection of people, animals, plants or things from which we may collect data. It is the entire group we are interested in, which we wish to describe or draw conclusions about.

#### **EXAMPLE 1. Population**

The population for a study of infant health might be all children born in the US in the 1980's. The sample might be all babies born on 7th May in any of the years.

### 9.2.2. Sample

A sample is a group of units selected from a larger group (the population). By studying the sample we hope to draw valid conclusions about the larger group.

#### **EXAMPLE 2. Sample**

The population for a study of infant health might be all children born in the US in the 1980's. The sample might be all babies born on 7th May in any of the years.

### 9.2.3. Parameter

A parameter is a value, usually unknown (and which therefore has to be estimated), used to represent a certain population characteristic. For example, the population mean is a parameter that is often used to indicate the average value of a quantity.

Parameters are often assigned Greek letters (e.g.  $\sigma$ ), whereas statistics are assigned Roman letters (e.g.  $s$ ).

### 9.2.4. Statistic

A statistic is a quantity that is calculated from a sample of data. It is used to give information about unknown values in the corresponding population. For example, the average of the data in a sample is used to give information about the overall average in the population from which that sample was drawn.

Statistics are often assigned Roman letters (e.g.  $m$  and  $s$ ), whereas the equivalent unknown values in the population (parameters) are assigned Greek letters (e.g.  $\mu$  and  $\sigma$ ).

### 9.2.5. Expected Value

The expected value (or population mean) of a random variable is a parameter that indicates the population's average or central value. It is a useful summary value (a number) of the variable's distribution. The expected value of a random variable  $X$  is symbolized by  $E(X)$  or  $\mu$ .

### 9.2.6. Variance

The population variance of a random variable is a non-negative number which gives an idea of how widely spread the values of the random variable are likely to be; the larger the variance, the more scattered the observations on average. Stating the variance gives an impression of how closely concentrated round the expected value the distribution is; it is a measure of the 'spread' of a distribution about its average value. The variance is symbolized by  $V(X)$  or  $Var(X)$  or  $\sigma^2$ .

### 9.2.7. Sample Mean

The sample mean is an estimator available for estimating the population mean. It is a measure of location, commonly called the average, often symbolized  $\bar{x}$ . The sample mean of size  $n$  is given by

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n} \quad (9.1)$$

Its value depends equally on all of the data which may include outliers. It may not appear representative of the central region for skewed (nonsymmetrical) data sets. The sample mean is a statistic.

### 9.2.8. Sample Variance

Sample variance is a measure of the spread of or dispersion within a set of sample data. The sample variance is the sum of the squared deviations from their average divided by one less than the number of observations in the data set. The sample variance is given by

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (9.2)$$

### 9.2.9. Random Variable

A Random Variable is a function, which assigns unique numerical values to all possible outcomes of a random experiment under fixed conditions (Ali 2000). A random variable is not a variable but rather a function that maps events to numbers.

#### **EXAMPLE 3.** Random Variables

This example is extracted from (Ali 2000). Suppose that a coin is tossed three times and the sequence of heads and tails is noted. The sample space for this experiment evaluates to:  $S = \{\text{HHH}, \text{HHT}, \text{HTH}, \text{HTT}, \text{THH}, \text{THT}, \text{TTH}, \text{TTT}\}$ . Now let the random variable  $X$  be the number of heads in three coin tosses.  $X$  assigns each outcome in  $S$  a number from the set  $SX = \{0, 1, 2, 3\}$ . The table below lists the eight outcomes of  $S$  and the corresponding values of  $X$ .

| Outcome | HHH | HHT | HTH | THH | HTT | THT | TTH | TTT |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| X       | 3   | 2   | 2   | 2   | 1   | 1   | 1   | 0   |

$X$  is then a random variable taking on values in the set  $SX = \{0, 1, 2, 3\}$ .

There are two types of random variables:

A Continuous Random Variable is one that takes an infinite number of possible values (Ali 2000). Example: Duration of a call in a telephone exchange.

A Discrete Random Variable is one that takes finite distinct values (Ali 2000). Example: A number of students who fail a test.

## 9.3. Conditional Probability and Bayes Rule

### 9.3.1. Basic Rules of Probability

In this section, we review some basic rules and definitions that you may have encountered during your previous study of probability.

**DEFINITION ■** Any situation where the outcome is uncertain is called an experiment. ■

For example, drawing a card from a deck of cards would be an experiment.

**DEFINITION ■** For any experiment, the sample space  $S$  of the experiment consists of all possible outcomes for the experiment. ■

For example, if we toss a die and are interested in the number of dots showing, then

$$S = \{1, 2, 3, 4, 5, 6\}.$$

**DEFINITION ■** An event  $E$  consists of any collection of points (set of outcomes) in the sample space. ■

A collection of events  $E_1, E_2, \dots, E_n$  is said to be a mutually exclusive collection of events if for  $i \neq j$  ( $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, n$ ),  $E_i$  and  $E_j$  have no points in common. ■

With each event  $E$ , we associate an event  $\bar{E}$ .  $\bar{E}$  consists of the points in the sample space that are not in  $E$ . With each event  $E$ , we also associate a number  $P(E)$ , which is the probability that event  $E$  will occur when we

perform the experiment. The probabilities of events must satisfy the following rules of probability:

**Rule 1.** For any event  $E$ ,  $P(E) \geq 0$ .

**Rule 2.** If  $E = S$  (that is, if  $E$  contains all points in the sample space), then  $P(E) = 1$ .

**Rule 3.** If  $E_1, E_2, \dots, E_n$  is a mutually exclusive collection of events, then

$$P(E_1 \cup E_2 \cup \dots \cup E_n) = \sum_{k=1}^n P(E_k)$$

**Rule 4.** The probability of the complement of an event  $E$  is  $P(\bar{E}) = 1 - P(E)$ .

**DEFINITION** ■ For two events  $E_1$  and  $E_2$ ,  $P(E_2|E_1)$  (the **conditional probability** of  $E_2$  given  $E_1$ ) is the probability that the event  $E_2$  will occur given that event  $E_1$  has occurred. Then

$$P(E_2|E_1) = \frac{P(E_1 \cap E_2)}{P(E_1)} \quad (9.3)$$

Suppose events  $E_1$  and  $E_2$  both occur with positive probability. Events  $E_1$  and  $E_2$  are **independent** if and only if  $P(E_2|E_1) = P(E_2)$  (or equivalently,  $P(E_1|E_2) = P(E_1)$ ). ■

Thus, events  $E_1$  and  $E_2$  are independent if and only if knowledge that  $E_1$  has occurred does not change the probability that  $E_2$  has occurred, and vice versa. From (1),  $E_1$  and  $E_2$  are independent if and only if

$$\frac{P(E_1 \cap E_2)}{P(E_1)} = P(E_2) \text{ or } P(E_1 \cap E_2) = P(E_1)P(E_2)$$

#### EXAMPLE 4. Machine Defects

A machine produces parts that are either good (90%), slightly defective (2%), or obviously defective (8%). Produced parts get passed through an automatic inspection machine, which is able to detect any part that is

obviously defective and discard it. What is the quality of the parts that make it through the inspection machine and get shipped?

Let  $G$  ( $SD$ ,  $OD$ ) be the event that a randomly chosen shipped part is good. Let  $SD$  be the event that a randomly shipped part is slightly defective. Let  $OD$  be the event that a randomly shipped part is obviously defective. We are told that  $P(G) = 0.90$ ,  $P(SD) = 0.02$ , and  $P(OD) = 0.08$ .

We want to compute the probability that a part is good *given* that it passed the inspection machine (i.e., it is *not* obviously defective), which is

$$P(G|\overline{OD}) = \frac{P(G \cap \overline{OD})}{P(\overline{OD})} = \frac{P(G)}{1 - P(OD)} = \frac{0.90}{1 - 0.09} = \frac{90}{92} = 0.978$$

**Multiplication Rule:** The “Multiplication Rule” (also known as the “Law of Multiplication”) states that, assuming  $P(F) > 0$ ,

$$P(E \cap F) = P(F) \cdot P(E|F),$$

which is (trivially) just a rewriting of the definition of conditional probability. The more general form is equally easy to prove from the definition:

$$\begin{aligned} & P(E_1 \cap E_2 \cap \cdots \cap E_n) \\ &= P(E_1)P(E_2|E_1)P(E_3|E_1 \cap E_2) \cdots P(E_n|E_1 \cap E_2 \cap \cdots \cap E_{n-1}). \end{aligned}$$

### EXAMPLE 5. Defective Components

Suppose that five good fuses and two defective ones have been mixed up. To find the defective fuses, we test them one-by-one, at random and without replacement. What is the probability that we are lucky and find both of the defective fuses in the first two tests?

Let  $D_1$  be the event that we find a defective fuse in the first (resp., second) test and  $D_2$  be the event that we find a defective fuse in the second test. We want to compute the probability of the event that we find a defective fuse in the first test and the second test.

$$P(D_1 \cap D_2) = P(D_1)P(D_2|D_1) = \frac{2}{7} \cdot \frac{1}{6} = \frac{1}{21}.$$

**Law of Total Probability:** The “Law of Total Probability” (also known as the “Method of Conditioning”) allows one to compute the probability of an event  $E$  by conditioning on cases, according to a partition of the sample space. For example, one way to partition  $S$  is to break into sets  $F$  and  $\bar{F}$ , for any event  $F$ . This gives us the simplest form of the law of total probability:

$$P(E) = P(E \cap F) + P(E \cap \bar{F}) = P(E|F)P(F) + P(E|\bar{F})P(\bar{F}).$$

More generally for any partition of  $S$  into sets  $F_1, \dots, F_n$ ,

$$P(E) = \sum_{i=1}^n P(E|F_i) P(F_i). \quad (9.4)$$

### EXAMPLE 6 (Parts Inspection)

Consider the parts problem again, but now assume that a one-year warranty is given for the parts that are shipped to customers. Suppose that a good part fails within the first year with probability 0.01, while a slightly defective part fails within the first year with probability 0.10. What is the probability that a customer receives a part that fails within the first year and therefore is entitled to a warranty replacement?

From before, we know that  $P(G) = \frac{90}{92}$  and  $P(SD) = \frac{2}{92}$ . Let  $E$  be the event that a randomly selected customer’s part fails in the first year. We are told that  $P(E|G) = 0.01$  and  $P(E|SD) = 0.10$ . We want to compute

$$\begin{aligned} P(E) &= P(E|G)P(G) + P(E|SD)P(SD) \\ &= (.01)\frac{90}{92} + (.10)\frac{2}{92} = \frac{11}{920} = 0.012 \end{aligned}$$

#### 9.3.2. Bayes’ Rule

An important decision often depends on the “state of the world.” For example, we may want to know whether a person has tuberculosis. Then

we would be concerned with the probability of the following states of the world:

More generally,  $n$  mutually exclusive states of the world  $(S_1, S_2, \dots, S_n)$  may occur. The states of the world are **collectively exhaustive**:  $S_1, S_2, \dots, S_n$  include all possibilities. Suppose a decision maker assigns a probability  $P(S_i)$  to  $S_i$ .  $P(S_i)$  is the **prior probability** of  $S_i$ . To obtain more information about the state of the world, the decision maker may observe the outcome of an experiment. Suppose that for each possible outcome  $O_j$  and each possible state of the world  $S_i$ , the decision maker knows  $P(O_j|S_i)$ , the **likelihood** of the outcome  $O_j$  given state of the world  $S_i$ . Bayes' rule combines prior probabilities and likelihoods with the experimental outcomes to determine a post-experimental probability, or **posterior probability**, for each state of the world. To derive Bayes' rule, observe that (1) implies that

$$P(S_i|O_j) = \frac{P(S_i \cap O_j)}{P(O_j)} \quad (9.5)$$

From (1), it also follows that

$$P(S_i \cap O_j) = P(O_j|S_i)P(S_i) \quad (9.6)$$

The states of the world  $S_1, S_2, \dots, S_n$  are collectively exhaustive, so the experimental outcome  $O_j$  (if it occurs) must occur with one of the  $S_i$  (see Figure 3). Since  $S_1 \cap O_j, S_2 \cap O_j, \dots, S_n \cap O_j$  are mutually exclusive events, probability rule 3 implies that

$$P(O_j) = P(S_1 \cap O_j) + P(S_2 \cap O_j) + \dots + P(S_n \cap O_j) \quad (9.7)$$

The probabilities of the form  $P(S_i \cap O_j)$  are often referred to as **joint probabilities**, and the probabilities  $P(O_j)$  are called **marginal probabilities**. Substituting (4) into (5), we obtain

$$P(O_j) = \sum_{k=1}^n P(O_j|S_k)P(S_k) \quad (9.8)$$

| $S_1$          | $S_2$          | $S_3$          | $S_4$          |
|----------------|----------------|----------------|----------------|
|                |                |                |                |
| $O_j \cap S_1$ | $O_j \cap S_2$ | $O_j \cap S_3$ | $O_j \cap S_4$ |

**FIGURE 1.** Illustration of (9.25):  $P(O_j) = P(S_1 \cap O_j) + P(S_2 \cap O_j) + P(S_3 \cap O_j) + P(S_4 \cap O_j)$

Substituting (4) and (6) into (3) yields **Bayes' rule**:

$$P(S_i|O_j) = \frac{P(O_j|S_i)P(S_i)}{\sum_{k=1}^n P(O_j|S_k)P(S_k)} \quad (9.9)$$

### EXAMPLE 7. Bayes' Rule

Suppose that 1% of all children have tuberculosis (TB). When a child who has TB is given the Mantoux test, a positive test result occurs 95% of the time. When a child who does not have TB is given the Mantoux test, a positive test result occurs 1% of the time. Given that a child is tested and a positive test result occurs, what is the probability that the child has TB?

**Solution.** The states of the world are

$S_1$  = child has TB

$S_2$  = child does not have TB The possible experimental outcomes are

$O_1$  = positive test result

$O_2$  = nonpositive test result

We are given the prior probabilities  $P(S_1) = .01$  and  $P(S_2) = .99$  and the likelihoods  $P(O_1|S_1) = .95$ ,  $P(O_1|S_2) = .01$ ,  $P(O_2|S_1) = .05$ , and  $P(O_2|S_2) = .99$ . We seek  $P(S_1|O_1)$ . From (9.9),

$$P(S_1|O_1) = \frac{P(O_1|S_1)P(S_1)}{P(O_1|S_1)P(S_1) + P(O_1|S_2)P(S_2)}$$

$$= \frac{.95(.010)}{.95(.01) + .01(.99)} = \frac{95}{194} = .49$$

### 9.3.3. Problems

1. One half percent of the population has a particular disease. A test is developed for the disease. The test gives a false positive 3% of the time and a false negative 2% of the time.
  - a. What is the probability that Joe (a random person) tests positive?
  - b. Joe just got the bad news that the test came back positive; what is the probability that Joe has the disease?
2. Consider the game of Let's Make a Deal in which there are three doors (numbered 1, 2, 3), one of which has a car behind it and two of which are empty (have "booby prizes"). You initially select Door 1, then, before it is opened, Monty Hall tells you that Door 3 is empty (has a booby prize). You are then given the option to switch your selection from Door 1 to the unopened Door 2. What is the probability that you will win the car if you switch your door selection to Door 2? Also, compute the probability that you will win the car if you do not switch. (What would you do?)
3. Let  $A$  and  $B$  be independent events with  $P(A) = \frac{1}{4}$  and  $P(A \cup B) = 2P(B) - P(A)$ . Find
  - a.  $P(B)$
  - b.  $P(A|B)$
  - c.  $P(\bar{B}|A)$ .
4. Let three fair coins be tossed. Let  $A = \{\text{all heads or all tails}\}$ ,  $B = \{\text{at least two heads}\}$ , and  $C = \{\text{at most two tails}\}$ . Of the pairs of events,  $(A, B)$ ,  $(A, C)$ , and  $(B, C)$ , which are independent and which are dependent? (Justify.)

### 9.4. Probability Distributions

The probability distribution of a discrete random variable is a list of probabilities associated with each of its possible values. It is also sometimes called the probability function or the probability mass function.

More formally, the probability distribution of a **discrete random variable**  $X$  is a function which gives the probability  $p(x_i)$  that the random variable equals  $x_i$ , for each value  $x_i$ :

$$p(x_i) = P(X = x_i) \quad (9.10)$$

It satisfies the following conditions:

$$0 \leq p(x_i) \leq 1$$

$$\sum p(x_i) = 1$$

The expected value and variance of a discrete random variable is given by

$$\mu = E(X) = \sum(x \cdot p(x_i)) \quad (9.11)$$

$$\sigma^2 = E(X^2) - \mu^2 = \sum(x^2 \cdot p(x_i)) - \mu^2 \quad (9.12)$$

### Cumulative Distribution Function

All random variables (discrete and continuous) have a cumulative distribution function. It is a function giving the probability that the random variable  $X$  is less than or equal to  $x$ , for every value  $x$ .

Formally, the cumulative distribution function  $F(x)$  is defined to be:

$$F(X) = P(X \leq x) \quad (9.13)$$

for  $-\infty < x < \infty$ .

For a discrete random variable, the cumulative distribution function is found by summing up the probabilities.

For a continuous random variable, the cumulative distribution function is the integral of its probability density function.

#### 9.4.1. Probability Density Function

The probability density function of a **continuous random variable** is a function which can be integrated to obtain the probability that the random variable takes a value in a given interval.

More formally, the probability density function,  $f(x)$ , of a continuous random variable  $X$  is the derivative of the cumulative distribution function  $F(x)$ :

$$f(x) = \frac{d}{dx} F(x) \quad (9.14)$$

Since  $F(X) = P(X \leq x)$ , it follows that:

$$\int f(x)dx = F(b) - F(a) = P(a < X < b) \quad (9.15)$$

If  $f(x)$  is a probability density function then it must obey two conditions: that the total probability for all possible values of the continuous random variable  $X$  is 1:  $\int f(x)dx = 1$

that the probability density function can never be negative:  $f(x) > 0$  for all  $x$ .

The expected value and variance of a continuous random variable is given by

$$\mu = E(X) = \int x \cdot f(x)dx \quad (9.16)$$

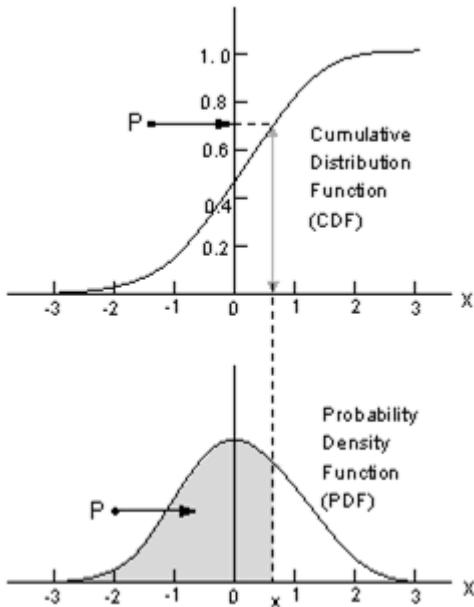
$$\sigma^2 = V(X) = E(X - E[X])^2 - E(X)^2 \quad (9.17)$$

#### 9.4.2. The Two Statistical Representations of a Population

Figure 1 depicts a typical relationship between the cumulative distribution function (cdf) and the probability density function (pdf), for continuous **random variables**.

A random variable is a real function (it is called "variable", but in reality it is a function) that assigns a numerical value to each simple event. For example, in sampling for quality control, an item could be defective or non-defective. Therefore, we may assign  $X = 1$  and  $X = 0$  for a defective and non-defective item, respectively. You may assign any other two distinct real numbers, as you wish; however, non-negative integer random variables are easy to work with. Random variables are

needed since we cannot do arithmetic operations on words; the random variable enables us to compute statistics, such as average and variance. Any random variable has a distribution of probabilities associated with it.



**FIGURE 1.** Relationship between two different typical representations of a population

All characteristics of the population are well described by either of these two functions. The figure also illustrates their applications in determining the (lower) percentile measures denoted by  $P$ :

$P = P[X \leq x]$  = Probability that the random variable  $X$  is less than or equal to a given number  $x$ .

Notice that the probability  $P$  is the area under the density function curve, while numerically equal to the height of cdf curve at point  $x$ .

Both functions can be estimated by *smoothing the empirical (i.e., observed) cumulative step-function*, and *smoothing the histogram* constructed from a random sample.

### 9.4.3. Normal Distributions

The term "normal" possibly arose because of the various attempts made to establish this distribution as the underlying rule governing all continuous variables. These attempts were based on false premises and consequently failed. Nonetheless, the normal distribution rightly occupies a preeminent place in the field of probability. In addition to portraying the distribution of many types of natural and physical phenomena (such as the heights of men, diameters of machined parts, etc.), it also serves as a convenient approximation of many other distributions which are less tractable. Most importantly, it describes the manner in which certain estimators of population characteristics vary from sample to sample and, thereby, serves as the foundation upon which much **statistical inference** from a random sample to population are made.

Normal Distribution (called also Gaussian) curves, which have a bell-shaped appearance (it is sometimes even referred to as the "bell-shaped curves") are very important in statistical analysis. In any normal distribution its observations are distributed symmetrically around the mean; 68% of all values under the curve lie within one standard deviation of the mean and 95% lie within two standard deviations.

There are many reasons for their popularity. The following are the most important reasons for its applicability:

One reason the normal distribution is important is that a wide variety of *naturally occurring random variables* such as heights and weights of all creatures are distributed evenly around a central value, average, or norm (hence, the name normal distribution). Although the distributions are only approximately normal, they are usually quite close.

Whenever there are too many factors influencing the outcome of a random outcome, then the underlying **distribution** is approximately normal. For example, the height of a tree is determined by the "sum" of such factors as rain, soil quality, sunshine, disease, etc.

Almost all **statistical tables are limited** by the size of their parameters. However, when these parameters are large enough one may use normal distribution for calculating the critical values for these tables. For

example, the  $F$ -statistic is related to standard normal  $z$ -statistic as follows:  $F = z^2$ , where  $F$  has ( $d.f._1 = 1$ , and  $d.f._2$  is the largest available in the  $F$ -table).

If the mean and standard deviation of a normal distribution are known, it is easy to convert back and forth from **raw scores to percentiles**.

It has been proven that the underlying distribution is normal if and only if the sample mean is independent of the sample variance, **this characterizes the normal distribution**. Therefore many effective **transformations** can be applied to convert almost any shaped distribution into a normal one.

The most important reason for popularity of normal distribution is the **Central Limit Theorem (CLT)**. The distribution of the sample averages of a large number of **independent random variables** will be approximately normal **regardless** of the distributions of the individual random variables. The Central Limit Theorem is a useful tool when you are dealing with a population with an unknown distribution. Often, you may analyze the mean (or the sum) of a sample of size  $n$ . For example, instead of analyzing the weights of individual items you may analyze the batch of size  $n$ , that is, the packages each containing  $n$  items.

**The Sampling distribution** of normal populations provide more information than any other distributions. For example, the following standard errors (i.e., having the same unit as the data have) are readily available:

- Standard Error of the Median
- Standard Error of the Standard Deviation
- Standard Error of the Variance
- Standard Error of the Interquartiles Half-Range
- Standard Error of the Skewness
- Standard Error of the Skewness of Sample Mean
- Standard Error of the Kurtosis
- Standard Error of the Correlation

The other reason the normal distributions are so important is that the normality condition is required by almost all kinds of parametric

**statistical tests.** Using most statistical tables, such as T-table (except its last row),  $\chi^2$ -table, and F-tables, all required the **normality condition** of the population. This condition must be tested before using these tables, otherwise the conclusion might be wrong.

#### 9.4.4. Sampling Distributions

A sampling distribution describes probabilities associated with a statistic when a random sample is drawn from the entire population. The sampling distribution is the density (for a continuous statistic, such as an estimated mean), or probability function (for discrete statistic, such as an estimated proportion). Derivation of the sampling distribution is the first step in calculating a **confidence interval** or carrying out a **hypothesis testing for a parameter**.

Example: Suppose that  $x_1, \dots, x_n$  are a simple random sample from a normally distributed population with expected value  $\mu$  and known variance  $\sigma^2$ . Then, the sample mean is normally distributed with expected value  $\bar{x}$  and variance  $s^2/n$ .

The main idea of **statistical inference** is to take a random sample from the entire particular **population** and then to use the information from the sample to make inferences about the particular population characteristics such as the mean  $\mu$  (measure of central tendency), the standard deviation (measure of dispersion, spread)  $\sigma$  or the proportion of units in the population that have a certain characteristic. Sampling saves money, time, and effort. Additionally, a sample can provide, in some cases, as much or more accuracy than a corresponding study that would attempt to investigate an entire population. Careful collection of data from a sample will often provide better information than a less careful study that tries to look at everything.

Often, we must also study the behavior of the mean of sample values taken from different specified populations; e.g., for comparison purposes. Because a sample examines only part of a population, the sample mean will not exactly equal the corresponding mean of the population  $\mu$ . Thus, an important consideration for those planning and interpreting sampling results is the degree to which sample estimates,

such as the sample mean, will agree with the corresponding population characteristic.

In practice, we usually take one sample only. In some cases we use a small “pilot sample” to test the data-gathering mechanisms and to get preliminary information for planning the main sampling scheme. However, for purposes of understanding the degree to which sample means will agree with the corresponding population mean  $\mu$ , it is useful to consider what would happen if 10, or 50, or 100 separate sampling studies, of the same type, were conducted. How consistent would the results be across these different studies? If we could see that the results from each of the samples would be nearly the same (and nearly correct!), then we would have confidence in the single sample that will actually be used. On the other hand, seeing that answers from the repeated samples were too variable for the needed accuracy would suggest that a different sampling plan (perhaps with a larger sample size) should be used.

We use a **sampling distribution** to describe the distribution of outcomes that one would observe from replication of a particular sampling plan.

Know that estimates computed from one sample will be different from estimates that would be computed from another sample.

Understand that estimates are expected to differ from the population characteristics (parameters) that we are trying to estimate, but that the properties of sampling distributions allow us to quantify, based on probability, how they will differ.

Understand that different statistics have different sampling distributions with distribution shape depending on (a) the specific statistic, (b) the sample size, and (c) the parent distribution.

Understand the relationship between sample size and the distribution of sample estimates.

Understand that increasing the sample size can reduce the variability in a sampling distribution.

See that in large samples, many sampling distributions can be approximated with a normal distribution.

**Sampling Distribution of the Mean and the Variance for Normal Populations:** Given the random variable  $X$  is distributed normally with mean  $\mu$  and standard deviation  $\sigma$ , then for a random sample of size  $n$ :

The sampling distribution of  $(\bar{x} - \mu) \times \sqrt{n} \div \sigma$ , is the standard normal distribution.

The sampling distribution of  $(\bar{x} - \mu) \times \sqrt{n} \div s$ , is a T-distribution with parameter d.f. =  $n-1$ .

The sampling distribution of,  $[s^2(n - 1) \div \sigma^2]$  is a  $\chi^2$ -distribution with parameter d.f. =  $n-1$ .

For two independent samples, the sampling distribution of  $(s_1^2/s_2^2)$ , is an  $F$ -distribution with parameters d.f.<sub>1</sub> =  $n_1-1$ , and d.f.<sub>2</sub> =  $n_2-1$ .

#### 9.4.5. Problems

1. The number of persons X, in a Singapore family chosen at random has the following probability distribution:

|        |      |      |      |      |      |      |      |      |
|--------|------|------|------|------|------|------|------|------|
| $X$    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    |
| $P(X)$ | 0.34 | 0.44 | 0.11 | 0.06 | 0.02 | 0.01 | 0.01 | 0.01 |

2. In a card game with my friend, I pay a certain amount of money each time I lose. I win \$4 if I draw a jack or a queen and I win \$5 if I draw a king or ace from an ordinary pack of 52 playing cards. If I draw other cards, I lose. What should I pay so that we come out even? (That is, the game is "fair"?)
3. I throw a die and get \$1 if it is showing 1, and get \$2 if it is showing 2, and get \$3 if it is showing 3, etc. What is the amount of money I can expect if I throw it 100 times?
4. Find the variance of the random variable X, given

|        |               |               |               |               |                |
|--------|---------------|---------------|---------------|---------------|----------------|
| $X$    | 8             | 12            | 16            | 20            | 24             |
| $P(X)$ | $\frac{1}{8}$ | $\frac{1}{6}$ | $\frac{3}{8}$ | $\frac{1}{4}$ | $\frac{1}{12}$ |

5. *SCILAB* can be used to calculate probabilities from a number of probability distribution functions, one being the normal distribution (pdf, cdf, and inverse):

```
normpdf(x, mean, standard deviation)
normcdf(x, mean, standard deviation)
norminv(P(x), mean, standard deviation)
```

- Find the probability that  $X = 100$  for a normally distributed random variable with mean 100 and standard deviation 5.
- Find the cumulative probability that  $X = 100$  for a normally distributed random variable with mean 100 and standard deviation 5.
- Find the value of the normally distributed random variable  $X$ , with mean 100 and standard deviation 5, when  $X = 75$ .
- Find the value of the normally distributed random variable  $X$ , with  $\mu = 0$  and  $\sigma = 1$ , when  $X = 0.3$ .

#### 9.4.6. Probability Distribution with *SCILAB*

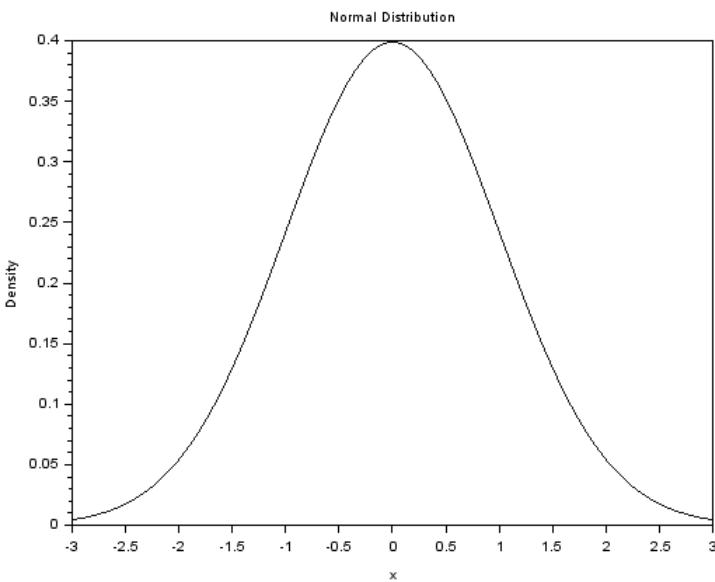
In this section we will look at *SCILAB*'s capability for computing normal pdfs, cdfs and probabilities.

##### **EXAMPLE 8.** Normal PDF

Suppose we have a random variable  $X$  that is normally distributed with mean 0 and standard deviation 1, that is,  $X \sim (x, 0, 1)$ . We wish to plot this pdf on the interval  $-3 < x < 3$ .

**Solution.** *SCILAB*'s function for the normal pdf is `distfun_normpdf(x,mu,sigma)`. In our situation, we have `x=[-3:0.1:3]` and `y=distfun_normpdf(x,0,1)`. So, in *SCILAB* we have:

```
x=[-3:0.05:3];
y=distfun_normpdf(x,0,1);
plot(x,y) //we will add axies labels and a title
xlabel("x")
ylabel("Density")
title("Normal Distribution")
```



**Figure 1.** Standard Normal Distribution ( $\mu = 0$  and  $\sigma = 1$ )

#### 9.4.6.1      Function cdfnor

Function cdfnor stands for “cumulative distribution function for the normal distribution.” A call to function cdfnor may involve three or more parameters which we will identify as follows:

$p$  = a probability representing the normal cdf, i.e.,  $P = P(X < x)$

$q = 1 - p$  = the probability of the complement , i.e.,  $q = P(X > x)$

$x$  = the value of  $X$  in the previous two expressions

$\mu$  = the mean value of the distribution ( $\mu$ )

$\sigma$  = the standard deviation of the distribution ( $\sigma$ )

Also, any of these parameters may be returned by the function call based on the values of the other parameters used as arguments in the call to function cdfnor. The following are the four possible function calls specifying the arguments to use and the parameters returned:

1.  $[p, q] = \text{cdfnor}("PQ", x, \mu, \sigma)$

2.  $[x] = \text{cdfnor}("X", mu, sigma, p, q)$
3.  $[mu] = \text{cdfnor}("Mean", sigma, p, q, x)$
4.  $[sigma] = \text{cdfnor}("Std", p, q, x, mu)$

Function call number 1 returns the probabilities  $p$  and  $q$  given the value of  $x$ , the mean,  $mu$ , and the standard deviation,  $sigma$ .

Function call number 2 returns the value of  $x$  given the mean,  $mu$ , the standard deviation,  $sigma$ , and the probabilities  $p$  and  $q$ .

Function call number 3 returns the value of the mean,  $mu$ , given the standard deviation,  $sigma$ , and the probabilities  $p$  and  $q$  corresponding to the value of  $x$ .

Function call number 4 returns the standard deviation of the distribution,  $sigma$ , given the mean value,  $mu$ , and the probabilities  $p$  and  $q$  corresponding to the value of  $x$ .

#### **EXAMPLE 9.** CDF Calculations

Calculate the values of  $p$  and  $q$  of the normal distribution with  $mu = 0$  and  $sigma = 1$ , for  $x = 1.1$ .

Solution.

$$P(X < 1.1) = F_x(1.1)$$

```
mu=0;
sigma=1;
cdfnor('PQ',1.1,mu,sigma)
ans =
0.8643339
```

#### **EXAMPLE 10.** CDF Calculations

Determine the mean given  $P(X < 2) = 0.35$  with a standard deviation of  $sigma = 0.5$ .

Solution

```

sigma=cdfnor("Mean",0.5, 0.35, 0.65, 2)
sigma =
2.1926602

```

#### 9.4.6.2 Calculation of mean, variance and standard deviation for a continuous random variable

For a continuous random variable with pdf  $f_x(x)$  the following formulas apply:

$$\mu_x = \int_{-\infty}^{\infty} x \cdot f_x(x) \cdot dx$$

$$VAR[x] = \int_{-\infty}^{\infty} (x - \mu_x)^2 \cdot f_x(x) \cdot dx$$

For example, the exponential distribution with pdf  $f_x(x) = \lambda e^{-\lambda x}$ , for  $\lambda > 0$ ,  $x > 0$ , has the following values for the mean, variance and standard deviation:

$$\mu_x = \frac{1}{\lambda}, \quad VAR[X] = \frac{1}{\lambda^2}, \quad \sigma_x = \frac{1}{\lambda}$$

#### **EXAMPLE 11.** Mean, Variance and Standard Deviation

Consider the probability density function given by

$$f_x(x) = k(1 - x^2)$$

Find the mean, variance and standard deviation.

Solution

The value  $k$  can be determined from the fact that  $\int_0^1 k f_x(x) dx = k \int_0^1 f_x(x) dx = k \int_0^1 (1 - x^2) dx = 1$ . To calculate the integral  $\int_0^1 (1 - x^2) dx$  using *SCILAB*, use the following commands:

```

integrate('1-x^2', 'x', 0,1)
ans =

```

```
0.6666667
```

The value of  $k$  turns out to be

```
1/ans  
ans =  
1.5
```

We can obtain a plot of the pdf by using:

```
deff('[y]=fx(x)', 'y=1.5*(1-x^2)';  
xx=[0:0.05:1];  
yy=fx(xx);  
plot(xx,yy);  
xlabel("x")  
ylabel("fx(x)")  
title("1-x^2")
```

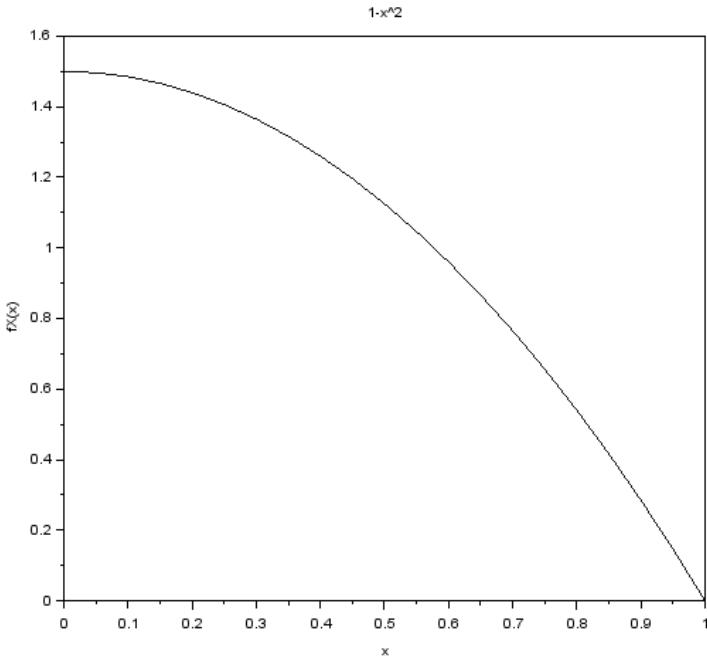
To calculate probabilities, we define the function  $f(x)$  and calculate the probabilities using the *SCILAB* function integrate as we indicate below.

$P(X < 0.3)$

```
integrate('fx(x)', 'x', 0, 0.3)  
ans =  
0.4365
```

$P(X > 0.7)$

```
integrate('fx(x)', 'x', 0.7, 1)  
ans =  
0.1215
```



**Figure 2.** PDF of  $1.5 * (1 - x^2)$

$$P(0.2 < X < 0.6)$$

```
integrate('fx(x)', 'x', 0.2, 0.6)
ans =
```

$$0.496$$

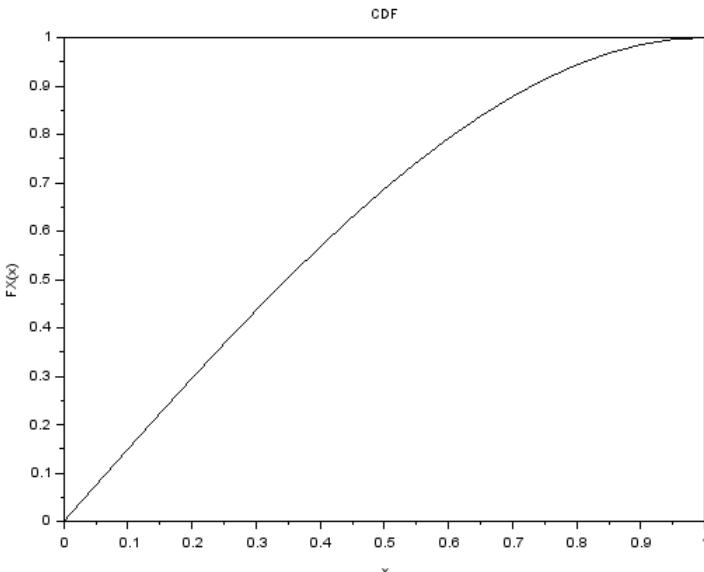
The corresponding cumulative distribution function (cdf) is easily calculated by hand as

$$F_X(x) = k \int_0^x (1 - \xi)^2 d\xi = k \left[ \xi - \frac{\xi^3}{3} \right]_0^x = k \left( x - \frac{x^3}{3} \right)$$

We can plot this function using *SCILAB* as follows:

```
deff(' [y]=FX(x)', 'y=1.5*(x-x^3/3)')
YY=FX(xx);
plot(xx,YY,)
```

```
xlabel("x")
ylabel("FX(x)")
title("CDF")
```



**Figure 3.** CDF of  $1.5 * (1 - x^2)$

## 9.5. The Central Limit Theorem

The **Central Limit Theorem (CLT)** is a “limit” that is “central” to statistical practice. For practical purposes, the main idea of the CLT is that the average (center of data) of a sample of observations drawn from some population is approximately distributed as a normal distribution if certain conditions are met. In theoretical statistics there are several versions of the central limit theorem depending on how these conditions are specified. These are concerned with the types of conditions made about the distribution of the parent population (population from which the sample is drawn) and the actual sampling procedure.

One of the simplest versions of the central limit theorem stated by many textbooks is:

For any population distribution with mean  $\mu$  and standard deviation  $\sigma$ , the sampling distribution of the sample mean (the distribution of all possible means for independent random samples of size  $n$ ) is approximately normal with mean  $\mu$  and standard deviation  $\sigma/\sqrt{n}$ , and the approximation improves as  $n$  increases.

The expected value of the sampling distribution of the sample mean is given by

$$E(\bar{X}) = \mu \quad (9.18)$$

The variance of the sampling distribution of the sample mean is given by

$$\sigma_{\bar{X}}^2 = \frac{\sigma^2}{n} \quad (9.19)$$

Note: Some students having difficulty reconciling their own understanding of the central limit theorem with some textbook statements. Some textbooks do not place an emphasis the on the **independent, random samples of fixed-size  $n$**  (say more than 30).

If the parent population has mean  $\mu$  and a finite standard deviation  $\sigma$ , then the sample mean distribution has the same mean  $\mu$  but with smaller standard deviation which is  $\sigma$  divided by  $\sqrt{n}$ . The shape of the sampling distributions for means becomes increasingly normal as the sample size  $n$  becomes larger. The increasing sample size is what causes the distribution to become increasingly normal and the independence condition provides the  $\sqrt{n}$  contraction of the standard deviation.

In applications of the central limit theorem to practical problems in **statistical inference**, however, we are more interested in how closely the approximate distribution of the sample mean follows a normal distribution for finite sample size, than in the limiting distribution itself. Sufficiently close agreement with a normal distribution allows us to use normal theory for making inferences about population parameters (such as the mean) using the sample mean, irrespective of the actual form of the parent population.

You know by now that, whatever the parent population is, the standardized variable  $Z = (X - \mu)/\sigma$  will have a distribution with a mean  $\mu= 0$  and standard deviation  $\sigma = 1$  under random sampling. Moreover, if the parent population is normal, then  $Z$  is distributed exactly as the standard normal. The central limit theorem states the remarkable result that, even when the parent population is non-normal, the standardized variable is approximately normal if the sample size is large enough. It is generally not possible to state conditions under which the approximation given by the central limit theorem works and what sample sizes are needed before the approximation becomes good enough. As a general guideline, statisticians have used the prescription that, if the parent distribution is symmetric and relatively short-tailed, then the sample mean **more** closely approximates normality for smaller samples than if the parent population is skewed or long-tailed.

### 9.5.1. An Illustration of CLT

**Sampling Distribution of the Sample Means:** Instead of working with individual scores, statisticians often work with means. What happens is that several samples are taken, the mean is computed for each sample, and then the means are used as the data, rather than individual scores being used. The sample is a sampling distribution of the sample means.

The central limit theorem explains why many distributions tend to be close to the normal distribution. The key ingredient is that the random variable being observed should be the sum or mean of many independent identically distributed random variables.

We can draw the probability distribution of the following random variables:

**EXAMPLE 12.** Sampling Distribution of Values ( $X$ )

Consider the case where a single, fair die is rolled. Here are the values that are possible and their probabilities.

**TABLE 1**

|             |     |     |     |     |     |     |
|-------------|-----|-----|-----|-----|-----|-----|
| $X$ Values  | 1   | 2   | 3   | 4   | 5   | 6   |
| Probability | 1/6 | 1/6 | 1/6 | 1/6 | 1/6 | 1/6 |

Here are the mean (9.1)), and variance (9.2) of this random variable  $X$ :

$$\text{Mean} = \mu = E[X] = \sum [x \cdot p(x)] = 3.5$$

$$\text{Variance} = \sigma^2 = E[X^2] - \mu^2 = \sum [x^2 \cdot p(x)] - \mu^2 = 2.92$$

**EXAMPLE 13.** Sampling Distribution of Samples' Mean ( $\bar{X}$ ).

Consider the case where two fair dice are rolled instead of one.

Here are the sums that are possible and their probabilities.

**TABLE 2**

| Sum  | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   |
|------|------|------|------|------|------|------|------|------|------|------|------|
| Prob | 1/36 | 2/36 | 3/36 | 4/36 | 5/36 | 6/36 | 5/36 | 4/36 | 3/36 | 2/36 | 1/36 |

But, we are not interested in the sum of the dice, we are interested in the sample mean. We find the sample mean by dividing the sum by the sample size.

**Table 3**

| $\bar{X}$ | 1.0  | 1.5  | 2.0  | 2.5  | 3.0  | 3.5  | 4.0  | 4.5  | 5.0  | 5.5  | 6.0  |
|-----------|------|------|------|------|------|------|------|------|------|------|------|
| Prob      | 1/36 | 2/36 | 3/36 | 4/36 | 5/36 | 6/36 | 5/36 | 4/36 | 3/36 | 2/36 | 1/36 |

Now we compute the mean, and variance, of this new random variable  $\bar{X}$ .

Here are the mean, and variance of the random variable  $\bar{X}$ :

$$\text{Mean}(\bar{X}) = E[\bar{X}] = \sum [\bar{x} \cdot p(\bar{x})] = 3.5$$

$$\text{Variance}(\bar{X}) = \sigma_{\bar{X}}^2 = E[\bar{X}^2] - \mu_{\bar{X}}^2 = \sum [\bar{x}^2 \cdot p(\bar{x})] - \mu_{\bar{X}}^2 = 1.46$$

Another way to think of sampling distributions is as **Probability Distribution of Random Variables**.

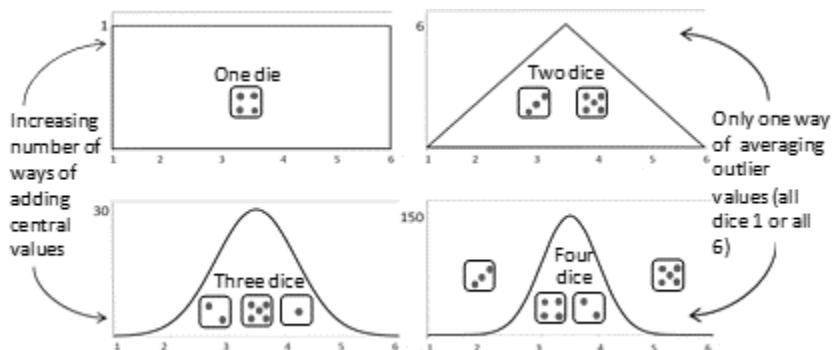
But, if we take repeated samples of the same size from a population, and then we plot the means of all those samples, our distribution will look a little better. We call distributions of sample statistics, **Sampling Distributions**.

The reason for this is that you can get the middle values in many more different ways than the extremes.

#### EXAMPLE 14: Dice Experiment

When throwing two dice:  $1+6 = 2+5 = 3+4 = 7$ , but only  $1+1 = 2$  and only  $6+6 = 12$ . That is: even though you get any of the six numbers equally likely when throwing one die, the extremes are less probable than middle values in sums of several dice.

To see how the central limit theorem works is in the distribution of scores from increasing numbers of dice throws.



**FIGURE 2.** Limiting Behavior of the Sample Mean: An Experimental Demonstration.

In Figure 2, the number on the top of each rolled die is an independent random event. Independent because the results of each die roll does not depend on the result of any previous roll, and random because, assuming that the die is “fair”, the value on the top of the rolled die cannot be predicted in advance. The sum of their results is the total number of dots on the tops of all the rolled dice. The bar chart illustrates the distribution of the sum. The distribution of each independent die roll is flat, not bell-shaped. See for yourself. Roll one die a bunch of times and watch the bar chart evolve. The distribution of the sum of two

independent die rolls is triangular. Try it and see. What about five dice? What about ten?

The CLT says that no matter what the distribution of the population looks like, the sampling distribution will be distributed normally, as long as your sample size is big enough (about 30). The distribution will have a mean equal to the population mean and a standard error equal to the population standard deviation divided by the square root of the sample size.

The measure of spread that we use for Sampling Distributions is the standard error (SE). The SE will always be smaller than the population standard deviation since the sampling distribution is one of sample statistics. Each sample mean will dampen the effect of outliers, bringing the tails of the sampling distribution in and creating a bigger “lump” in the middle, centered on the population mean. You can interpret the SE for sampling distributions in the same way as the standard deviation for populations.

### 9.5.2. Properties of the Sampling Distribution of the Sample Means:

When all of the possible sample means are computed, then the following properties are true:

- The mean of the sample means will be the mean of the population.
- The variance of the sample means will be the variance of the population divided by the sample size.
- The standard deviation of the sample means (known as the standard error of the mean) will be smaller than the population mean and will be equal to the standard deviation of the population divided by the square root of the sample size.
- If the population has a normal distribution, then the sample means will have a normal distribution.
- If the population is not normally distributed, but the sample size is sufficiently large, then the sample means will have an approximately normal distribution. Some books define sufficiently large as at least 30 and others as at least 31.

## 9.6. Exponential Density Function

An important class of decision problems under uncertainty concerns the chance between events. For example, the chance of the length of time to next breakdown of a machine not exceeding a certain time, such as the copying machine in your office not to break during this week.

Exponential distribution gives distribution of time between independent events occurring at a constant rate. Its density function is:

$$f(t) = \lambda \exp(-\lambda t) = \lambda e^{-\lambda t} \quad (9.20)$$

where  $\lambda$  is the average number of events per unit of time, which is a positive number.

The mean and the variance of the random variable  $t$  (time between events) are  $1/\lambda$ , and  $1/\lambda^2$ , respectively.

**Applications** include probabilistic assessment of the time between arrival of patients to the emergency room of a hospital, and arrival of ships to a particular port.

**Comments:** Special case of both Weibull and gamma distributions.

## 9.7. Poisson Process

An important class of decision problems under uncertainty is characterized by the small chance of the occurrence of a particular event, such as an accident. The Poisson Process gives probability of exactly  $x$  independent occurrences during a given period of time if events take place independently and at a constant rate. It may also represent number of occurrences over constant areas or volumes. The following statements describe the *Poisson Process*:

- The occurrences of the events are independent. The occurrence of events from a set of assumptions in an interval of space or time has no effect on the probability of a second occurrence of the event in the same, or any other, interval.
- Theoretically, an infinite number of occurrences of the event must be possible in the interval.

- The probability of the single occurrence of the event in a given interval is proportional to the length of the interval.
- In any infinitesimally small portion of the interval, the probability of more than one occurrence of the event is negligible.

Poisson process is often used, for example in quality control, reliability, insurance claim, incoming number of telephone calls, and queuing theory.

**An Application:** One of the most useful applications of the Poisson Process is in the field of queuing theory. In many situations where queues occur it has been shown that the number of people joining the queue in a given time period follows the Poisson model. For example, if the rate of arrivals to an emergency room is  $\lambda$  per unit of time period (say 1 hr), then:

$$P(n \text{ arrivals}) = \lambda^n e^{-\lambda} / n! \quad (9.21)$$

The mean and variance of random variable  $n$  are both  $\lambda$ . However, if the mean and variance of a random variable have equal numerical values, then it is not necessary that its distribution is a Poisson.

Applications:

$$P(0 \text{ arrivals}) = e^{-\lambda}$$

$$P(1 \text{ arrival}) = \lambda^1 e^{-\lambda} / 1!$$

$$P(2 \text{ arrivals}) = \lambda^2 e^{-\lambda} / 2!$$

and so on. In general:

$$P(n + 1 \text{ arrivals}) = \lambda P(n \text{ arrivals}) / n.$$

## 9.8. Goodness-of-Fit Tests

If we perform the input data analysis “by hand”, there are general four steps in the process.

1. Build a histogram of the data
2. Estimate (subjectively) shape and hypothesize a distribution
3. Estimate the parameters

Test the validity of the parameters:

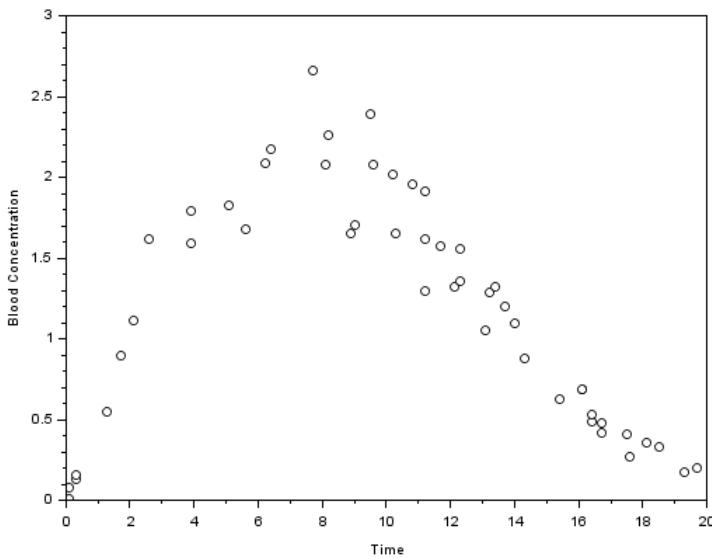
- Chi-Squared Test
- Kolmogorov-Smirnov (K-S) Test
- Anderson-Darling (A-D) Test

There is software, like *Stat::Fit*, to fit input data to discrete or continuous distribution. *Stat::Fit* is bundled with *ExtendSim 8*. *Stat::Fit* will “read” input data from a file, build a histogram, estimate the shape of a fitted distribution, and so on. However, we will perform this task with *SCILAB* and *R*.

Consider an experiment where we measure the concentration of a compound in blood samples taken from several subjects at various times after taking an experimental medication.

```
time = [ 0.1 0.1 0.3 0.3 1.3 1.7 2.1 2.6 3.9 3.9 ...
5.1 5.6 6.2 6.4 7.7 8.1 8.2 8.9 9.0 9.5 ...
9.6 10.2 10.3 10.8 11.2 11.2 11.2 11.7 12.1 12.3 ...
12.3 13.1 13.2 13.4 13.7 14.0 14.3 15.4 16.1 16.1 ...
16.4 16.4 16.7 16.7 17.5 17.6 18.1 18.5 19.3 19.7];
conc = [0.01 0.08 0.13 0.16 0.55 0.90 1.11 1.62 1.79 1.59
...
1.83 1.68 2.09 2.17 2.66 2.08 2.26 1.65 1.70 2.39 ...
2.08 2.02 1.65 1.96 1.91 1.30 1.62 1.57 1.32 1.56 ...
1.36 1.05 1.29 1.32 1.20 1.10 0.88 0.63 0.69 0.69 ...
0.49 0.53 0.42 0.48 0.41 0.27 0.36 0.33 0.17 0.20];
plot(time,conc, 'o');
xlabel('Time');
ylabel('Blood Concentration');
```

Notice that we have one response variable, blood concentration, and one predictor variable, time after ingestion. The predictor data are assumed to be measured with little or no error, while the response data are assumed to be affected by experimental error. The main objective in analyzing data like these is often to define a model that predicts the response variable. That is, we are trying to describe the trend line, or the mean response of y (blood concentration), as a function of x (time). This is curve fitting with bivariate data.



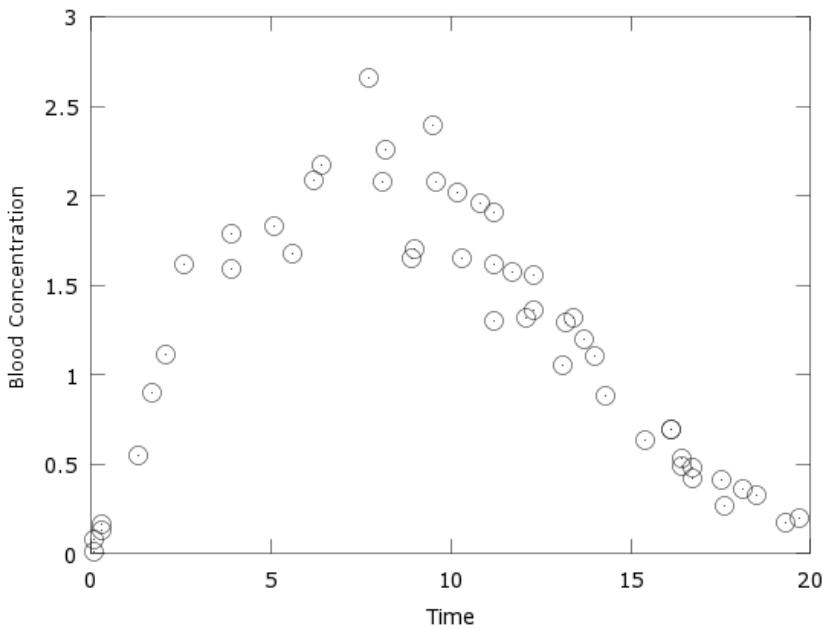
**Figure 4.** Scatterplot of Blood Concentration.

Based on theoretical models of absorption into and breakdown in the bloodstream, we might, for example, decide that the concentrations ought to follow a Weibull curve as a function of time. The Weibull curve, which takes the form

$$y = c \left( \frac{x}{a} \right)^{\{b-1\}} \exp \left( - \left( \frac{x}{a} \right)^b \right)$$

is defined with three parameters: the first scales the curve along the horizontal axis, the second defines the shape of the curve, and the third scales the curve along the vertical axis. Notice that while this curve has almost the same form as the Weibull probability density function, it is not a density because it includes the parameter  $c$ , which is necessary to allow the curve's height to adjust to data.

We can fit the Weibull model using nonlinear least squares.



**Figure 5.** Scatterplot

### 9.8.1. Distribution Fitting

#### 9.8.1.1 Using SCILAB

Consider an experiment where we've measured the time to failure for 50 identical electrical components.

```
life = [6.2 16.1 16.3 19.0 12.2 8.1 8.8 5.9 7.3 8.2 ...
16.1 12.8 9.8 11.3 5.1 10.8 6.7 1.2 8.3 2.3 ...
4.3 2.9 14.8 4.6 3.1 13.6 14.5 5.2 5.7 6.5 ...
5.3 6.4 3.5 11.4 9.3 12.4 18.3 15.9 4.0 10.4 ...
8.7 3.0 12.1 3.9 6.5 3.4 8.5 0.9 9.9 7.9];
```

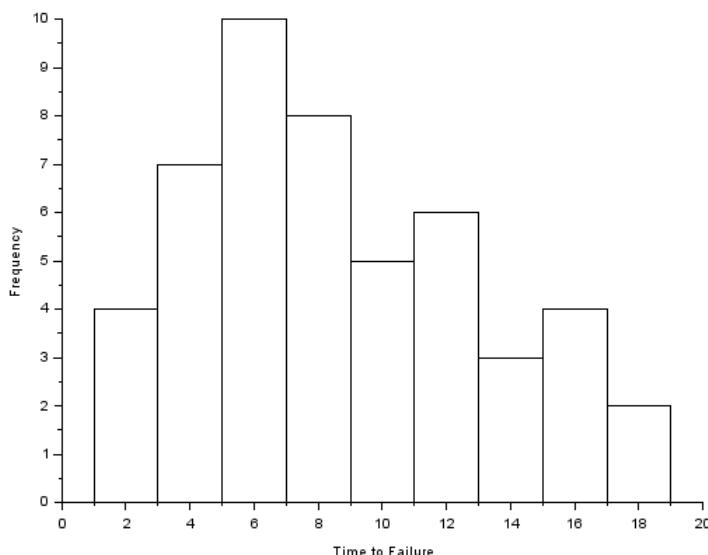
Notice that only one variable has been measured -- the components' lifetimes. There is no notion of response and predictor variables; rather, each observation consists of just a single measurement. The objective of an analysis for data like these is not to predict the lifetime of a new component given a value of some other variable, but rather to describe the full distribution of possible lifetimes. This is distribution fitting with univariate data.

One simple way to visualize these data is to make a histogram.

```

binwidth = 2;
binCtrs = 1:binwidth:19;
hist(life,binCtrs);
xlabel('Time to Failure');
ylabel('Frequency');
ylim([0 10]);

```



**Figure 5.** Histogram of Life using SCILAB

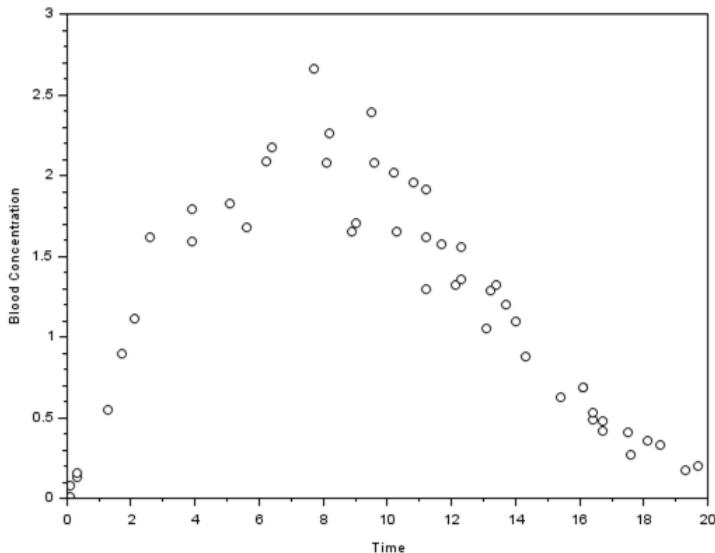
Consider an experiment where we measure the concentration of a compound in blood samples taken from several subjects at various times after taking an experimental medication.

```

time = [ 0.1 0.1 0.3 0.3 1.3 1.7 2.1 2.6 3.9 3.9 ...
5.1 5.6 6.2 6.4 7.7 8.1 8.2 8.9 9.0 9.5 ...
9.6 10.2 10.3 10.8 11.2 11.2 11.2 11.7 12.1 12.3 ...
12.3 13.1 13.2 13.4 13.7 14.0 14.3 15.4 16.1 16.1 ...
16.4 16.4 16.7 16.7 17.5 17.6 18.1 18.5 19.3 19.7];
conc = [0.01 0.08 0.13 0.16 0.55 0.90 1.11 1.62 1.79 1.59
...
1.83 1.68 2.09 2.17 2.66 2.08 2.26 1.65 1.70 2.39 ...
2.08 2.02 1.65 1.96 1.91 1.30 1.62 1.57 1.32 1.56 ...
1.36 1.05 1.29 1.32 1.20 1.10 0.88 0.63 0.69 0.69 ...
0.49 0.53 0.42 0.48 0.41 0.27 0.36 0.33 0.17 0.20];
plot(time,conc, 'o');
xlabel('Time');

```

```
ylabel('Blood Concentration');
```



**Figure 6.** Scatterplot of Life using **SCILAB**

Notice that we have one response variable, blood concentration, and one predictor variable, time after ingestion. The predictor data are assumed to be measured with little or no error, while the response data are assumed to be affected by experimental error. The main objective in analyzing data like these is often to define a model that predicts the response variable. That is, we are trying to describe the trend line, or the mean response of  $y$  (blood concentration), as a function of  $x$  (time). This is curve fitting with bivariate data. The **SCILAB** function we define will use the coefficients generated by the least squares procedure.

```
function y=yth(t, x)
    y = x(1)*cos(x(2)*t)+x(2)*sin(x(1)*t)
endfunction
// we have the m measures (ti, yi):
m = 50;
tm = [ 0.1 0.1 0.3 0.3 1.3 1.7 2.1 2.6 3.9 3.9 ...
5.1 5.6 6.2 6.4 7.7 8.1 8.2 8.9 9.0 9.5 ...
9.6 10.2 10.3 10.8 11.2 11.2 11.7 12.1 12.3 ...
12.3 13.1 13.2 13.4 13.7 14.0 14.3 15.4 16.1 16.1 ...
16.4 16.4 16.7 16.7 17.5 17.6 18.1 18.5 19.3 19.7]';
```

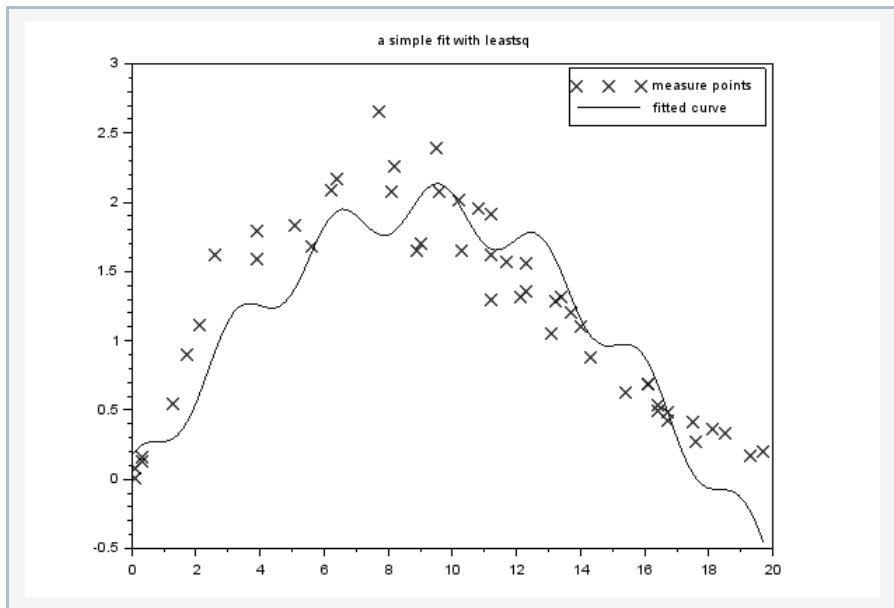
```

ym = [0.01 0.08 0.13 0.16 0.55 0.90 1.11 1.62 1.79 1.59 ...
1.83 1.68 2.09 2.17 2.66 2.08 2.26 1.65 1.70 2.39 ...
2.08 2.02 1.65 1.96 1.91 1.30 1.62 1.57 1.32 1.56 ...
1.36 1.05 1.29 1.32 1.20 1.10 0.88 0.63 0.69 0.69 ...
0.49 0.53 0.42 0.48 0.41 0.27 0.36 0.33 0.17 0.20]';
// measure weights (here all equal to 1...)
wm = ones(m,1);
// initial parameters guess
x1 = 0.25*ones(m,1);
x0=x1'
// in the first examples, we define the function fun and
dfun
function e=myfun(x, tm, ym, wm)
  e = wm.* ( yth(tm, x) - ym )
endfunction
//
function g=mydfun(x, tm, ym, wm)
  v = wm.*x(1)*cos(x(2)*tm)+wm.*x(2)*sin(x(1)*tm)
  g = [v , -x(1)*tm.*v]
endfunction
// now we could call leastsq:
[f,xopt, gopt] = leastsq(list(myfun,tm,ym,wm),x0)
[f,xopt, gopt] = leastsq(list(myfun,tm,ym,wm),mydfun,x0)
tt = linspace(0,1.0*max(tm),100)';
yy = yth(tt, xopt);
scf();
plot(tm, ym, "kx")
plot(tt, yy, "b")
legend(["measure points", "fitted curve"]);
xtitle("a simple fit with leastsq")

//Showing only the first columns of xopt
xopt =

```

|      | column 1 to 9 |          |      |      |      |      |      |      |      |
|------|---------------|----------|------|------|------|------|------|------|------|
| 0.25 | 0.1733551     | 1.969835 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 |



**Figure 7.** Blood concentration data with fitted mathematical expression

## 9.9. Introduction to R

This introduction serves as background material for our examples with R. The only hardware requirement for is a PC with the latest free open source R software installed. R has extensive documentation and active online community support. It is the perfect environment to get started in predictive modeling.

**Installation.** See Appendix A.

**Using External Data.** R offers plenty of options for loading external data, including Excel, Minitab and SPSS files.

**R Session.** After R is started, there is a console awaiting for input. At the prompt (>), you can enter numbers and perform calculations.

```
> 1 + 2
[1] 3
```

**Variable Assignment.** We assign values to variables with the assignment operator “=”. Just typing the variable by itself at the prompt will print out the value. We should note that another form of assignment operator “<-” is also in use.

```
> x = 1  
> x  
[1] 1
```

**Functions.** R functions are invoked by its name, then followed by the parenthesis, and zero or more arguments. The following apply the function `c` to combine three numeric values into a vector.

```
> c(1, 2, 3)  
[1] 1 2 3
```

**Comments.** All text after the pound sign “#” within the same line is considered a comment.

```
> 1 + 1      # this is a comment  
[1] 2
```

**Extension Package.** Sometimes we need additional functionality beyond those offered by the core R library. In order to install an extension package, you should invoke the `install.packages` function at the prompt and follow the instruction.

```
> install.packages()
```

**Getting Help.** R provides extensive documentation. For example, entering `?c` or `help(c)` at the prompt gives documentation of the function `c` in R.

```
> help(c)
```

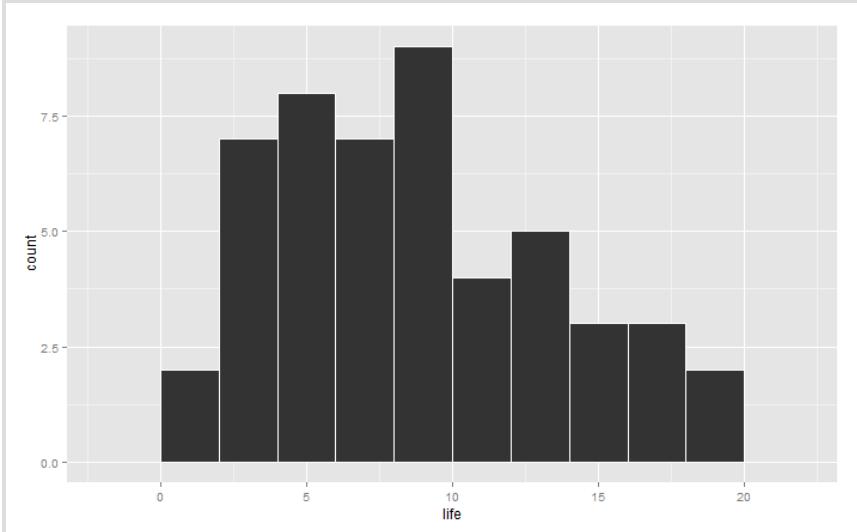
### 9.9.1.1 Using R

First we duplicate the histogram of the life data in R, and see that R renders the same result.

```
life = c(6.2, 16.1, 16.3, 19.0, 12.2, 8.1, 8.8, 5.9, 7.3,  
8.2,  
16.1, 12.8, 9.8, 11.3, 5.1, 10.8, 6.7, 1.2, 8.3, 2.3,  
4.3, 2.9, 14.8, 4.6, 3.1, 13.6, 14.5, 5.2, 5.7, 6.5,
```

```
5.3, 6.4, 3.5, 11.4, 9.3, 12.4, 18.3, 15.9, 4.0, 10.4,
8.7, 3.0, 12.1, 3.9, 6.5, 3.4, 8.5, 0.9, 9.9, 7.9);
```

```
mm=qplot(life)
mm+geom_histogram(binwidth = 2, color="white")
```



**Figure 8.** Histogram of Life data using R

Next we take the blood concentration data and fit a curve of the same family to the data using R.

```
# construct the data vectors using c()
time = c(0.01, 0.10, 0.30, 0.30, 1.30, 1.70, 2.10,
2.60, 3.90, 3.90, 5.10, 5.60, 6.20, 6.40, 7.70, 8.10,
8.20, 8.90, 9.00, 9.50, 9.60, 10.2, 10.3, 10.8, 11.2,
11.2, 11.2, 11.7, 12.1, 12.3, 12.3, 13.1, 13.2, 13.4,
13.7, 14.0, 14.3, 15.4, 16.1, 16.1, 16.4, 16.4, 16.7,
16.7, 17.5, 17.6, 18.1, 18.5, 19.3, 19.7)

conc = c(0.01, 0.08, 0.13, 0.16, 0.55, 0.90, 1.11,
1.62, 1.79, 1.59, 1.83, 1.68, 2.09, 2.17, 2.66, 2.08,
2.26, 1.65, 1.70, 2.39, 2.08, 2.02, 1.65, 1.96, 1.91,
1.30, 1.62, 1.57, 1.32, 1.56, 1.36, 1.05, 1.29, 1.32,
1.20, 1.10, 0.88, 0.63, 0.69, 0.69, 0.49, 0.53, 0.42,
0.48, 0.41, 0.27, 0.36, 0.33, 0.17, 0.20)

# look at it
plot(time,conc)

# some starting values
p1 = 0.1
```

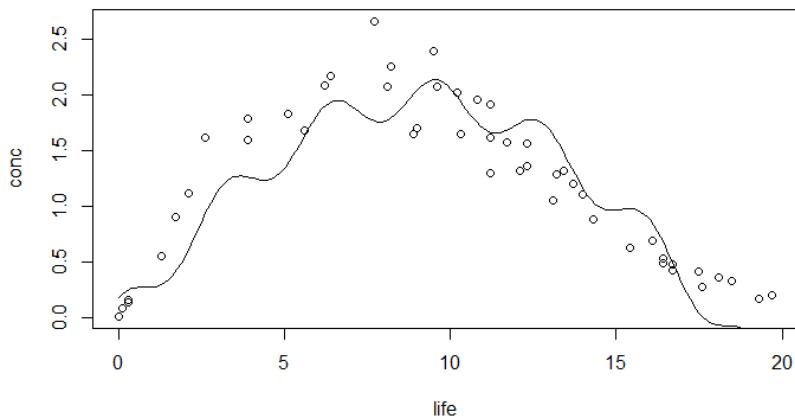
```

p2 = 0.2

# do the fit
fit = nls(conc ~ p1*cos(p2*time) + p2*sin(p1*time),
start=list(p1=p1,p2=p2))

# summarize
summary(fit)
new = data.frame(time = seq(min(time),max(time),len=200))
lines(new$time,predict(fit,newdata=new))

```



**Figure 9.** Fitted curve to Blood Concentration using R

### 9.9.1.2 Comparison

From looking at the fitted curves, it appears that both programs render similar fits. If we look at the coefficients  $x(1)$  and  $x(2)$  from **SCILAB**, we see that they are 0.1733551 and 1.969835, respectively. Comparing these with  $p1$  and  $p2$  from **R**, we have 0.17336 and 1.96986, respectively. Thus, the only difference is due to round-off error. If the number of digits are indicative of the actual decimal places used in calculations, I would take the **SCILAB** result over **R**. However, the R least squares routine was much easier to implement.

### 9.9.2. Hypothesis Testing Primer

There are two types of statistical inferences: estimation of population parameters and hypothesis testing. Hypothesis testing is one of the most important tools of application of statistics to real life problems. Most often, decisions are required to be made concerning populations on the basis of sample information. Statistical tests are used in arriving at these decisions.

There are five ingredients to any statistical test:

- Null Hypothesis
- Alternate Hypothesis
- Test Statistic
- Rejection/Critical Region
- Conclusion

In attempting to reach a decision, it is useful to make an educated guess or assumption about the population involved, such as the type of distribution.

**Statistical Hypotheses** : They are defined as assertion or conjecture about the parameter or parameters of a population, for example the mean or the variance of a normal population. They may also concern the type, nature or probability distribution of the population.

Statistical hypotheses are based on the concept of proof by contradiction. For example, say, we test that data obtained from observing interarrival times are exponentially distributed with a mean of 5 minutes. We do this by proof of contradiction by formulating a null hypothesis.

**Null Hypothesis** : It is a hypothesis which states that there is no difference between the procedures and is denoted by  $H_0$ . For the above example the corresponding  $H_0$  would be that “the data follows an exponential distribution with a mean of 5 minutes.” Always the null hypothesis is tested, i.e., we want to either accept or reject the null hypothesis because we have information only for the null hypothesis.

**Alternative Hypothesis** : It is a hypothesis which states that there is a difference between the procedures and is denoted by  $H_A$ .

**Test Statistic :** If we are concerned with the random variable  $X$  whose value is tested using samples to arrive at a decision, then the hypotheses test if for inference. The Central Limit Theorem states that for large sample sizes ( $n > 30$ ) drawn randomly from a population, the distribution of the means of those samples will approximate normality, even when the data in the parent population are not distributed normally. A  $z$  statistic is usually used for large sample sizes ( $n > 30$ ), but often large samples are not easy to obtain, in which case the  $t$ -distribution can be used. The population standard deviation  $\sigma$  is estimated by the sample standard deviation,  $s$ . The  $t$  curves are bell shaped and distributed around  $t = 0$ . The exact shape on a given  $t$ -curve depends on the degrees of freedom.

If we are concerned with conformity of data to a particular distribution, we are testing for goodness-of-fit. The test statistics differ according to the kind test (Chi-Squared, Kolmogorov-Smirnov, Anderson-Darling). The critical value of the test is usually taken from a table for the particular kind of test.

**Rejection Region :** It is the part of the sample space (critical region) where the null hypothesis  $H_0$  is rejected. The size of this region, is determined by the probability ( $\alpha$ ) of the sample point falling in the critical region when  $H_0$  is true.  $\alpha$  is also known as the level of significance, the probability of the value of the random variable falling in the critical region. Also it should be noted that the term "Statistical significance" refers only to the rejection of a null hypothesis at some level  $\alpha$ . It implies only that the observed difference between the sample statistic and the mean of the sampling distribution did not occur by chance alone.

**Conclusion :** If the test statistic falls in the rejection/critical region,  $H_0$  is rejected, else  $H_0$  is accepted.

#### EXAMPLE 15. Court Room Trial

A statistical test procedure is comparable to a trial; a defendant is considered innocent as long as his guilt is not proven. The prosecutor tries to prove the guilt of the defendant. Only when there is enough charging evidence the defendant is condemned.

In the start of the procedure, there are two hypotheses  $H_0$ : "the defendant is innocent", and  $H_1$ : "the defendant is guilty". The first one is called *null hypothesis*, and is for the time being accepted. The second one is called *alternative (hypothesis)*. It is the hypothesis one tries to prove.

The hypothesis of innocence is only rejected when an error is very unlikely, because one doesn't want to condemn an innocent defendant. Such an error is called *error of the first kind* (i.e. the condemnation of an innocent person), and the occurrence of this error is controlled to be seldom. As a consequence of this asymmetric behavior, the *error of the second kind* (setting free a guilty person), is often rather large.

## 9.10. Review Problems

- Suppose we have a random variable  $X$  that is normally distributed with mean 0 and standard deviation 1, that is,  $X \sim (x, 0, 1)$ . We wish to plot this pdf on the interval  $-3 < x < 3$ . Find  $P(X < 1.2) = F_x(1.1)$
- Calculate the mean and variance of the exponentially distributed random variable with  $\lambda = 7$ .
- The Iris data set in R gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.
  - Make a scatterplot, using SCILAB or R, of Sepal.Length versus Petal.Length. Observe that there might be a linear relationship between Sepal.Length and Petal.Length.
  - Fit a line to the data using least squares in SCILAB or R.
  - What is the equation of the line in (b)?
  - How would you describe the fit in (b)?
- The following table shows the sample means of a of the case where two fair dice are rolled.

| $\bar{X}$ | 1.0  | 1.5  | 2.0  | 2.5  | 3.0  | 3.5  | 4.0  | 4.5  | 5.0  | 5.5  | 6.0  |
|-----------|------|------|------|------|------|------|------|------|------|------|------|
| Prob      | 1/36 | 2/36 | 3/36 | 4/36 | 5/36 | 6/36 | 5/36 | 4/36 | 3/36 | 2/36 | 1/36 |

- a. Find the mean, variance and standard deviation.
- b. Interpret your results in the context of gambling with die.

# 10.What is Predictive Modeling?

**Predictive modeling** is the process by which a model is created or chosen to try to best predict the probability of an outcome (Geisser, 1993). In many cases the model is chosen on the basis of detection theory to try to guess the probability of an outcome given a set amount of input data, for example, given an email determining how likely that it is spam. Models can use one or more classifiers in trying to determine the probability of a set of data belonging to another set, say spam or ‘ham’.

There are three types of predictive models marketers should know about, but I will only talk about the first one in this article:

- Propensity models (predictions)
- Clustering models (segments)
- Collaborative filtering (recommendations)

Propensity models are what most people think of when they hear “predictive analytics”. Propensity models make predictions about a customer’s future behavior. With propensity models you can anticipate a customers’ future behavior. However, keep in mind that even propensity models are abstractions and do not necessarily predict absolute true behavior. we will go through six examples of propensity models to explain the concept.

## 10.1. Propensity Models

**Model 1:** Predicted customer lifetime value

CLV (Customer Lifetime Value) is a prediction of all the value a business will derive from their entire relationship with a customer. The Pareto Principle states that, for many events, roughly 80% of the effects come from 20% of the causes. When applied to e-commerce, this means that 80% of your revenue can be attributed to 20% of your customers. While the exact percentages may not be 80/20, it is still the case that some customers are worth a whole lot more than others, and identifying your “All-Star” customers can be extremely valuable to your business. Algorithms can predict how much a customer will spend with you long before customers themselves realizes this.

At the moment a customer makes their first purchase you may know a lot more than just their initial transaction record: you may have email and web engagement data for example, as well as demographic and geographic information. By comparing a customer to many others who came before them, you can predict with a high degree of accuracy their future lifetime value. This information is extremely valuable as it allows you to make value based marketing decisions. For example, it makes sense to invest more in those acquisition channels and campaigns that produce customers with the highest predicted lifetime value.

#### **Model 2:** Predicted share of wallet

Predicted share of wallet refers to the amount of the customer's total spending that a business captures in the products and services that it offers. Increasing the share of a customer's wallet a company receives is often a cheaper way of boosting revenue than increasing market share. For example if a customer spends \$100 with you on groceries, is this 10% or 90% of their grocery spending for a given year? Knowing this allows you to see where future revenue potential is within your existing customer base and to design campaigns to capture this revenue.

#### **Model 3:** Propensity to engage

A propensity to engage model predicts the likelihood that a person will engage in some activity, like unethical behavior or post purchases. For example, a propensity to engage model can predict how likely it is that a customer will click on your email links. Armed with this information you can decide not to send an email to a certain "low likelihood to click" segment.

#### **Model 4:** Propensity to unsubscribe

A propensity to unsubscribe model tells you which customers not to touch: if there are high value customers you are at risk of losing to unsubscribe, you need to find other ways to reaching out to them that are not by email. For example, you can predict how likely it is that a customer will unsubscribe from your email list at any given point in time. Armed with this information you can optimize email frequency. For "high likelihood to unsubscribe" segments, you should decrease send frequency; whereas for "low likelihood to unsubscribe" segments, you

can increase email send frequency. You could also decide to use different channels (like direct mail or LinkedIn) to reach out to “high likelihood to unsubscribe” customers.

### **Model 5:** Propensity to buy

The propensity to buy model tells you which customers are ready to make their purchase, so you can find who to target. Moreover, once you know who is ready and who is not helps you provide the right aggression in your offer. Those that are likely to buy won’t need high discounts (You can stop cannibalizing your margin) while customers who are not likely to buy may need a more aggressive offer, thereby bringing you incremental revenue.

For example, a “propensity to buy a new vehicle” model built with only data the automotive manufacturer has in their database can be used to predict percent of sales. By incorporating demographic and lifestyle data from third parties, the accuracy of that model can be improved. That is, if the first model predicts 50% sales in the top five deciles (there are ten deciles), then the latter could improve the result to 70% in the top five deciles.

### **Model 6:** Propensity to churn

Companies often rely on customer service agents to “save” customers who call to say they are taking their business elsewhere. But by this time, it is often too late to save the relationship. The propensity to churn model tells you which active customers are at risk, so you know which high value, at risk customers to put on your watch list and reach out. Armed with this information, you may be able to save those customers with preemptive marketing programs designed to retain them.

Often propensity models can be combined to make campaign decisions. For example, you may want to do an aggressive customer win back campaign for customers who have both a high likelihood to unsubscribe and a high predicted lifetime value.

## 10.2. Cluster Models

Clustering is the predictive analytics term for customer segmentation. Clustering, like classification, is used to segment the data. Unlike classification, clustering models segment data into groups that were not previously defined. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them.

With clustering you let the algorithms, rather than the marketers, create customer segments. Think of clustering as auto-segmentation. Algorithms are able to segment customers based on many more variables than a human being ever could. It's not unusual for two clusters to be different on 30 customer dimensions or more. In this article I will talk about three different types of predictive clustering models.

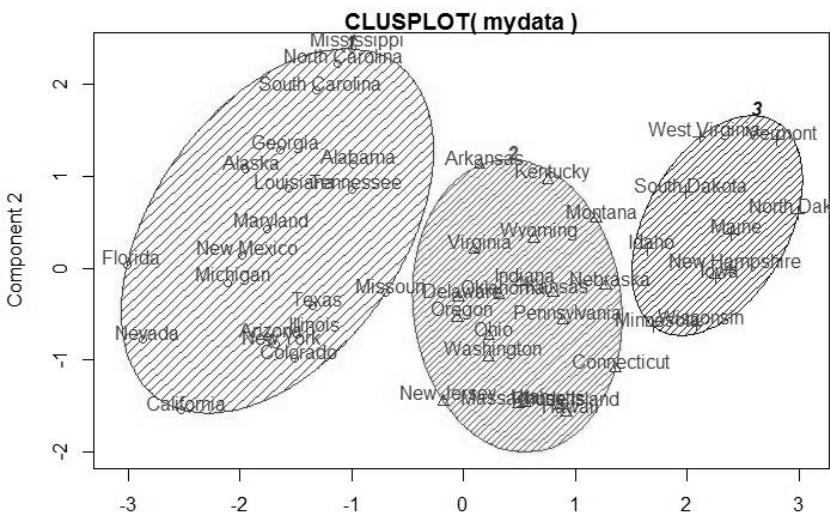
### **Model 7:** Behavioral clustering

Behavioral clustering informs you how people behave while purchasing. Do they use the web site or the call center? Are they discount addicts? How frequently do they buy? How much do they spend? How much time will go by before they purchase again? This algorithm helps set the right tone while contacting the customer. For instance, customers that buy frequently but with low sized orders might react well to offers like 'Earn double rewards points when you spend \$100 or more.'

|             | Murder | Assault | UrbanPop | Rape | classif |
|-------------|--------|---------|----------|------|---------|
| Alabama     | 13.2   | 236     | 58       | 21.2 | 1       |
| Alaska      | 10.0   | 263     | 48       | 44.5 | 1       |
| Arizona     | 8.1    | 294     | 80       | 31.0 | 1       |
| Arkansas    | 8.8    | 190     | 50       | 19.5 | 2       |
| California  | 9.0    | 276     | 91       | 40.6 | 1       |
| Colorado    | 7.9    | 204     | 78       | 38.7 | 1       |
| Connecticut | 3.3    | 110     | 77       | 11.1 | 2       |
| Delaware    | 5.9    | 238     | 72       | 15.8 | 2       |
| Florida     | 15.4   | 335     | 80       | 31.9 | 1       |
| Georgia     | 17.4   | 211     | 60       | 25.8 | 1       |

Behavioral clustering can also inform us on other behaviors, such as crime and is used in performing crime analysis. In the example below there are three crime clusters (only the top ten are shown in the table):

Graphically, the clusters appear as follow:



Cluster plot created in R Studio

**Model 8:** Product based clustering (also called category based clustering)

Product based clustering algorithms discover what different groupings of products people buy from. See the example below of a category (or product) based segment or cluster. You can see people in one customer segment ONLY buy Pinot Noir, whereas those in another customer segment buy different types of Varietal products, such as Champagne, Chardonnay, Pinot Grigio and Prosecco – but never Cabernet Sauvignon, Malbec or Espumante. This is useful information when deciding which product offers or email content to send to each of these customer segments.

Brand-based clustering, on the other hand, focuses on the brand of items customers purchase. Marketers can use this information to project what other brands those customers are likely to buy. Customers are then ordered according to Nike, Adidas, Under Armour, etc. Now you know what specific brands to pitch to certain customers. When a brand releases new products – you know who is likely to be interested.

| Offer # | Offer date | Product      | Minimur | Discount | Origin       | Past Peak | 1  | 2  | 3  | 4  |
|---------|------------|--------------|---------|----------|--------------|-----------|----|----|----|----|
| 1       | January    | Malbec       | 72      | 56       | France       | FALSE     | 0  | 0  | 4  | 6  |
| 2       | January    | Pinot Noir   | 72      | 17       | France       | FALSE     | 4  | 0  | 4  | 2  |
| 3       | February   | Espumante    | 144     | 32       | Oregon       | TRUE      | 0  | 0  | 2  | 4  |
| 4       | February   | Champagne    | 72      | 48       | France       | TRUE      | 0  | 0  | 7  | 5  |
| 5       | February   | Cabernet Sa  | 144     | 44       | New Zealar   | TRUE      | 0  | 0  | 2  | 2  |
| 6       | March      | Prosecco     | 144     | 86       | Chile        | FALSE     | 0  | 0  | 5  | 7  |
| 7       | March      | Prosecco     | 6       | 40       | Australia    | TRUE      | 0  | 12 | 4  | 3  |
| 8       | March      | Espumante    | 6       | 45       | South Africa | FALSE     | 0  | 11 | 6  | 3  |
| 9       | April      | Chardonnay   | 144     | 57       | Chile        | FALSE     | 0  | 0  | 7  | 3  |
| 10      | April      | Prosecco     | 72      | 52       | California   | FALSE     | 0  | 0  | 5  | 2  |
| 11      | May        | Champagne    | 72      | 85       | France       | FALSE     | 0  | 0  | 7  | 6  |
| 12      | May        | Prosecco     | 72      | 83       | Australia    | FALSE     | 0  | 0  | 3  | 2  |
| 13      | May        | Merlot       | 6       | 43       | Chile        | FALSE     | 0  | 6  | 0  | 0  |
| 14      | June       | Merlot       | 72      | 64       | Chile        | FALSE     | 0  | 0  | 5  | 4  |
| 15      | June       | Cabernet Sa  | 144     | 19       | Italy        | FALSE     | 0  | 0  | 2  | 4  |
| 16      | June       | Merlot       | 72      | 88       | California   | FALSE     | 0  | 0  | 5  | 0  |
| 17      | July       | Pinot Noir   | 12      | 47       | Germany      | FALSE     | 7  | 0  | 0  | 0  |
| 18      | July       | Espumante    | 6       | 50       | Oregon       | FALSE     | 0  | 11 | 2  | 1  |
| 19      | July       | Champagne    | 12      | 66       | Germany      | FALSE     | 0  | 0  | 2  | 3  |
| 20      | August     | Cabernet Sa  | 72      | 82       | Italy        | FALSE     | 0  | 0  | 4  | 2  |
| 21      | August     | Champagne    | 12      | 50       | California   | FALSE     | 0  | 0  | 2  | 2  |
| 22      | August     | Champagne    | 72      | 63       | France       | FALSE     | 0  | 0  | 0  | 23 |
| 23      | September  | Chardonnay   | 144     | 39       | South Africa | FALSE     | 0  | 0  | 3  | 2  |
| 24      | September  | Pinot Noir   | 6       | 34       | Italy        | FALSE     | 12 | 0  | 0  | 0  |
| 25      | October    | Cabernet Sa  | 72      | 59       | Oregon       | TRUE      | 0  | 0  | 3  | 3  |
| 26      | October    | Pinot Noir   | 144     | 83       | Australia    | FALSE     | 8  | 0  | 5  | 2  |
| 27      | October    | Champagne    | 72      | 88       | New Zealar   | FALSE     | 0  | 0  | 6  | 3  |
| 28      | November   | Cabernet Sa  | 12      | 56       | France       | TRUE      | 0  | 0  | 4  | 2  |
| 29      | November   | Pinot Grigio | 6       | 87       | France       | FALSE     | 0  | 15 | 2  | 0  |
| 30      | December   | Malbec       | 6       | 54       | France       | FALSE     | 0  | 16 | 2  | 4  |
| 31      | December   | Champagne    | 72      | 89       | France       | FALSE     | 0  | 0  | 10 | 7  |
| 32      | December   | Cabernet Sa  | 72      | 45       | Germany      | TRUE      | 0  | 0  | 3  | 1  |

## Model 9: Brand based clustering

### 10.3. Conclusion

Predictive analytics models are great, but they are ultimately useless unless you can actually tie them to your day-to-day marketing campaigns. This leads me to the first rule of predictive analytics:

“Always make sure that your predictive analytics platform is directly integrated with your marketing execution systems such as your email service provider, web site, call center or Point of Sell (POS) system.”

It is better to start with just one model that you use in day-to-day marketing campaigns than to have 10 models without the data being actionable in the hands of marketers.

### 10.4. Presenting and Using the Results of a Predictive Model

Predictive models can either be used directly to estimate a response (output) given a defined set of characteristics (input), or indirectly to drive the choice of decision rules (Steyerberg, 2010).

Depending on the methodology employed for the prediction, it is often possible to derive a formula that may be used in a spreadsheet software. This has some advantages for end users or decision makers, the main one being familiarity with the software itself, hence a lower barrier to adoption.

Nomograms are useful graphical representation of a predictive model. As in spreadsheet software, their use depends on the methodology chosen. The advantage of nomograms is the immediacy of computing predictions without the aid of a computer.

Point estimates tables are one of the simplest form to represent a predictive tool. Here combination of characteristics of interests can either be represented via a table or a graph and the associated prediction read off the y-axis or the table itself.

Tree based methods (e.g. CART, survival trees) provide one of the most graphically intuitive ways to present predictions. However, their usage is limited to those methods that use this type of modeling approach which can have several drawbacks (Breiman L. , 1996). Trees can also be employed to represent decision rules graphically.

Score charts are graphical tabular or graphical tools to represent either predictions or decision rules.

A new class of modern tools are represented by web based applications. For example, *Shiny* is a web based tool developed by *R studio*, an *R IDE* (integrated development environment). With a *Shiny* app, a modeler has the advantage to represent any which way he or she chooses to represent the predictive model while allowing the user some control. A user can choose a combination of characteristics of interest via sliders or input boxes and results can be generated, from graphs to confidence intervals to tables and various statistics of interests. However, these tools often require a server installation of *R studio*.

## 10.5. Applications

### 10.5.1. Uplift Modeling

Uplift Modeling (see Chapter 17) is a technique for modeling the change in probability caused by an action. Typically this is a marketing action such as an offer to buy a product, to use a product more or to re-sign a contract. For example in a retention campaign you wish to predict the change in probability that a customer will remain a customer if they are contacted. A model of the change in probability allows the retention campaign to be targeted at those customers on whom the change in probability will be beneficial. This allows the retention program to avoid triggering unnecessary churn or customer attrition without wasting money contacting people who would act anyway.

### 10.5.2. Archaeology

Predictive modeling in archaeology gets its foundations from Gordon Willey's mid-fifties work in the Virú Valley of Peru (Willey, 1953). Complete, intensive surveys were performed then covariability between cultural remains and natural features such as slope, and vegetation were determined. Development of quantitative methods and a greater availability of applicable data led to growth of the discipline in the 1960s and by the late 1980s, substantial progress had been made by major land managers worldwide.

Generally, predictive modeling in archaeology is establishing statistically valid causal or covariable relationships between natural proxies such as soil types, elevation, slope, vegetation, proximity to water, geology, geomorphology, etc., and the presence of archaeological features. Through analysis of these quantifiable attributes from land that has undergone archaeological survey, sometimes the "archaeological sensitivity" of unsurveyed areas can be anticipated based on the natural proxies in those areas. Large land managers in the United States, such as the Bureau of Land Management (BLM), the Department of Defense (DOD) (Altschul, Sebastian, & Heidelberg, 2004), and numerous highway and parks agencies, have successfully employed this strategy. By using predictive modeling in their cultural resource management plans, they are capable of making more informed decisions when planning for activities that have the potential to require ground disturbance and subsequently affect archaeological sites.

### 10.5.3. Customer relationship management

Predictive modeling is used extensively in analytical customer relationship management and data mining to produce customer-level models that describe the likelihood that a customer will take a particular action. The actions are usually sales, marketing and customer retention related.

For example, a large consumer organization such as a mobile telecommunications operator will have a set of predictive models for product cross-sell, product deep-sell and churn. It is also now more common for such an organization to have a model of “savability” using an uplift model. This predicts the likelihood that a customer can be saved at the end of a contract period (the change in churn probability) as opposed to the standard churn prediction model.

#### 10.5.4. Auto insurance

Predictive Modeling is utilized in vehicle insurance to assign risk of incidents to policy holders from information obtained from policy holders. This is extensively employed in usage-based insurance solutions where predictive models utilize telemetry based data to build a model of predictive risk for claim likelihood. Black-box auto insurance predictive models utilize GPS or accelerometer sensor input only. Some models include a wide range of predictive input beyond basic telemetry including advanced driving behavior, independent crash records, road history, and user profiles to provide improved risk models.

#### 10.5.5. Health care

In 2009 Parkland Health & Hospital System began analyzing electronic medical records in order to use predictive modeling to help identify patients at high risk of readmission. Initially the hospital focused on patients with congestive heart failure, but the program has expanded to include patients with diabetes, acute myocardial infarction, and pneumonia.

### 10.6. Notable failures of predictive modeling

Although not widely discussed by the mainstream predictive modeling community, predictive modeling is a methodology that has been widely used in the financial industry in the past and some of the spectacular

failures have contributed to the financial crisis of 2008. These failures exemplify the danger of relying blindly on models that are essentially backward looking in nature. The following examples are by no mean a complete list:

- 1) Bond rating. S&P, Moody's and Fitch quantify the probability of default of bonds with discrete variables called rating. The rating can take on discrete values from AAA down to D. The rating is a predictor of the risk of default based on a variety of variables associated with the borrower and macro-economic data that are drawn from historicals. The rating agencies failed spectacularly with their ratings on the 600 billion USD mortgage backed CDO market. Almost the entire AAA sector (and the super-AAA sector, a new rating the rating agencies provided to represent super safe investment) of the CDO market defaulted or severely downgraded during 2008, many of which obtained their ratings less than just a year ago.
- 2) Statistical models that attempt to predict equity market prices based on historical data. So far, no such model is considered to consistently make correct predictions over the long term. One particularly memorable failure is that of Long Term Capital Management, a fund that hired highly qualified analysts, including a Nobel Prize winner in economics, to develop a sophisticated statistical model that predicted the price spreads between different securities. The models produced impressive profits until a spectacular debacle that caused the then Federal Reserve chairman Alan Greenspan to step in to broker a rescue plan by the Wall Street broker dealers in order to prevent a meltdown of the bond market.

## 10.7. Possible fundamental limitations of predictive model based on data fitting

- 1) History cannot always predict future: using relations derived from historical data to predict the future implicitly assumes there are certain steady-state conditions or constants in the complex system. This is almost always wrong when the system involves people.
- 2) The issue of unknown unknowns: in all data collection, the collector first defines the set of variables for which data is collected. However, no

matter how extensive the collector considers his selection of the variables, there is always the possibility of new variables that have not been considered or even defined, yet critical to the outcome.

3) Self-defeat of an algorithm: after an algorithm becomes an accepted standard of measurement, it can be taken advantage of by people who understand the algorithm and have the incentive to fool or manipulate the outcome. This is what happened to the CDO rating. The CDO dealers actively fulfilled the rating agencies input to reach an AAA or super-AAA on the CDO they are issuing by cleverly manipulating variables that were “unknown” to the rating agencies’ “sophisticated” models.

## 10.8. More on Cluster Models

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, botryology (from Greek βότρυς “grape”) and typological analysis. The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest. This often leads to misunderstandings between researchers coming from the fields of data mining and machine learning, since they use the same terms and often the same algorithms, but have different goals.

Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert Tryon in 1939 (Bailey K. , 1994) (Tryon, 1939) and famously used by Cattell beginning in 1943 (Cattell, 1943) for trait theory classification in personality psychology.

## 10.9. Definition

According to Vladimir Estivill-Castro, the notion of a “cluster” cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms (Estivill-Castro, 2002). There is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these “cluster models” is key to understanding the differences between the various algorithms. Typical cluster models include:

**Connectivity models:** for example hierarchical clustering builds models based on distance connectivity.

**Centroid models:** for example the  $k$ -means algorithm represents each cluster by a single mean vector.

**Distribution models:** clusters are modeled using statistical distributions, such as multivariate normal distributions used by the Expectation-maximization algorithm.

**Density models:** for example DBSCAN and OPTICS defines clusters as connected dense regions in the data space.

**Subspace models:** in Biclustering (also known as Co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.

**Group models:** some algorithms do not provide a refined model for their results and just provide the grouping information.

**Graph-based models:** a clique, i.e., a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques.

A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Clusterings can be roughly distinguished as:

- hard clustering: each object belongs to a cluster or not
- soft clustering (also: fuzzy clustering): each object belongs to each cluster to a certain degree (e.g. a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- strict partitioning clustering: here each object belongs to exactly one cluster
- strict partitioning clustering with outliers: objects can also belong to no cluster, and are considered outliers.
- overlapping clustering (also: alternative clustering, multi-view clustering): while usually a hard clustering, objects may belong to more than one cluster.
- hierarchical clustering: objects that belong to a child cluster also belong to the parent cluster
- subspace clustering: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap.

## 10.10. Algorithms

Clustering algorithms can be categorized based on their cluster model, as listed above. The following overview will only list the most prominent examples of clustering algorithms, as there are possibly over 100 published clustering algorithms. Not all provide models for their clusters and can thus not easily be categorized. An overview of algorithms explained in Wikipedia can be found in the list of statistics algorithms.

There is no objectively “correct” clustering algorithm, but as it was noted, “clustering is in the eye of the beholder.” (Estivill-Castro, 2002) The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another. It should be noted that an algorithm that is designed for one kind of model has no chance on a data set that contains a radically different kind of model. For example,  $k$ -means cannot find non-convex clusters (Estivill-Castro, 2002).

### 10.10.1. Connectivity based clustering (hierarchical clustering)

Connectivity based clustering, also known as hierarchical clustering, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect “objects” to form “clusters” based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name “hierarchical clustering” comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the  $y$ -axis marks the distance at which the clusters merge, while the objects are placed along the  $x$ -axis such that the clusters don’t mix.

Strategies for hierarchical clustering generally fall into two types:

**Agglomerative:** This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

**Divisive:** This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Connectivity based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance to) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances) or UPGMA ("Unweighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

#### 10.10.2. Metric

The choice of an appropriate metric will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point (1,0) and the origin (0,0) is always 1 according to the usual norms, but the distance between the point (1,1) and the origin (0,0) can be 2 under Manhattan distance,  $\sqrt{2}$  under Euclidean distance, or 1 under maximum distance.

Some commonly used metrics for hierarchical clustering are (The DISTANCE Procedure: Proximity Measures):

| Names                      | Formula                                     |
|----------------------------|---|
| Euclidean distance         | $\ a - b\ _2 = \sqrt{\sum_i (a_i - b_i)^2}$ |
| Squared Euclidean distance | $\ a - b\ _2^2 = \sum_i (a_i - b_i)^2$      |

|                      |  |
|----------------------|--|
| Manhattan distance   | $\ a - b\ _1 = \sum_i  a_i - b_i $                                   |
| maximum distance     | $\ a - b\ _\infty = \max_i  a_i - b_i $                              |
| Mahalanobis distance | $\sqrt{(a - b)^T S^{-1} (a - b)}$ where $S$ is the Covariance matrix |

For text or other non-numeric data, metrics such as the Hamming distance or Levenshtein distance are often used. A review of cluster analysis in health psychology research found that the most common distance measure in published studies in that research area is the Euclidean distance or the squared Euclidean distance

#### 10.10.3. Linkage criteria

The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations.

Some commonly used linkage criteria between two sets of observations A and B are (The CLUSTER Procedure: Clustering Methods):

| Names  | Formula   |
|--|---|
| Maximum or complete linkage clustering       | $\max\{d(a, b) : a \in A, b \in B\}$  |
| Minimum or single-linkage clustering         | $\min\{d(a, b) : a \in A, b \in B\}$  |
| Mean or average linkage clustering, or UPGMA | $\frac{1}{ A  B } \sum_{a \in A} \sum_{b \in B} d(a, b)$  |
| Centroid linkage clustering, or UPGMC        | $\ c_s - c_t\ $ where $c_s$ and $c_t$ are the centroids of clusters $s$ and $t$ , respectively.   |
| Minimum clustering energy                    | $\frac{2}{nm} \sum_{i,j=1}^{n,m} \ a_i - b_j\ _2 - \frac{1}{n^2} \sum_{i,j=1}^n \ a_i - a_j\ _2 - \frac{1}{m^2} \sum_{i,j=1}^m \ b_i - b_j\ _2$ |

where  $d$  is the chosen metric. Other linkage criteria include:

- The sum of all intra-cluster variance.
- The decrease in variance for the cluster being merged (Ward's criterion) (Ward, 1963).
- The probability that candidate clusters spawn from the same distribution function (V-linkage).
- The product of in-degree and out-degree on a k-nearest-neighbor graph (graph degree linkage) (Zhang e. a., October 7–13, 2012).
- The increment of some cluster descriptor (i.e., a quantity defined for measuring the quality of a cluster) after merging two clusters (Zhang e. a., Agglomerative clustering via maximum incremental path integral, 2013).

These methods will not produce a unique partitioning of the data set, but a hierarchy from which the user still needs to choose appropriate clusters. They are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as “chaining phenomenon”, in particular with single-linkage clustering). In the general case, the complexity is  $\mathcal{O}(n^3)$ , which makes them too slow for large data sets. For some special cases, optimal efficient methods (of complexity  $\mathcal{O}(n^2)$ ) are known: SLINK (Sibson, 1973) for single-linkage and CLINK (Defays, 1977) for complete-linkage clustering. In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering.

## 10.11. Examples Using R

The ‘cluster’ package provides several useful functions for clustering analysis. We will use one here called ‘agnes’, which performs agglomerative hierarchical clustering of a dataset. The dataset we will use, ‘votes.repub’ is included in the package.

```
## First load the package.  
> library(cluster)
```

```

> data(votes.repub)
> agn1 <- agnes(votes.repub, metric = "manhattan",
+     stand = TRUE)
agn1
Call: agnes(x = votes.repub, metric = "manhattan", stand =
TRUE)
Agglomerative coefficient:  0.7977555
Order of objects:
[1] Alabama          Georgia          Arkansas         Louisiana
Mississippi       South Carolina
[7] Alaska           Vermont          Arizona          Montana
Nevada            Colorado
[13] Idaho            Wyoming         Utah             California
Oregon            Washington
[19] Minnesota       Connecticut      New York        New Jersey
Illinois           Ohio
[25] Indiana          Michigan         Pennsylvania     New
Hampshire         Wisconsin       Delaware
[31] Kentucky         Maryland         Missouri        New Mexico
West Virginia     Iowa
[37] South Colorado   North Colorado  Kansas          Nebraska
Maine             Massachusetts
[43] Rhode Island    Florida          North Carolina  Tennessee
Virginia          Oklahoma
[49] Hawaii           Texas

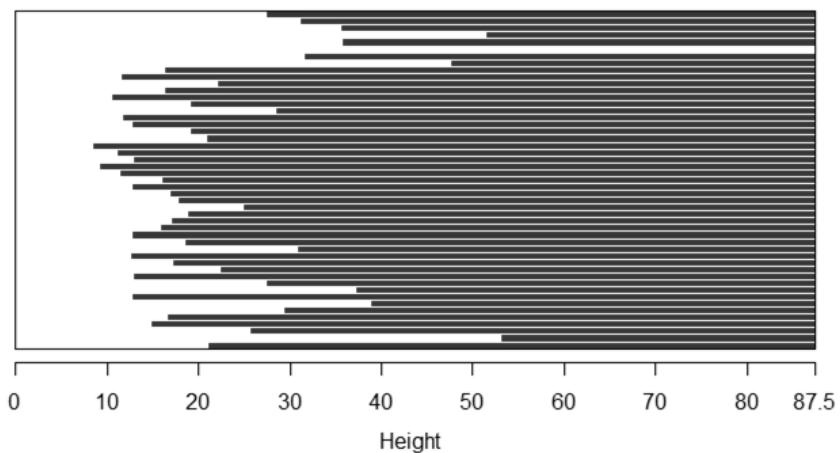
Height (summary):
  Min. 1st Qu. Median   Mean 3rd Qu.   Max.
  8.382 12.800 18.530 23.120 28.410 87.460

Available components:
[1] "order"      "height"      "ac"          "merge"      "diss"
"call"        "method"      "order.lab"
[9] "data"

> plot(agn1)

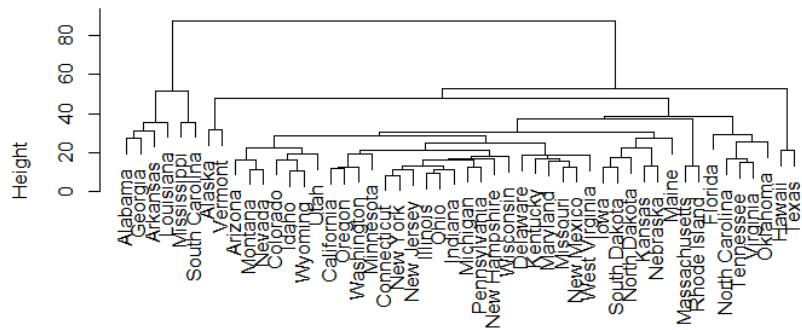
```

**Banner of agnes(x = votes.repub, metric = "manhattan", stand = TRUE)**



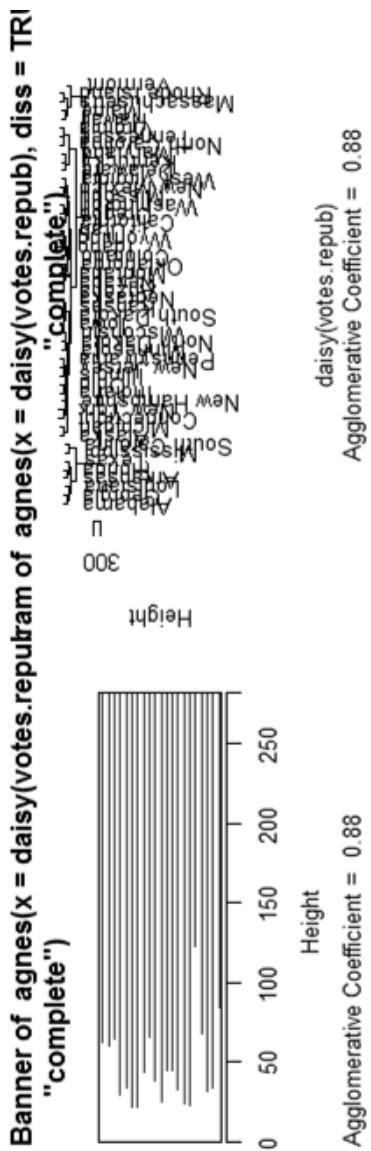
Agglomerative Coefficient = 0.8

**Dendrogram of agnes(x = votes.repub, metric = "manhattan", stand = TRUE)**



votes.repub  
Agglomerative Coefficient = 0.8

```
> op <- par(mfrow=c(2,2))
> agn2 <- agnes(daisy(votes.repub), diss = TRUE,
+     method = "complete")
> plot(agn2)
## alpha = 0.625 ==> beta = -1/4 is "recommended" by some
```



```
> agns <- agnes(votes.repub, method = "flexible",
+     par.meth = 0.625)
> plot(agns)
```

**Banner of agnes(x = daisy(votes.repub), method = "complete")**



Agglomerative Coefficient = 0.88  
daisy(votes.repub)

**Banner of agnes(x = votes.repub, method = "flexible")**  
**0.625**



Agglomerative Coefficient = 0.625  
daisy(votes.repub)

**Banner of agnes(x = votes.repub, method = "single")**  
**0.94**



Agglomerative Coefficient = 0.94  
votes.repub

```
par(op)
## "show" equivalence of three "flexible" special cases
> d.vr <- daisy(votes.repub)
> a.wgt <- agnes(d.vr, method = "weighted")
> a.sing <- agnes(d.vr, method = "single")
```

```

> a.comp <- agnes(d.vr, method = "complete")
> ic <- -(6:7) # not using 'call' and 'method' for
+     comparisons
> stopifnot(
+     all.equal(a.wgt [ic], agnes(d.vr, method="flexible",
+     par.method = 0.5)[ic]) ,
+     all.equal(a.sing[ic], agnes(d.vr, method="flex",
+     par.method= c(.5,.5,0, -.5))[ic]),
+     all.equal(a.comp[ic], agnes(d.vr, method="flex",
+     par.method= c(.5,.5,0, +.5))[ic]))

```

If you choose any height along the  $y$ -axis of the dendrogram, and move across the dendrogram counting the number of lines that you cross, each line represents a group that was identified when objects were joined together into clusters. The observations in that group are represented by the branches of the dendrogram that spread out below the line. For example, if we look at a height of 60, and move across the  $x$ -axis at that height, we'll cross two lines. That defines a two-cluster solution; by following the line down through all its branches, we can see the names of the states that are included in these two clusters. Since the  $y$ -axis represents how close together observations were when they were merged into clusters, clusters whose branches are very close together (in terms of the heights at which they were merged) probably aren't very reliable. But if there is a big difference along the  $y$ -axis between the last merged cluster and the currently merged one, which indicates that the clusters formed are probably doing a good job in showing us the structure of the data. Looking at the dendrogram for the voting data, there are (maybe not clearly) five distinct groups at the 20-level; at the 0-level there seems to be nine distinct groups.

For this data set, it looks like either five or six groups might be an interesting place to start investigating. This is not to imply that looking at solutions with more clusters would be meaningless, but the data seems to suggest that five or six clusters might be a good start. For a problem of this size, we can see the names of the states, so we could start interpreting the results immediately from the dendrogram, but when there are larger numbers of observations, this won't be possible.

```
## Exploring the dendrogram structure

> (d2 <- as.dendrogram(agn2)) # two main branches
'dendrogram' with 2 branches and 50 members total, at height
281.9508

> d2[[1]] # the first branch
'dendrogram' with 2 branches and 8 members total, at height
116.7048

> d2[[2]] # the 2nd one { 8 + 42 = 50 }
'dendrogram' with 2 branches and 42 members total, at height
178.4119

> d2[[1]][[1]]# first sub-branch of branch 1 .. and shorter
form
'dendrogram' with 2 branches and 6 members total, at height
72.92212

> identical(d2[[c(1,1)]], d2[[1]][[1]])
[1] TRUE

## a "textual picture" of the dendrogram :
str(d2)
```

```

--[dendrogram w/ 2 branches and 50 members at h = 282]
| --[dendrogram w/ 2 branches and 8 members at h = 117]
| | --[dendrogram w/ 2 branches and 6 members at h = 72.9]
| | | --[dendrogram w/ 2 branches and 3 members at h = 60.9]
| | | | --[dendrogram w/ 2 branches and 2 members at h = 48.2]
| | | | | --leaf "Alabama"
| | | | | --leaf "Georgia"
| | | | | --leaf "Louisiana"
| | | | --[dendrogram w/ 2 branches and 3 members at h = 58.8]
| | | | | --[dendrogram w/ 2 branches and 2 members at h = 56.1]
| | | | | | --leaf "Arkansas"
| | | | | | --leaf "Florida"
| | | | | --leaf "Texas"
| | | | --[dendrogram w/ 2 branches and 2 members at h = 63.1]
| | | | | --leaf "Mississippi"
| | | | | --leaf "South Carolina"
| | | | --[dendrogram w/ 2 branches and 42 members at h = 178]
| | | | | --[dendrogram w/ 2 branches and 37 members at h = 121]
| | | | | | --[dendrogram w/ 2 branches and 31 members at h = 80.5]
| | | | | | | --[dendrogram w/ 2 branches and 17 members at h = 64.5]
| | | | | | | | --[dendrogram w/ 2 branches and 13 members at h =
56.4]
| | | | | | | | | --[dendrogram w/ 2 branches and 10 members at h =
47.2]
| | | | | | | | | | --[dendrogram w/ 2 branches and 2 members at h
= 28.1]
| | | | | | | | | | | --leaf "Alaska"
| | | | | | | | | | | --leaf "Michigan"
| | | | | | | | | | --[dendrogram w/ 2 branches and 8 members at h
= 39.2]
| | | | | | | | | | | | --[dendrogram w/ 2 branches and 5 members
at h = 36.8]
| | | | | | | | | | | | --[dendrogram w/ 2 branches and 3 members
at h = 32.9]
| | | | | | | | | | | | | --[dendrogram w/ 2 branches and 2
members at h = 19.4]
| | | | | | | | | | | | | | --leaf "Connecticut"
| | | | | | | | | | | | | | --leaf "New York"
| | | | | | | | | | | | | | --leaf "New Hampshire"
| | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 2 members
at h = 20.2]
| | | | | | | | | | | | | | | --leaf "Indiana"
| | | | | | | | | | | | | | | --leaf "Ohio"
| | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 3 members
at h = 25.3]
| | | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 2 members
at h = 20.9]
| | | | | | | | | | | | | | | | | --leaf "Illinois"
| | | | | | | | | | | | | | | | --leaf "New Jersey"
| | | | | | | | | | | | | | | | --leaf "Pennsylvania"
| | | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 3 members at h =
42.2]
| | | | | | | | | | | | | | | | | | --leaf "Minnesota"
| | | | | | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 2 members at h
= 33.7]
| | | | | | | | | | | | | | | | | | | | --leaf "North Colorado"
| | | | | | | | | | | | | | | | | | | | --leaf "Wisconsin"
| | | | | | | | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 4 members at h =
37.5]
| | | | | | | | | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 2 members at h =
26.2]
| | | | | | | | | | | | | | | | | | | | | | | --leaf "Iowa"
| | | | | | | | | | | | | | | | | | | | | | | --leaf "South Colorado"
| | | | | | | | | | | | | | | | | | | | | | | | --[dendrogram w/ 2 branches and 2 members at h =
25.9]
| | | | | | | | | | | | | | | | | | | | | | | | | --leaf "Kansas"
| | | | | | | | | | | | | | | | | | | | | | | | | --leaf "Nebraska"

```

```

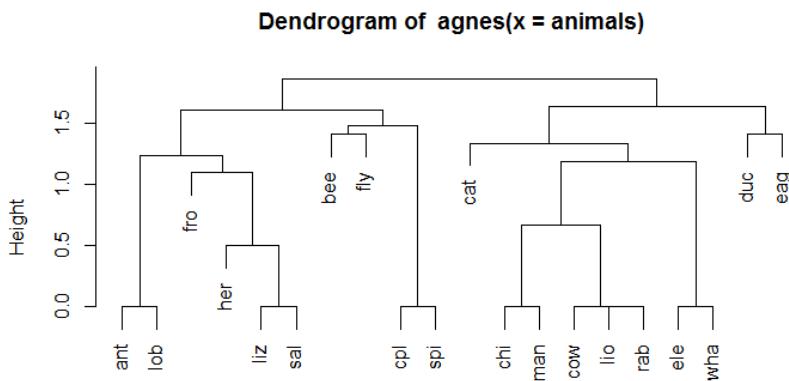
    |   |   `--[dendrogram w/ 2 branches and 14 members at h = 70.5]
    |   |   |   |--[dendrogram w/ 2 branches and 8 members at h = 48]
    |   |   |   |   |--[dendrogram w/ 2 branches and 4 members at h =
43.4]   |   |   |   |   |   |--[dendrogram w/ 2 branches and 3 members at h
= 27.8]   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members
at h = 23.4]   |   |   |   |   |   |   |   |--leaf "Arizona"
    |   |   |   |   |   |   |   |   |--leaf "Nevada"
    |   |   |   |   |   |   |   |   |--leaf "Montana"
    |   |   |   |   |   |   |   |   |--leaf "Oklahoma"
    |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 4 members at h =
43.7]   |   |   |   |   |   |   |   |   |   |--leaf "Colorado"
    |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 3 members at h
= 31.2]   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members
at h = 17.2]   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Idaho"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Wyoming"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Utah"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 6 members at h =
54.3]   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 3 members at h =
33.2]   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "California"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members at h
= 22.2]   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Oregon"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Washington"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 3 members at h =
35.1]   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members at h
= 21.1]   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Missouri"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "New Mexico"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "West Virginia"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 6 members at h =
66.8]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 3 members at h =
43.4]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Delaware"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members at h =
33.5]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Kentucky"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Maryland"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 3 members at h =
30.2]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members at h =
29.5]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "North Carolina"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Tennessee"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Virginia"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 5 members at h =
83.1]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 4 members at h =
55.4]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members at h =
32.8]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Hawaii"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Maine"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--[dendrogram w/ 2 branches and 2 members at h =
22.6]
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Massachusetts"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Rhode Island"
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |--leaf "Vermont"

```

Now, we need to interpret the results of this analysis. From the dendrogram we can see some logical clustering at the 0-level. For instance, California, Oregon and Washington are clustered together as we would expect. Also, at the 40-level Georgia, Alabama, Mississippi, Arkansas, South Carolina and Louisiana are grouped together. What other clusters make sense?

The next

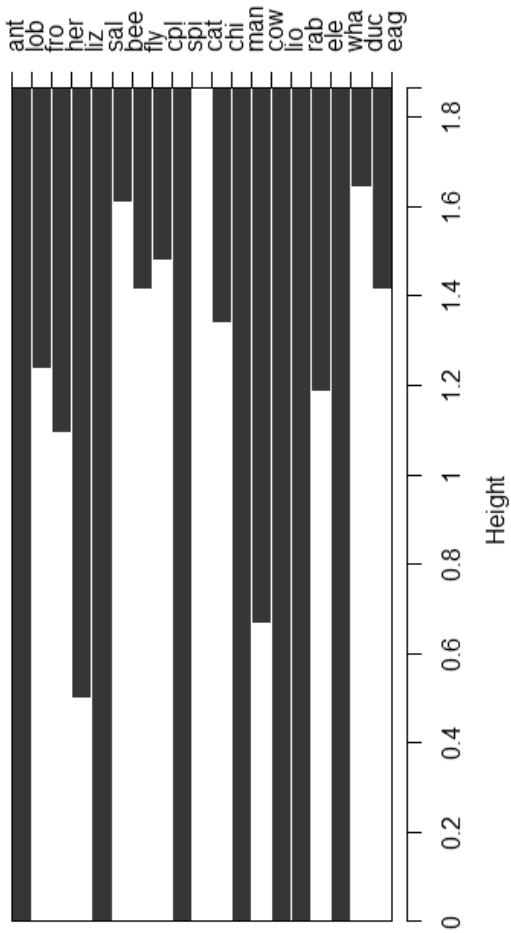
```
plot(agnes(animals), ask = TRUE)
> data(animals)
> aa.a <- agnes(animals) # default method = "average"
> aa.ga <- agnes(animals, method = "gaverage")
> op <- par(mfcol=1:2, mgp=c(1.5, 0.6, 0),
+           mar=c(.1+ > c(4,3,2,1)),cex.main=0.8)
> plot(aa.a, which.plot = 2)
```



animals  
Agglomerative Coefficient = 0.77

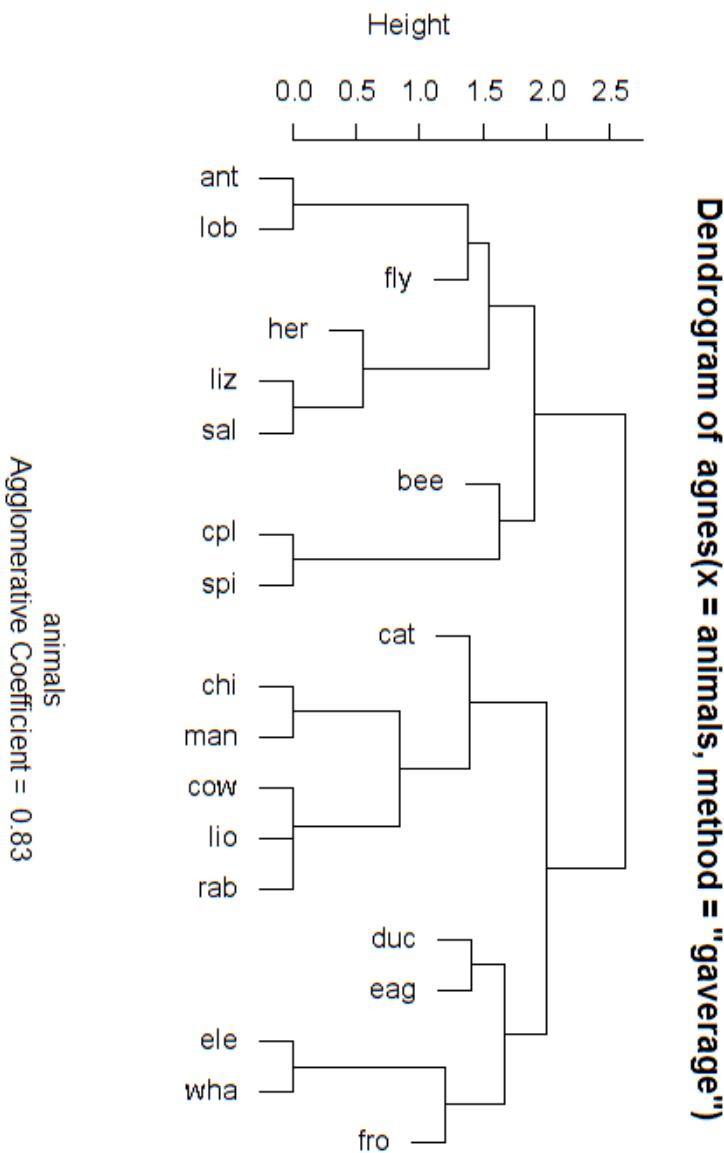
```
plot(agnes(agriculture), ask = TRUE)
Make a plot selection (or 0 to exit):
1: plot All
2: plot Banner
3: plot Clustering Tree
Selection:
Enter an item from the menu, or 0 to exit
Selection: 2
```

**Banner of agnes(x = animals)**



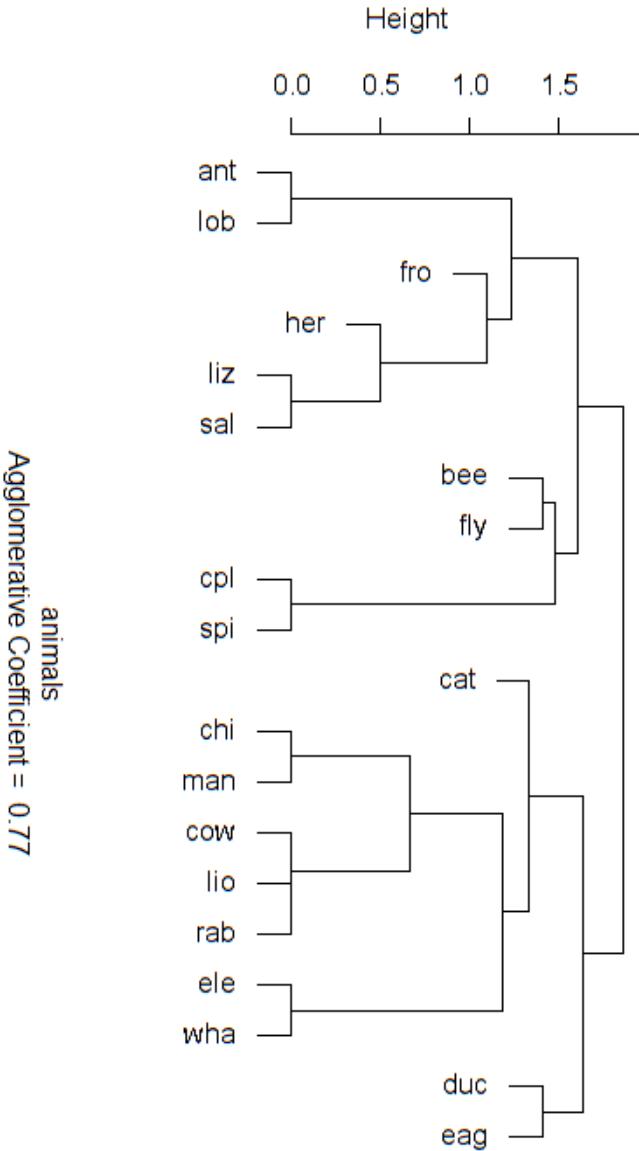
Agglomerative Coefficient = 0.77

```
> plot(aa.ga, which.plot = 2)
```



```
> par(op)
## Show how "gaverage" is a "generalized average":
> aa.ga.0 <- agnes(animals, method = "gaverage",
+     par.method = 0)
```

```
> stopifnot(all.equal(aa.ga.0[ic], aa.a[ic]))
> plot(aa.ga.0, which.plot=2) plot(aa.ga.0, which.plot=2)
```



Dendrogram of agnes(x = animals, method = "gaverage", par.method = 0)

We next introduce another popular clustering technique,  $k$ -means. The format of the  $k$ -means function in R is  $kmeans(x, centers)$  where  $x$  is a numeric dataset (matrix or data frame) and centers is the number of clusters to extract. The function returns the cluster memberships, centroids, sums of squares (within, between, total), and cluster sizes.

$K$ -means cluster analysis starts with  $k$  randomly chosen centroids, a different solution can be obtained each time the function is invoked. Use the `set.seed()` function to guarantee that the results are reproducible. Additionally, this clustering approach can be sensitive to the initial selection of centroids. The `kmeans()` function has an `nstart` option that attempts multiple initial configurations and reports on the best one. For example, adding `nstart=25` will generate 25 initial configurations. This approach is often recommended.

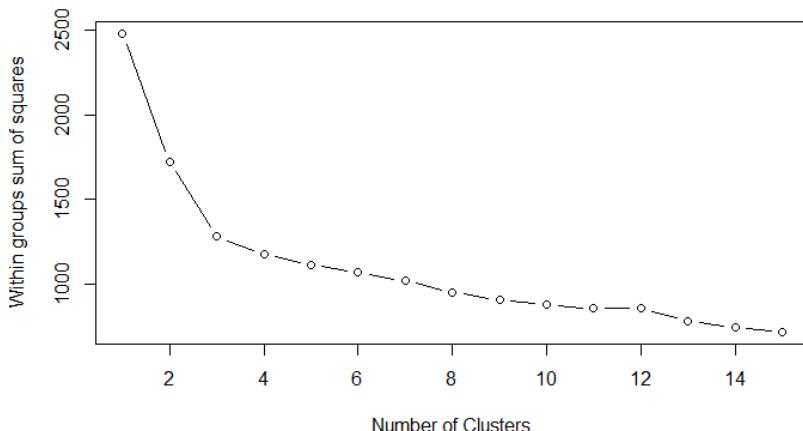
Unlike hierarchical clustering,  $k$ -means clustering requires that the number of clusters to extract be specified in advance. Here, a dataset containing 13 chemical measurements on 178 Italian wine samples is analyzed. The data originally come from the UCI Machine Learning Repository (<http://www.ics.uci.edu/~mlearn/MLRepository.html>) but we will access it via the rattle package.

```
> library(rattle)
> data(wine)
> head(wine)
```

|      | Type       | Alcohol | Malic | Ash  | Alcalinity | Magnesium | Phenols |
|------|------------|---------|-------|------|------------|-----------|---------|
|      | Flavanoids |         |       |      |            |           |         |
| 1    | 1          | 14.23   | 1.71  | 2.43 | 15.6       | 127       | 2.80    |
| 3.06 |            |         |       |      |            |           |         |
| 2    | 1          | 13.20   | 1.78  | 2.14 | 11.2       | 100       | 2.65    |
| 2.76 |            |         |       |      |            |           |         |
| 3    | 1          | 13.16   | 2.36  | 2.67 | 18.6       | 101       | 2.80    |
| 3.24 |            |         |       |      |            |           |         |
| 4    | 1          | 14.37   | 1.95  | 2.50 | 16.8       | 113       | 3.85    |
| 3.49 |            |         |       |      |            |           |         |
| 5    | 1          | 13.24   | 2.59  | 2.87 | 21.0       | 118       | 2.80    |
| 2.69 |            |         |       |      |            |           |         |
| 6    | 1          | 14.20   | 1.76  | 2.45 | 15.2       | 112       | 3.27    |
| 3.39 |            |         |       |      |            |           |         |

It looks like the variables are measured on different scales, so we will likely want to standardize the data before proceeding. The ‘`scale`’ function will do this. Additionally, a plot of the total within-groups sums of squares against the number of clusters in a  $k$ -means solution can be helpful. A bend in the graph can suggest the appropriate number of clusters. The graph can be produced by the following function. We also use the `NbClust` package here. `NbClust` package provides 30 indices for determining the number of clusters and proposes to user the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods.

```
> df <- scale(wine)
> wssplot(df)
```



```
> Library(NbClust)
> set.seed(1234)
> nc <- NbClust(df, min.nc=2, max.nc=15, method="kmeans")
*** : The Hubert index is a graphical method of determining the number of clusters.
```

In the plot of Hubert index, we seek a significant knee that corresponds to a significant increase of the value of the measure i.e. the significant peak in Hubert index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dindex second differences plot) that corresponds to a significant increase of the value of the measure.

All 178 observations were used.

\*\*\*\*\*

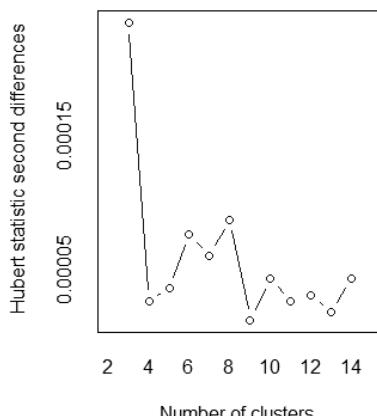
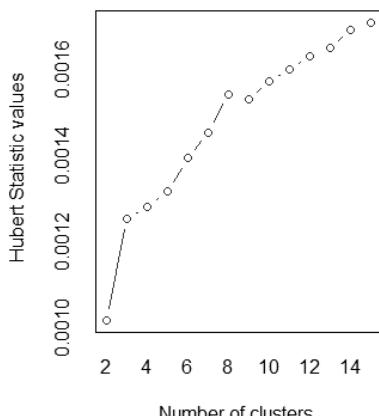
\* Among all indices:

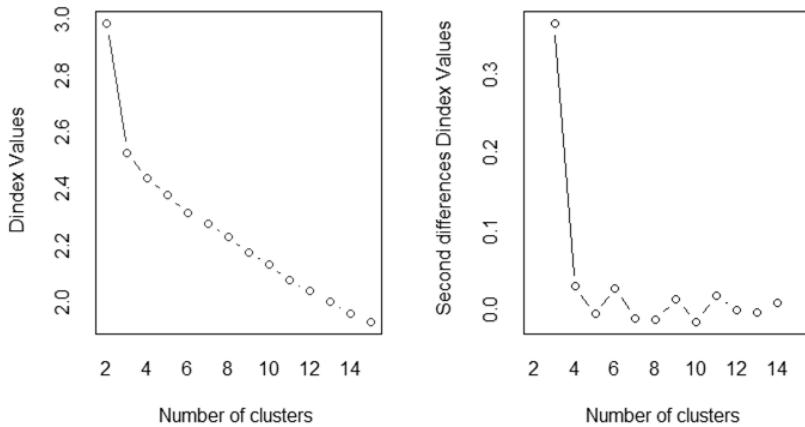
- \* 4 proposed 2 as the best number of clusters
- \* 16 proposed 3 as the best number of clusters
- \* 1 proposed 11 as the best number of clusters
- \* 2 proposed 15 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

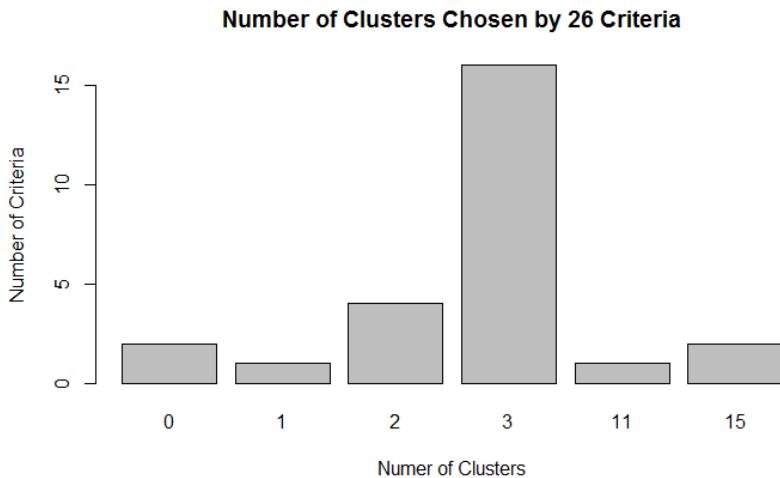
\* According to the majority rule, the best number of clusters is 3

\*\*\*\*\*





```
> table(nc$Best.n[1,])
0 1 2 3 11 15
2 1 4 16 1 2
> barplot(table(nc$Best.n[1,]),
+         xlab="Number of Clusters", ylab="Number of Criteria",
+         main="Number of Clusters Chosen by 26 Criteria")
```



```

> set.seed(1234)
> fit.km <- kmeans(df, 3, nstart=25)

> fit.km$size
[1] 61 68 49

> fit.km$centers

          v1           v2           v3           v4           v5
v6      v7           v8
1 -1.16822514  0.8756272 -0.3037196  0.3180446 -0.6626544
0.56329925  0.87403990  0.94098462
2  0.07973544 -0.9195318 -0.3778231 -0.4643776  0.1750133 -
0.46892793 -0.07372644  0.04416309
3  1.34366784  0.1860184  0.9024258  0.2485092  0.5820616 -
0.05049296 -0.98577624 -1.23271740

          v9           v10          v11          v12          v13
v14
1 -0.583942581  0.58014642  0.1667181  0.4823674  0.7648958
1.1550888
2  0.008736157  0.01821349 -0.8598525  0.4233092  0.2490794
-0.7630972
3  0.714825281 -0.74749896  0.9857177 -1.1879477 -1.2978785
-0.3789756

> aggregate(wine[-1], by=list(cluster=fit.km$cluster), mean)

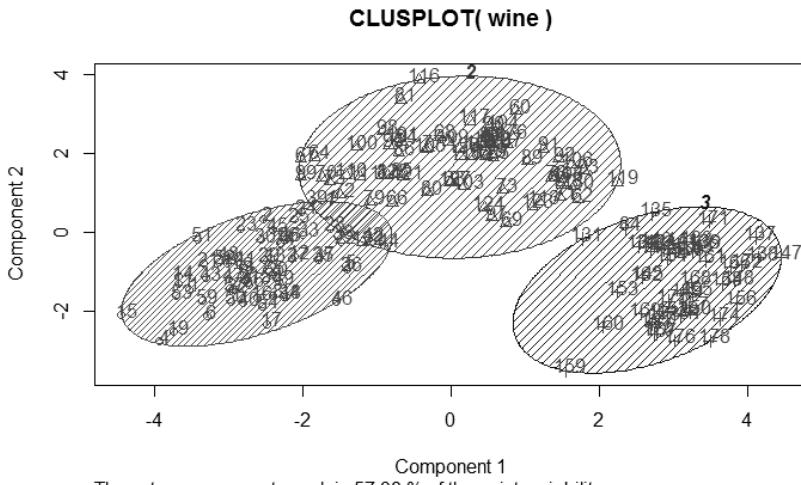
    cluster      v2      v3      v4      v5      v6
v7      v8      v9      v10     v11
1           1 13.71148 1.997049 2.453770 17.28197 107.78689
2.842131 2.9691803 0.2891803 1.922951 5.444590
2           2 12.25412 1.914265 2.239118 20.07941   93.04412
2.248971 2.0733824 0.3629412 1.601324 3.064706
3           3 13.15163 3.344490 2.434694 21.43878   99.02041
1.678163 0.7979592 0.4508163 1.163061 7.343265

      v12      v13      v14
1 1.0677049 3.154754 1110.6393
2 1.0542059 2.788529  506.5882
3 0.6859184 1.690204  627.5510

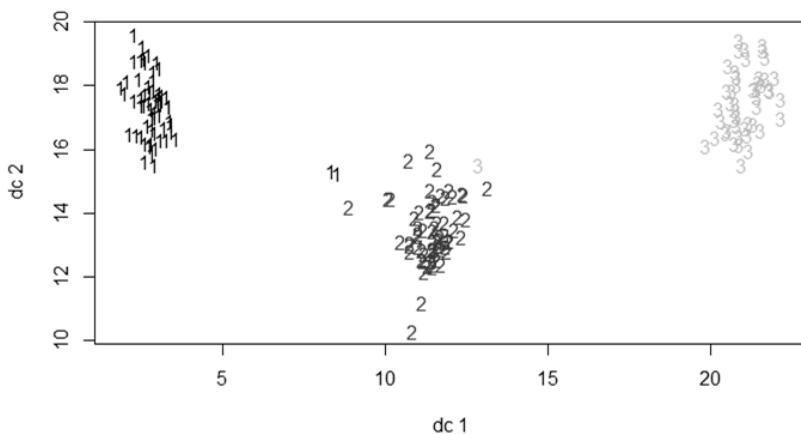
```

Two additional cluster plot may be useful in your analysis.

```
> clusplot(wine, fit.km$cluster, color=TRUE, shade=TRUE,  
+           labels=2, lines=0)
```



```
> plotcluster(wine, fit.km$cluster)
```



Finally, there is an R specific Internet search engine at <http://www.rseek.org> for more assistance.



# 11.What is Data Fitting?

## 11.1. Introduction

You had a preview of this chapter in Chapter 9. Fitting distributions consists in finding a mathematical function which represents in a good way a statistical variable. A statistician often is facing with this problem: he has some observations of a quantitative character  $x_1, x_2, \dots, x_n$  and he wishes to test if those observations, being a sample of an unknown population, belong from a population with a pdf (probability density function)  $f(x, \theta)$ , where  $\theta$  is a vector of parameters to estimate with available data.

We can identify 4 steps in fitting distributions:

1. Model/function choice: hypothesize families of distributions;
2. Estimate parameters;
3. Evaluate quality of fit;
4. Goodness-of-fit statistical tests.

This chapter aims to face fitting distributions dealing shortly with theoretical issues and practical ones using the statistical environment and language R.

R is a language and an environment for statistical computing and graphics flexible and powerful. We are going to use some R statements concerning graphical techniques (§ 2.0), model/function choice (§ 3.0), parameters estimate (§ 4.0), measures of goodness-of-fit (§ 5.0) and most common goodness-of-fit tests (§ 6.0).

To understand this work a basic knowledge of R is needed. We suggest a reading of “An introduction to R”. R statements, if not specified, are included in stats package.

## 11.2. Graphics

Exploratory data analysis can be the first step, getting descriptive statistics (mean, standard deviation, skewness, kurtosis, etc.) and using

graphical techniques (histograms, density estimate, ECDF) which can suggest the kind of pdf to use to fit the model.

We can obtain samples from some pdf (such as Gaussian, Poisson, Weibull, gamma, etc.) using R statements and after we draw a histogram of these data.

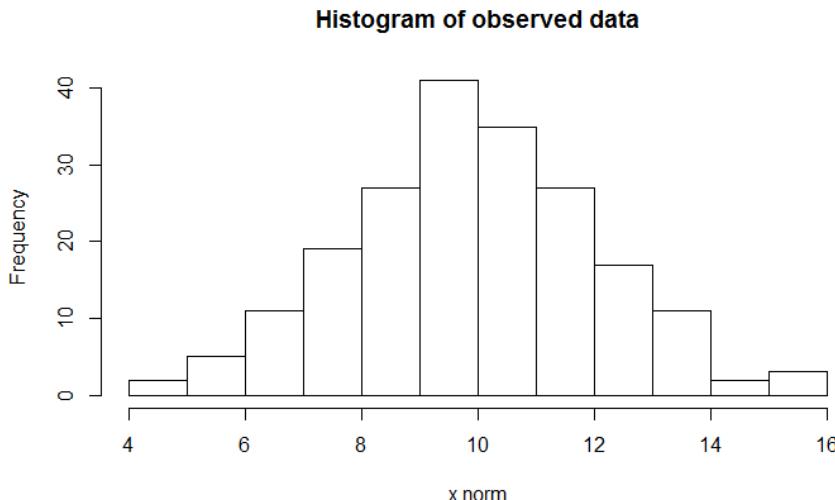
### EXAMPLE 1. Fitting a Normal Distribution

Suppose we have a sample of size  $n=100$  belonging from a normal population  $N(10,2)$  with mean=10 and standard deviation=2:

```
x.norm<-rnorm(n=200,m=10,sd=2)
```

We can get a histogram using `hist()` statement (Fig. 1):

```
hist(x.norm,main="Histogram of observed data")
```



Histograms can provide insights on skewness, behavior in the tails, presence of multi-modal behavior, and data outliers; histograms can be compared to the fundamental shapes associated with standard analytic distributions.

We can estimate frequency density using `density()` and `plot()` to plot the graphic (Fig. 2):

```
plot(density(x.norm),main="Density estimate of data")
```

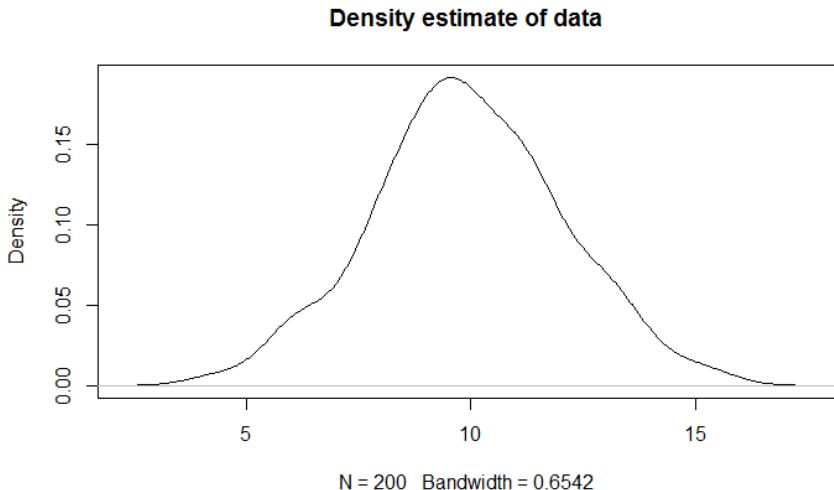
R allows to compute the empirical cumulative distribution function by `ecdf()` (Fig. 3):

```
plot(ecdf(x.norm), main=" Empirical cumulative distribution function")
```

A Quantile-Quantile (Q-Q) plot<sup>3</sup> is a scatterplot comparing the fitted and empirical distributions in terms of the dimensional values of the variable (i.e., empirical quantiles). It is a graphical technique for determining if a data set come from a known population. In this plot on the y-axis we have empirical quantiles e on the x- axis we have the ones got by the theoretical model.

R offers to statements: `qqnorm()`, to test the goodness-of-fit of a Gaussian distribution, or `qqplot()` for any kind of distribution. In our example we have (Fig. 4):

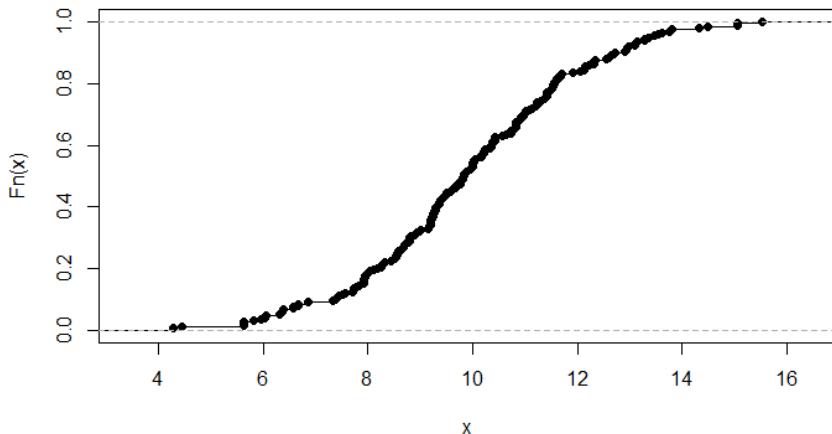
```
z.norm<-(x.norm-mean(x.norm))/sd(x.norm)
## standardized data qqnorm(z.norm) drawing the QQplot
abline(0,1) ## drawing a 45-degree reference line
abline(0,1) ## drawing a 45-degree reference line
```



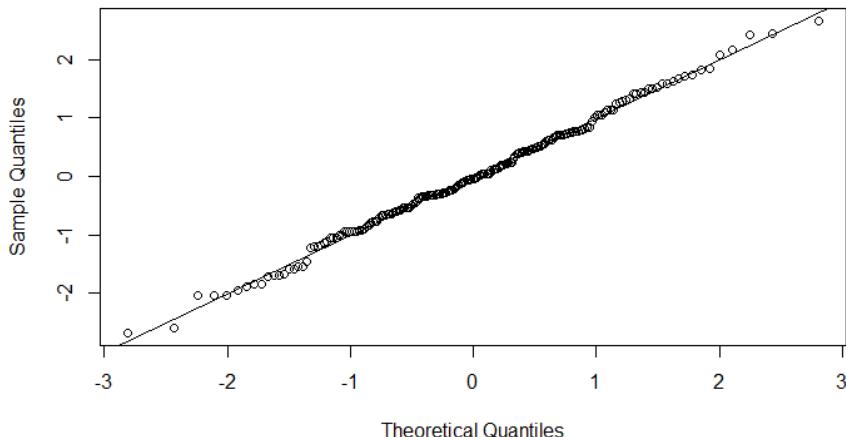
A 45-degree reference line is also plotted. If the empirical data come from the population with the chosen distribution, the points should fall approximately along this reference line. The greater the departure from

this reference line, the greater the evidence for the conclusion that the data set have come from a population with a different distribution.

**Empirical cumulative distribution function**



**Normal Q-Q Plot**



### EXAMPLE 2. Fitting other Distributions

If data differ from a normal distribution (i.e. data belonging from a Weibull pdf) we can use `qqplot()` in this way (Fig. 5):

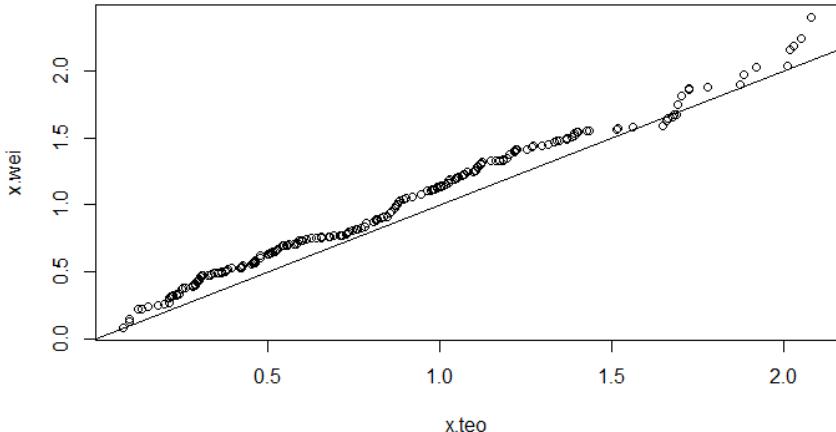
```
## sampling from a weibull distribution with parameters  
## shape=2.1 and scale=1.1  
x.wei<-rweibull(n=200,shape=2.1,scale=1.1)
```

```

## theoretical quantiles from a weibull population with
## known parameters shape=2 e scale=1
x.teo<-rweibull(n=200,shape=2,scale=1)
## QQ-plot abline(0,1) a 45-degree reference line is
## plotted
qqplot(x.teo,x.wei,main="QQ-plot distr. Weibull")

```

QQ-plot distr. Weibull



where  $x.\text{wei}$  is the vector of empirical data, while  $x.\text{teo}$  are quantiles from theoretical model.

### 11.3. Model choice

The first step in fitting distributions consists in choosing the mathematical model or function to represent data in the better way.

Sometimes the type of model or function can be argued by some hypothesis concerning the nature of data, often histograms and other graphical techniques can help in this step (see § 2.0), but graphics could be quite subjective, so there are methods based on analytical expressions such as the **Pearson's K criterion**. Solving a particular differential equation we can obtain several families of function able to represent quite all empirical distributions. Those curves depend only by mean, variability, skewness and kurtosis. Standardizing data, the type of curve depends only by skewness and kurtosis measures as shown in this formula:

$$K = \frac{\gamma_1^2(\gamma_2 + 6)^2}{4(4\gamma_2 - 3\gamma_1^2 + 12)(2\gamma_2 - 3\gamma_1^2)}$$

Where (11.1)

$$\gamma_1 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n\sigma^2}$$

is Pearson's skewness coefficient. (11.2)

$$\gamma_2 = \frac{\sum_{i=1}^n (x_i - \mu)^4}{n\sigma^4} - 3$$

is Pearson's kurtosis coefficient. (11.3)

According to the value of  $K$ , obtained by available data, we have a particular kind of function. Here are some examples of continuous and discrete distributions, they will be used afterwards in this paper. For each distribution there is the graphic shape and R statements to get graphics.

### EXAMPLE 3. Fitting a Discrete Distribution

Dealing with discrete data we can refer to Poisson's distribution (Figure 6) with probability mass function:

$$f(x, \lambda) = e^{-\lambda} \frac{\lambda^x}{x!}, \text{ where } x = 0, 1, 2, \dots \quad (11.4)$$

```
x.poi<-rpois(n=200,lambda=2.5)
hist(x.poi,main="Poisson distribution")
```

As concern continuous data we have:

normal (Gaussian) distribution (Figure 7):

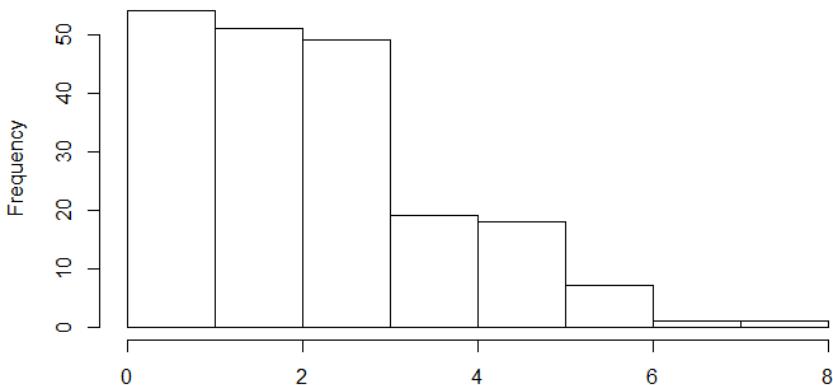
$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{ with } x \in \mathcal{R}. \quad (11.5)$$

```
curve(dnorm(x,m=10,sd=2),from=0,to=20,main="Normal
distribution")
```

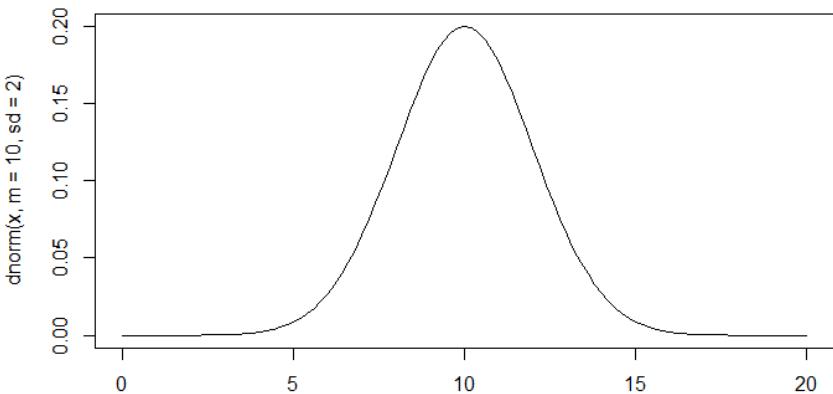
gamma distribution (Figure 8):

$$f(x, \alpha, \lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}, \text{ with } x \in \mathcal{R}^+. \quad (11.6)$$

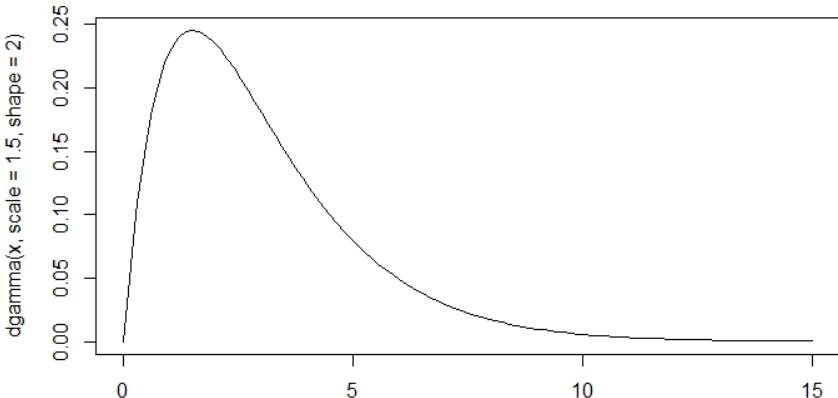
```
curve(dgamma(x, scale=1.5, shape=2), from=0, to=15,  
main="Gamma distribution")
```



**Figure 6.** A Poisson Distribution



**Figure 7.** A normal (Gaussian) distribution



**Figure 8.** A Gamma Distribution

Weibull distribution (Figure 9):

$$f(x, \alpha, \beta) = \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(\frac{x}{\beta})^\alpha}, \text{ with } x \in \mathbb{R}^+. \quad (11.7)$$

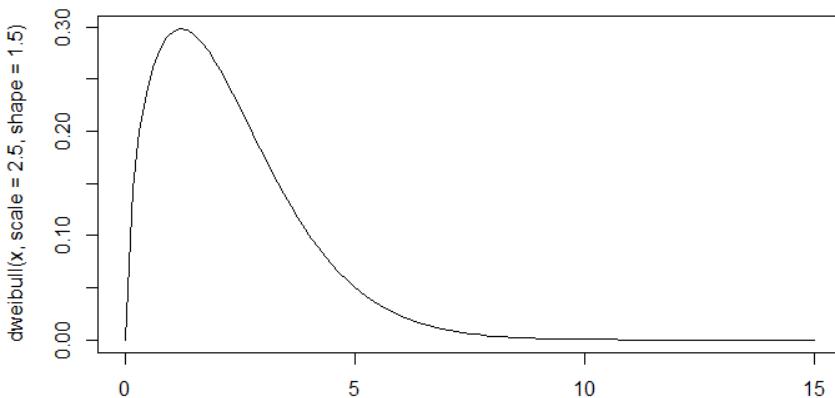
**EXAMPLE 4.** Weibull Distribution

Suppose we have a Weibull distribution with scale parameter 2.5 and shape parameter 1.5.

```
curve(dweibull(x, scale=2.5, shape=1.5), from=0, to=15,  
main="weibull distribution")
```

To compute skewness and kurtosis index we can use those statements: `skewness()` and `kurtosis()` included in `fBasics` package (you need to download this package from CRAN website):

```
library(fBasics) ## package loading  
skewness(x.norm) ## skewness of a normal distribution  
[1] 0.1242952  
kurtosis(x.norm) ## kurtosis of a normal distribution  
[1] 0.01372539  
skewness(x.wei) ## skewness of a weibull distribution  
[1] 0.7788843  
kurtosis(x.wei) ## kurtosis of a weibull distribution  
[1] 0.4331281
```



**Figure 9.** A Weibull Distribution

## 11.4. Parameters' estimate

After choosing a model that can mathematically represent our data we have to estimate parameters of such model. There are several estimate methods in statistical literature, but in this paper we are focusing on these ones:

1. analogic
2. moments
3. maximum likelihood

Analogic method consists in estimating model parameters applying the same function to empirical data. I.e., we estimate the unknown mean of a normal population using the sample mean:

```
mean.hat<-mean(x.norm)  
mean.hat  
[1] 9.935537
```

The **method of moments** is a technique for constructing estimators of the parameters that is based on matching the sample moments with the corresponding distribution moments. This method equates sample moments to population (theoretical) ones. When moment methods are available, they have the advantage of simplicity. We define sample (empirical) moments in this way:

$t$ -th sample moment about 0:

$$m_t = \sum_{i=1}^n x_i^t y_i, \quad t = 0, 1, 2, \dots$$

$t$ -th sample moment about the mean: (11.8)

$$m'_t = \sum_{i=1}^n (x_i - \mu)^t y_i, \quad t = 0, 1, 2, \dots$$

while theoretical (population) ones: (11.9)

$t$ -th sample moment about 0:

$$m_t^* = \int_{\beta}^{\alpha} x^t f(x, \theta) dx, \quad t = 0, 1, 2, \dots$$

$t$ -th sample moment about the mean: (11.10)

$$m_t'^* = \int_{\beta}^{\alpha} (x_i - \mu)^t f(x, \theta) dx, \quad t = 0, 1, 2, \dots$$

(11.11)

where  $\beta - \alpha$  is the range where  $f(x, \theta)$  is defined,  $\mu$  is the mean of the distribution, and  $y_i$  are empirical relative frequencies. I.e., we shall estimate parameters of a gamma distribution using the method of moments considering the first moment about 0 (mean) and the second moment about mean (variance):

$$\bar{x} = \frac{\alpha}{\lambda} \quad \text{and} \quad s^2 = \frac{\alpha}{\lambda^2}$$

(11.12) and (11.13)

where on the left there mean and variance of gamma distribution and on the right sample mean and sample corrected variance. Solving we can get parameters' estimates:

$$\hat{\lambda} = \frac{\bar{x}}{s^2} \quad \text{and} \quad \hat{\alpha} = \frac{\bar{x}^2}{s^2}$$

(11.14) and (11.15)

### EXAMPLE 5. A Gamma Distribution

Suppose we have a gamma distribution with rate parameter 0.5 and shape parameter 3.5.

```
x.gam<-rgamma(200,rate=0.5,shape=3.5) ## sampling from
a gamma distribution with λ=0.5 (scale parameter) and
α=3.5 (shape parameter)

med.gam<-mean(x.gam) ## sample mean
var.gam<-var(x.gam) ## sample variance

l.est<-med.gam/var.gam ## lambda estimate (corresponds to
rate)

a.est<-((med.gam)^2)/var.gam ## alfa estimate
```

```

1.est
[1] 0.5625486 a.est
[1] 3.916339

```

The **method of maximum likelihood** is used in statistical inference to estimate parameters. We have a random variable with a known pdf  $f(x, \theta)$  describing a quantitative character in the population. We should estimate the vector of constant and unknown parameters  $\theta$  according to sampling data:  $x_1, x_2, \dots, x_n$ . Maximum likelihood estimation begins with the mathematical expression known as a likelihood function of the sample data. Loosely speaking, the likelihood of a set of data is the probability of obtaining that particular set of data given the chosen probability model. This expression contains the unknown parameters. Those values of the parameter that maximize the sample likelihood are known as the maximum likelihood estimates (MLE). We define the likelihood function as:

$$L(x_1, x_2, \dots, x_n, \theta) = \prod_{i=1}^n f(x_i, \theta) \quad (11.16)$$

MLE consist in finding  $\theta$  which maximizes  $L(x_1, x_2, \dots, x_n, \theta)$  or its logarithmic function. We can employ mathematical analysis methods (partial derivatives equal to zero) when the likelihood function is rather simple, but very often we optimize  $L(x_1, x_2, \dots, x_n, \theta)$  using iterative methods. MLE have several statistical properties and advantages.

For instance, in case of a gamma distribution, the likelihood function is:

$$\begin{aligned} L(x_1, x_2, \dots, x_n, \alpha, \lambda) &= \prod_{i=1}^n f(x_i, \alpha, \lambda) = \prod_{i=1}^n \frac{\lambda^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\lambda x_i} \\ &= \left( \frac{\lambda^\alpha}{\Gamma(\alpha)} \right) \left( \prod_{i=1}^n x_i \right)^{\alpha-1} e^{-\lambda \sum_{i=1}^n x_i} \end{aligned}$$

while its logarithmic is:

(11.17)

$$\log(L) = n\alpha \log(\lambda) - n \log(\Gamma(\alpha)) + (\alpha - 1) \sum_{i=1}^n \log x_i - \lambda \sum_{i=1}^n x_i$$

(11.18)

In R environment we can get MLE by two statements:

1. `mle()` included in package `stats4`
2. `fitdistr()` included in package `MASS`

`mle()` allows to fit parameters by maximum likelihood method using iterative methods of numerical calculus to minimize the negative log-likelihood (which is the same of maximizing the log-likelihood). You have to specify the negative log-likelihood analytical expression as argument and giving some starting parameters estimates. In case of a gamma distribution:

```
## loading package stats4 l1<-function(lambda,alfa) {n<-200
library(stats4)
x<-x.gam
-n*alfa*log(lambda)+n*log(gamma(alfa))-(alfa-
1)*sum(log(x))+lambda*sum(x)}## -log-likelihood function
est<-mle(minuslog=l1,start=list(lambda=2,alfa=1))
summary(est)

Maximum likelihood estimation
Call:
mle(minuslog1 = l1, start = list(lambda = 2, alfa = 1))

Coefficients:
Estimate Std. Error
lambda      0.5290189   0.05430615
alfa        3.6829126   0.35287672
-2 log L: 1044.282
```

We supply as starting values of parameters estimates arbitrary ones, but, we could use estimates got by the methods of moments. The statement `mle()` allows to estimate parameters for every kind of pdf, it needs only to know the likelihood analytical expression to be optimized.

In MASS package is available `fitdistr()` for maximum-likelihood fitting of univariate distributions

without any information about likelihood analytical expression. It is enough to specify a data vector, the type of pdf (`densfun`) and eventually the list of starting values for iterative procedure (`start`).

```
library(MASS) ## loading package MASS
fitdistr(x.gam,"gamma") ## fitting gamma pdf parameters
shape      rate
3.68320097  0.52910229
(0.35290545) (0.05431458)
## fitting weibull pdf parameters scale shape
fitdistr(x.wei,densfun=dweibull,start=list(scale=1,shape=2))
)
1.04721828  2.04959159 (0.03814184) (0.11080258)
## fitting gaussian pdf parameters mean sd
fitdistr(x.norm,"normal")
9.9355373  2.0101691 (0.1421404) (0.1005085)
```

## 11.5. Measures of goodness-of-fit

A goodness-of-fit measure is useful for matching empirical frequencies with fitted ones by a theoretical model. We have absolute and relative measures. Among the absolute ones we can choice:

$$\xi = \frac{\sum_{i=1}^n |y_i - y_i^*|}{n} \quad \text{and} \quad {}^2\xi = \sqrt{\frac{\sum_{i=1}^n (y_i - y_i^*)^2}{n}}$$

(11.19) and (11.20)

where  $y_i$  are empirical frequencies and  $y_i^*$  are fitted ones.

Some relative measures are below:

$$\delta = \frac{\xi}{\sum_{i=1}^n y_i / n} = \frac{\sum_{i=1}^n |y_i - y_i^*|}{\sum_{i=1}^n y_i}$$

$$^2\delta = \frac{^2\xi}{\sum_{i=1}^n y_i/n} = \sqrt{\frac{\sum_{i=1}^n (y_i - y_i^*)^2}{n}} / \frac{\sum_{i=1}^n y_i/n}{\sum_{i=1}^n y_i/n}$$

$$^2\delta = \frac{^2\xi}{\sqrt{\sum_{i=1}^n y_i/n}} = \sqrt{\frac{\sum_{i=1}^n (y_i - y_i^*)^2}{\sum_{i=1}^n y_i^2}}$$

(11.22), (11.23) and (11.24)

Usually those indexes are expressed by a percent measure of the corresponding mean.

#### EXAMPLE 6. Using R for Count Data

Here is an example using R for count data (Poisson distribution):

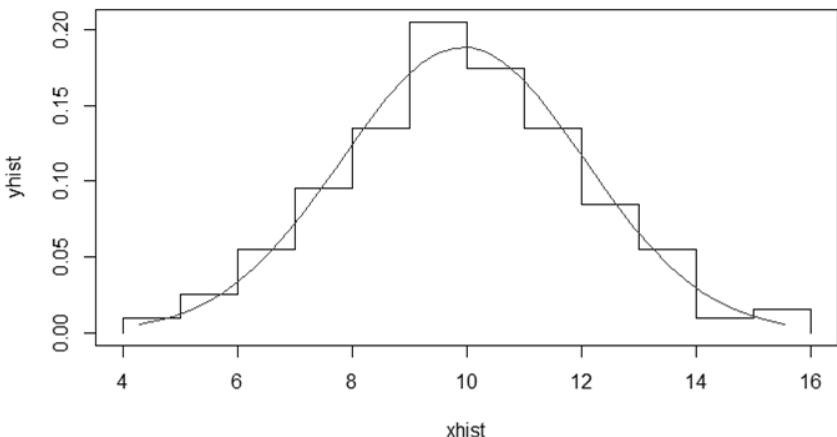
```
lambda.est<-mean(x.poi) ## estimate of parameter lambda
tab.os<-table(x.poi)## table with empirical frequencies
tab.os

0 1 2 3 4 5          6 7 8
21 29 46 53 28 16    4   2   1

x.poi
freq.os<-vector()
for(i in 1: length(tab.os)) freq.os[i]<-tab.os[[i]] ## vector of empirical frequencies
freq.ex<-(dpois(0:max(x.poi),lambda=lambda.est)*200)      ## vector of fitted (expected) frequencies freq.os
[1] 21 29 46 53 28 16 4 2 1
freq.ex
[1] 15.0040080 38.8603808 50.3241931 43.4465534
28.1316433 14.5721912
6.2903292
[8] 2.3274218 0.7535028
acc<-mean(abs(freq.os-trunc(freq.ex))) ## absolute goodness-of-fit index acc
[1] 2.111111
acc/mean(freq.os)*100 ## relative (percent) goodness-of-fit index
[1] 17
```

A graphical technique to evaluate the goodness-of-fit can be drawing pdf curve and histogram together (Fig. 10):

```
h<-hist(x.norm, breaks=15)
xhist<-c(min(h$breaks), h$breaks)
yhist<-c(0, h$density, 0)
xfit<-seq(min(x.norm), max(x.norm), length=40)
yfit<-dnorm(xfit, mean=mean(x.norm), sd=sd(x.norm))
plot(xhist, yhist, type="s", ylim=c(0, max(yhist, yfit)),
main="Normal pdf and histogram")
lines(xfit, yfit, col="red")
```



**Figure 10.** A Normal PDF and Histogram

## 11.6. Goodness of fit tests

Goodness-of-fit tests indicate whether or not it is reasonable to assume that a random sample comes from a specific distribution. They are a form of hypothesis testing where the null and alternative hypotheses are:

$H_0$ : Sample data come from the stated distribution

$H_A$ : Sample data do not come from the stated distribution

These tests are sometimes called as omnibus tests and they are distribution free, meaning they do not depend according the pdf. We shall point out our attention to normality tests.

The **Chi-square test** is the oldest goodness-of-fit test dating back to Karl Pearson (1900). The test may be thought of as a formal comparison of a histogram with the fitted density.

An attractive feature of the Chi-square ( $\chi^2$ ) goodness-of-fit test is that it can be applied to any univariate distribution for which you can calculate the cumulative distribution function. The chi-square goodness-of-fit test is applied to binned data (i.e., data put into classes). This is actually not a restriction since for non-binned data you can simply calculate a histogram or frequency table before generating the chi-square test. However, the value of the chi-square test statistic is dependent on how the data is binned. Another disadvantage of this test is that it requires a sufficient sample size in order for the chi square approximation to be valid. The chi-square goodness-of-fit test can be applied either to discrete distributions or continuous ones while the Kolmogorov-Smirnov and Anderson-Darling tests are restricted to continuous distributions. Estimating model parameters with sample is allowed with this test. The chi-square test is defined for the hypothesis:

$H_0$ : the data follow a specified distribution

$H_A$ : the data do not follow the specified distribution

For the chi-square goodness-of-fit computation, the data are divided into  $k$  bins and the test statistic is defined in this way:

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \quad (11.25)$$

where  $O_i$  is the observed frequency for bin  $i$  and  $E_i$  is the expected frequency for bin  $i$ . The expected frequency is calculated by cumulative distribution function. This statistic is distributed as a  $\chi^2$  random variable with  $k - p - 1$  degrees of freedom ( $p$  is the number of parameters estimated by sample data). The hypothesis that the data are from a population with the specified distribution is accepted if  $\chi^2$  is lower than the chi-square percent point function with  $k - p - 1$  degrees of

freedom and a significance level of  $\alpha$ . The chi-square test is sensitive to the choice of bins.

In R environment there are three ways to perform a Chi-square test.

In case of count data we can use `goodfit()` included in `vcd` package (to be downloaded from CRAN website):

```
library(vcd)## loading vcd package  
gf<-goodfit(x.poi,type= "poisson",method= "minchisq")  
summary(gf)
```

Goodness-of-fit test for Poisson distribution

```
x^2 df  P(> X^2) Pearson 8.378968 7 0.3003653  
plot(gf,main="Count data vs Poisson distribution")
```

In case of a continuous variable, such as a gamma distribution as in the following example, with parameters estimated by sample data:

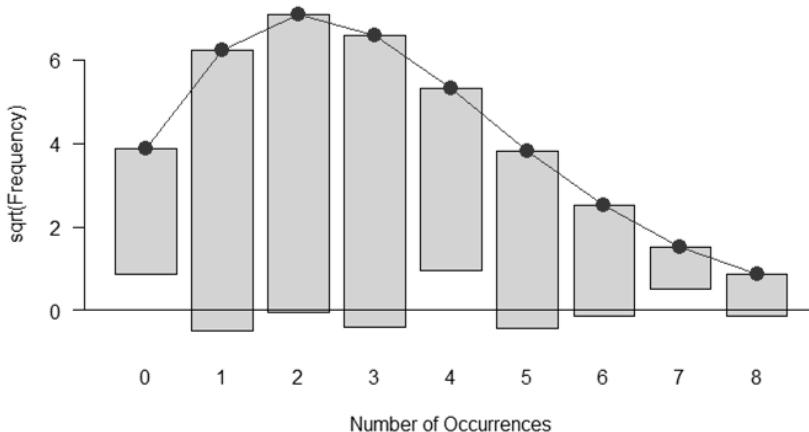
```
x.gam.cut<-cut(x.gam,breaks=c(0,3,6,9,12,18))      ##binning  
data.table(x.gam.cut) ## binned data table  
  
x.gam.cut  
(0,3]   (3,6]   (6,9]   (9,12]  (12,18]  
26       64       60       27       23  
  
## computing expected frequencies  
(pgamma(3,shape=a.est,rate=l.est)-  
pgamma(0,shape=a.est,rate=l.est))*200  
[1] 19.95678  
  
(pgamma(6,shape=a.est,rate=l.est)-  
pgamma(3,shape=a.est,rate=l.est))*200  
[1] 70.82366  
  
(pgamma(9,shape=a.est,rate=l.est)-  
pgamma(6,shape=a.est,rate=l.est))*200  
[1] 60.61188  
  
(pgamma(12,shape=a.est,rate=l.est)-  
pgamma(9,shape=a.est,rate=l.est))*200  
[1] 30.77605  
  
(pgamma(18,shape=a.est,rate=l.est)-  
pgamma(12,shape=a.est,rate=l.est))*200  
[1] 16.12495  
f.ex<-c(20,71,61,31,17) ## expected frequencies vector  
f.os<-vector()
```

```

for(i in 1:5) f.os[i]<- table(x.gam.cut)[[i]] ## empirical
frequencies
x2<-sum(((f.os-f.ex)^2)/f.ex) ## chi-square statistic
gdl<-5-2-1 ## degrees of freedom
1-pchisq(x2,gdl) ## p-value
[1] 0.07652367

```

$H_0$  is accepted as the  $p$ -value is greater than of a significance level fixed at least in 5%.



**Figure 11.** Count Data versus the Poisson Distribution

Whether we are dealing with a continuous variable and all its pdf parameters are known we can use chisq.test():

```

## computing relative expected frequencies
p<-c((pgamma(3,shape=3.5,rate=0.5)-
pgamma(0,shape=3.5,rate=0.5)),
(pgamma(6,shape=3.5,rate=0.5)-
pgamma(3,shape=3.5,rate=0.5)),
(pgamma(9,shape=3.5,rate=0.5)-
pgamma(6,shape=3.5,rate=0.5)),
(pgamma(12,shape=3.5,rate=0.5)-
pgamma(9,shape=3.5,rate=0.5)),
(pgamma(18,shape=3.5,rate=0.5)-
pgamma(12,shape=3.5,rate=0.5)))
chisq.test(x=f.os,p=p) ## chi-square test
Chi-squared test for given probabilities data: f.os
X-squared = 2.8361, df = 4, p-value = 0.5856

```

We can't reject null hypothesis as p-value is rather high, so probably sample data belong from a gamma distribution with shape parameter=3.5 and rate parameter=0.5.

The **Kolmogorov-Smirnov test** is used to decide if a sample comes from a population with a specific distribution. It can be applied both for discrete (count) data and continuous binned (even if some Authors do not agree on this point) and both for continuous variables. It is based on a comparison between the empirical distribution function (ECDF) and the theoretical one defined as  $F(x) = \int_{-\infty}^x f(y, \theta) dy$ , where  $f(y, \theta)$  is the pdf. Given  $n$  ordered data points  $X_1, X_2, \dots, X_n$ , the ECDF is defined as:

$$F_n(x_i) = \frac{N(i)}{n} \quad (11.26)$$

where  $N(i)$  is the number of points less than  $X_i$  ( $X_i$  are ordered from smallest to largest value). This is a step function that increases by  $1/n$  at the value of each ordered data point.

The test statistic used is:

$$D_n = \sup_{1 \leq i \leq n} |F(x_i) - F_n(x_i)| \quad (11.27)$$

that is the upper extreme among absolute value differences between ECDF and theoretical CDF. The hypothesis regarding the distributional form is rejected if the test statistic,  $D_n$ , is greater than the critical value obtained from a table, or, which is the same, if the p-value is lower than the significance level. Kolmogorov-Smirnov test is more powerful than Chi-square test when sample size is not too great. For large size sample both the tests have the same power. The most serious limitation of Kolmogorov-Smirnov test is that the distribution must be fully specified, that is, location, scale, and shape parameters can't be estimated from the data sample. Due to this limitation, many analysts prefer to use the **Anderson-Darling goodness-of-fit test**. However, the Anderson-Darling test is only available for a few specific distributions.

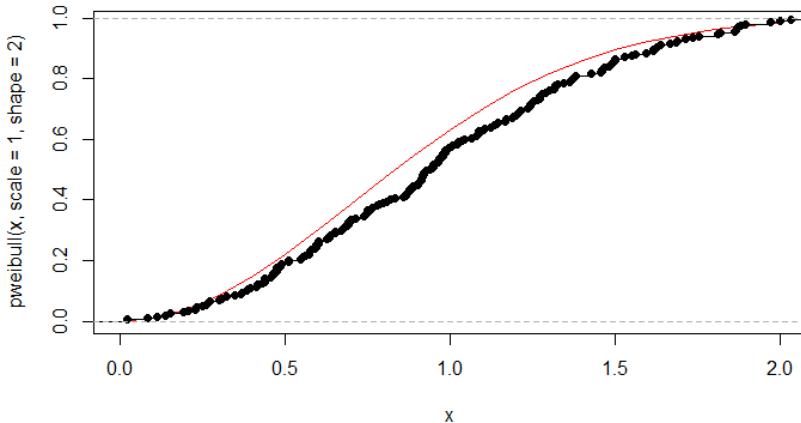
#### **EXAMPLE 7.** The K-S Test in R

In R we can perform Kolmogorov-Smirnov test using the function `ks.test()` and apply this test to a sample belonging from a Weibull pdf with known parameters (`shape=2` and `scale=1`):

```
ks.test(x.wei,"pweibull", shape=2,scale=1)
One-sample Kolmogorov-Smirnov test
data: x.wei
D = 0.0623, p-value = 0.4198
alternative hypothesis: two.sided
```

We accept null hypothesis that the data follow a Weibull distribution because the *p*-value is enough higher than significance levels usually referred in statistical literature. In Figure 12 is drawn both ECDF and theoretical one in the same plot:

```
x<-seq(0,2,0.1)
plot(x,pweibull(x,scale=1,shape=2),type="l",col="red",
main="ECDF and
weibull CDF")
plot(ecdf(x.wei),add=TRUE)
```



**Figure 12.** ECDF and a Weibull CDF

## 11.7. Normality tests

Very often a statistician is called to test if data collected come or not from a normal population, we shall examine the main normality tests. There are in statistical literature some tests useful for testing only skewness or only kurtosis (or both the two at the same time) of a distribution based on the well-known **b3 e b4** (or **gamma3 e gamma4**).

**Shapiro-Wilk test** is one of the most powerful normality tests, especially for small samples. Normality is tested by matching two alternative variance estimates: a non-parametric estimator got by a linear combination of ordered sample values and the usual parametric estimator. The weights ( $a_i$ ) are available in a statistical table:

$$W = \frac{\left(\sum_{i=1}^n a_i x_{(i)}\right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (11.28)$$

The statement performing Shapiro-Wilk test is `shapiro.test()` and it supplies  $W$  statistic and the  $p$ -value:

```
shapiro.test(x.norm)
Shapiro-wilk normality test
data: x.norm
W = 0.9938, p-value = 0.5659
```

The  $p$ -value is higher than significance levels usually used to test statistical hypotheses, we accept null hypothesis that is sample data belong form a gaussian distribution.

**Jarque-Bera test** is used a lot to test *normalità* in Econometric. It is based on skewness and kurtosis measures of a distribution considering the asymptotic distribution of **b3 e b4**, which, under null hypothesis, is a chi-square with 2 degrees of freedom.

In R such test is available in `tseries` package (it should be downloaded from CRAN website) using this statement: `jarque.bera.test()` which supplies the value of statistic, the degrees of freedom and the  $p$ -value:

```
library(tseries)      ##      package      tseries      Loading
jarque.bera.test(x.norm)
Jarque Bera Test
data: x.norm
X-squared = 0.539, df = 2, p-value = 0.7638
```

A test proposed by **Cucconi** (an Italian statistician) allows to test normality without the problem of estimating parameters by sample data. Let be  $x_1 \geq x_2 \geq \dots \geq x_n$  a sample from a continuous variable and

$\zeta_1, \zeta_2, \dots, \zeta_n$  a set of standardized random normal numbers of size  $n$ ; let be:

$$r = \zeta_n, \quad eq = \sqrt{\frac{\sum_{i=1}^n \zeta_i^2}{n - 1}}$$

(11.29) and (11.30)

we consider a transformation of  $x_i$ :  $y_i = \frac{x_i - \bar{x}}{\hat{\sigma}} + \frac{r}{\sqrt{n}}$  where  $\bar{x}$  is the sample mean and  $\hat{\sigma}$  is the square root of the corrected sample variance. It could be demonstrated that, if  $x_i$  come from a normal population,  $y_i$  have a normal standardized distribution. We can employ Kolmogorov-Smirnov test to check this hypothesis. Here is an example of R code:

```
zz<-rnorm(n=200,m=0,sd=1) ## sampling random numbers from N(0,1)
r<-zz[200]
q<-sd(zz[-200])
m<-mean(x.norm)
s<-sqrt(var(x.norm))
y<-q*((x.norm-m)/s)+(r/sqrt(200))
ks.test(y,"pnorm",m=0,sd=1)
One-sample Kolmogorov-Smirnov test data: y
D = 0.0298, p-value = 0.9943
alternative hypothesis: two.sided
```

A package called `nortest` (it should be downloaded from CRAN website) allows to perform 5 different normality test:

**1) sf.test()** performs Shapiro-Francia test:

```
library(nortest) ## package loading
sf.test(x.norm)
Shapiro-Francia normality test
data: x.norm
W = 0.9926, p-value = 0.3471
```

**2) ad.test()** performs Anderson-Darling21 test:

It is a modification of Kolmogorov-Smirnov test and it makes use of the specific distribution in calculating critical values. Currently, tables of critical values are available for the normal, lognormal, exponential,

Weibull, extreme value type I, and logistic distributions. Anderson-Darling test is based on this statistic:

$$A^2 = n - S \quad (11.29)$$

where

$$S = \sum_{i=1}^n \frac{(2i-1)}{n} [\ln F(x_i) + \ln(1 - F(x_{n+i+1}))] \quad (11.30)$$

Where  $n$  is the sample size and  $F(x)$  is the theoretical CDF (in our case is the normal CDF). In R environment this test can be used only to check normality:

```
library(nortest) ## package loading
ad.test(x.norm)
Anderson-Darling normality test
data: x.norm
A = 0.4007, p-value = 0.3581
```

**3) cvm.test()** performs Cramer-Von Mises test based on the statistic:

$$W^2 = \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 f(x) dx \quad (11.31)$$

```
library(nortest) ## package loading
cvm.test(x.norm)
Cramer-von Mises normality test
data: x.norm
w = 0.0545, p-value = 0.4449
```

**4) lillie.test()** performs Lilliefors test:

It is a modification of Kolmogorov-Smirnov test which can't be used for normality when the mean and standard deviation of the hypothesized normal distribution are not known (i.e., they are estimated from the sample data), it is particularly useful in case of small samples. The Lilliefors test evaluates the hypothesis that  $X$  has a normal distribution

with unspecified mean and variance, against the alternative that  $X$  does not have a normal distribution. This test compares the empirical distribution of  $X$  with a normal distribution having the same mean and variance as  $X$ . It is similar to the Kolmogorov-Smirnov test, but it adjusts for the fact that the parameters of the normal distribution are estimated from  $X$  rather than specified in advance.

```
library(nortest) ## package loading
lillie.test(x.norm)
Lilliefors (Kolmogorov-Smirnov) normality test
data: x.norm
D = 0.0414, p-value = 0.5509
```

5) `pearson.test()` performs Pearson's chi-square test: it is the same  $\chi^2$  test previously treated used to test the goodness-of-fit of a normal distribution:

```
library(nortest) ## package loading
pearson.test(x.norm)
Pearson chi-square normality test
data: x.norm
P = 10.12, p-value = 0.753
```

## 11.8. Appendix

List of R statements useful in fitting distributions. The package including statement is written in parenthesis.

`ad.test()`: Anderson-Darling test for normality (`nortest`)

`chisq.test()`: chi-squared test (`stats`)

`cut`: divides the range of data vector into intervals

`cvm.test()`: Cramer-von Mises test for normality (`nortest`)

`ecdf()`: computes an empirical cumulative distribution function (`stats`)

`fitdistr()`: Maximum-likelihood fitting of univariate distributions (`MASS`)

goodfit(): fits a discrete (count data) distribution for goodness-of-fit tests (vcd)

hist(): computes a histogram of the given data values (stats)

jarque.bera.test(): Jarque-Bera test for normality (tseries)

ks.test(): Kolmogorov-Simonov test (stats)

kurtosis(): returns value of kurtosis (fBasics)

lillie.test(): Lilliefors test for normality (nortest)

mle(): estimate parameters by the method of maximum likelihood (stats4)

pearson.test(): Pearson chi-square test for normality (nortest)

plot(): generic function for plotting of R objects (stats)

qqnorm(): produces a normal QQ plot (stats)

qqline(), qqplot(): produce a QQ plot of two datasets (stats)

sf.test(): test di Shapiro-Francia per la normalità (nortest)

shapiro.test(): Shapiro-Francia test for normalità (stats)

skewness(): returns value of skewness (fBasics)

table(): builds a contingency table (stats)

## 11.9. Review Problems

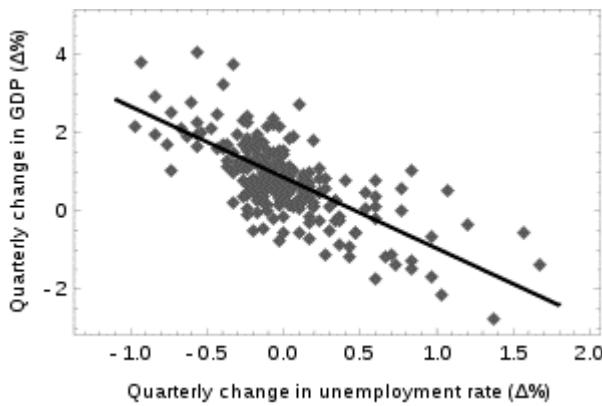
5. We can obtain samples from some pdf (such as gaussian, Poisson, Weibull, gamma, etc.) using R statements and after we draw a histogram of these data. Suppose we have a sample of size  $n = 500$  belonging from a normal population  $N(100,5)$  with mean = 100 and standard deviation = 5:
  - a. Define the distribution using the `rnorm()` statement
  - b. Create a histogram using the `hist()` statement
  - c. Estimate frequency density using `density()` and `plot()` to plot the graphic

- d. Compute the empirical cumulative distribution function by `ecdf()`
  - e. Test the goodness-of-fit of a gaussian distribution using `qqplot()` with a reference line
  - f. Perform the Shapiro-Wilk normality test
6. Repeat (1) (a)-(e) with a Weibull distribution with shape 3 and scale 1.5. Also perform the one-sample Kolmogorov-Smirnov test for goodness-of-fit.

## 12.What is Ordinary Least Squares?

In statistics, **ordinary least squares (OLS)** or linear least squares is a method for estimating the unknown parameters in a linear regression model. This method minimizes the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation. The resulting estimator can be expressed by a simple formula, especially in the case of a single regressor on the right-hand side.

The OLS estimator is consistent when the regressors are exogenous and there is no perfect multicollinearity, and optimal in the class of linear unbiased estimators when the errors are homoscedastic and serially uncorrelated. Under these conditions, the method of OLS provides minimum-variance mean-unbiased estimation when the errors have finite variances. Under the additional assumption that the errors be normally distributed, OLS is the maximum likelihood estimator. OLS is used in economics (econometrics), political science and electrical engineering (control theory and signal processing), among many areas of application.



**FIGURE 1.** Quarterly change in unemployment rate

Okun's law in macroeconomics states that in an economy the GDP growth should depend linearly on the changes in the unemployment rate. Here the ordinary least squares method is used to construct the regression line describing this law.

## 12.1. Linear model

Suppose the data consists of  $n$  observations  $\{y_i, x_i\}_{i=1}^n$ . Each observation includes a scalar response  $y_i$  and a vector of  $p$  predictors (or regressors)  $x_i$ . In a linear regression model the response variable is a linear function of the regressors:

$$y_i = x_i' \beta + \varepsilon_i, \quad (12.1)$$

where  $\beta$  is a  $p \times 1$  vector of unknown parameters;  $\varepsilon_i$ 's are unobserved scalar random variables (errors) which account for the discrepancy between the actually observed responses  $y_i$  and the “predicted outcomes”  $x_i' \beta$ ; and  $x'$  denotes matrix transpose, so that  $x' \beta$  is the dot product between the vectors  $x$  and  $\beta$ . This model can also be written in matrix notation as

$$\mathbf{y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}, \quad (12.2)$$

where  $\mathbf{y}$  and  $\boldsymbol{\varepsilon}$  are  $n \times 1$  vectors, and  $\mathbf{X}$  is an  $n \times p$  matrix of regressors, which is also sometimes called the *design matrix*.

As a rule, the constant term is always included in the set of regressors  $\mathbf{X}$ , say, by taking  $x_{i1} = 1$  for all  $i = 1, \dots, n$ .

The coefficient  $\beta_1$  corresponding to this regressor is called the *intercept*.

There may be some relationship between the regressors. For instance, the third regressor may be the square of the second regressor. In this case (assuming that the first regressor is constant) we have a quadratic model in the second regressor. But this is still considered a linear model because it is linear in the  $\beta$ s.

### 12.1.1. Assumptions

There are several different frameworks in which the linear regression model can be cast in order to make the OLS technique applicable. Each of these settings produces the same formulas and same results. The only difference is the interpretation and the assumptions which have to be imposed in order for the method to give meaningful results. The choice of the applicable framework depends mostly on the nature of data in hand, and on the inference task which has to be performed.

One of the lines of difference in interpretation is whether to treat the regressors as random variables, or as predefined constants. In the first case (random design) the regressors  $x_i$  are random and sampled together with the  $y_i$  's from some population, as in an observational study. This approach allows for more natural study of the asymptotic properties of the estimators. In the other interpretation (fixed design), the regressors  $X$  are treated as known constants set by a design, and  $y$  is sampled conditionally on the values of  $X$  as in an experiment. For practical purposes, this distinction is often unimportant, since estimation and inference is carried out while conditioning on  $X$ . All results stated in this article are within the random design framework.

The primary assumption of OLS is that there is zero or negligible errors in the independent variable, since this method only attempts to minimize the mean squared error in the dependent variable.

### 12.1.2. Classical linear regression model

The classical model focuses on the “finite sample” estimation and inference, meaning that the number of observations  $n$  is fixed. This contrasts with the other approaches, which study the asymptotic behavior of OLS, and in which the number of observations is allowed to grow to infinity.

- **Correct specification.** The linear functional form is correctly specified.
- **Strict exogeneity.** The errors in the regression should have conditional mean zero (Hayashi, 2000):

$$E[\varepsilon|X] = 0. \quad (12.3)$$

The immediate consequence of the exogeneity assumption is that the errors have mean zero:  $E[\varepsilon] = 0$ , and that the regressors are uncorrelated with the errors:  $E[X'\varepsilon] = 0$ .

The exogeneity assumption is critical for the OLS theory. If it holds then the regressor variables are called *exogenous* (Hayashi, 2000). If it doesn't, then those regressors that are correlated with the error term are called endogenous, and then the OLS estimates become invalid. In

such case the method of instrumental variables may be used to carry out inference.

- **No linear dependence.** The regressors in  $X$  must all be linearly independent. Mathematically it means that the matrix  $X$  must have full column rank almost surely (Hayashi, 2000):

$$Pr[\text{rank}(X) = p] = 1 \quad (12.4)$$

Usually, it is also assumed that the regressors have finite moments up to at least second. In such case the matrix  $Q_{xx} = E[X'X/n]$  will be finite and positive semi-definite. When this assumption is violated the regressors are called linearly dependent or perfectly multicollinear. In such case the value of the regression coefficient  $\beta$  cannot be learned, although prediction of  $y$  values is still possible for new values of the regressors that lie in the same linearly dependent subspace.

- Spherical errors:

$$Var[\varepsilon|X] = \sigma^2 I_n \quad (12.5)$$

where  $I_n$  is an  $n \times n$  identity matrix, and  $\sigma^2$  is a parameter which determines the variance of each observation. This  $\sigma^2$  is considered a nuisance parameter in the model, although usually it is also estimated. If this assumption is violated then the OLS estimates are still valid, but no longer efficient. It is customary to split this assumption into two parts:

- **Homoscedasticity:**  $E[\varepsilon_i^2|X] = \sigma^2$ , which means that the error term has the same variance  $\sigma^2$  in each  $i$  observation. When this requirement is violated this is called heteroscedasticity, in such case a more efficient estimator would be weighted least squares. If the errors have infinite variance then the OLS estimates will also have infinite variance (although by the law of large numbers they will nonetheless tend toward the true values so long as the errors have zero mean). In this case, robust estimation techniques are recommended.
- **Nonautocorrelation:** the errors are uncorrelated between observations:  $E[\varepsilon_i \varepsilon_j|X] = 0$  for  $i \neq j$ . This assumption may be violated in the context of time series data, panel data, cluster

samples, hierarchical data, repeated measures data, longitudinal data, and other data with dependencies. In such cases generalized least squares provides a better alternative than the OLS.

- **Normality.** It is sometimes additionally assumed that the errors have normal distribution conditional on the regressors (Hayashi, 2000):

$$\varepsilon | X \sim \mathcal{N}(0, \sigma^2 I_n). \quad (12.6)$$

This assumption is not needed for the validity of the OLS method, although certain additional finite-sample properties can be established in case when it does (especially in the area of hypotheses testing). Also when the errors are normal, the OLS estimator is equivalent to the maximum likelihood estimator (MLE), and therefore it is asymptotically efficient in the class of all regular estimators.

### 12.1.3. Independent and identically distributed

In some applications, especially with cross-sectional data, an additional assumption is imposed — that all observations are **independent and identically distributed (iid)**. This means that all observations are taken from a random sample which makes all the assumptions listed earlier simpler and easier to interpret. Also this framework allows one to state asymptotic results (as the sample size  $n \rightarrow \infty$ ), which are understood as a theoretical possibility of fetching new independent observations from the data generating process. The list of assumptions in this case is:

- iid observations:  $(x_i, y_i)$  is independent from, and has the same distribution as,  $(x_j, y_j)$  for all  $i \neq j$ ;
- no perfect multicollinearity:  $Q_{xx} = E[x_i x'_j]$  is a positive-definite matrix;
- exogeneity:  $E[\varepsilon_i | x_i] = 0$
- homoscedasticity:  $Var[\varepsilon_i | x_i] = \sigma^2$

### 12.1.4. Time series model

- The stochastic process  $\{x_i, y_i\}$  is stationary and ergodic;
- The regressors are predetermined:  $E[x_i \varepsilon_i] = 0$  for all  $i = 1, \dots, n$ ;

- The  $p \times p$  matrix  $Q_{xx} = E[x_i x_j']$  is of full rank, and hence positive-definite;
- $\{x_i \varepsilon_i\}$  is a martingale difference sequence, with a finite matrix of second moments  $Q_{xx\varepsilon^2} = E[\varepsilon_i^2 x_i x_j']$

## 12.2. Estimation

Suppose  $b$  is a “candidate” value for the parameter  $\beta$ . The quantity  $y_i - x_i'b$  is called the residual for the  $i$ -th observation, it measures the vertical distance between the data point  $(x_i, y_i)$  and the hyperplane  $y = x'b$ , and thus assesses the degree of fit between the actual data and the model. The sum of squared residuals (SSR) (also called the error sum of squares (ESS) or residual sum of squares (RSS)) (Hayashi, 2000) is a measure of the overall model fit:

$$S(b) = \sum_{i=1}^n (y_i - x_i'b)^2 = (y - Xb)^T(y - Xb), \quad (12.7)$$

where  $T$  denotes the matrix transpose. The value of  $b$  which minimizes this sum is called the OLS estimator for  $\beta$ . The function  $S(b)$  is quadratic in  $b$  with positive-definite Hessian, and therefore this function possesses a unique global minimum at  $b = \hat{\beta}$ , which can be given by the explicit formula (Hayashi, 2000):

$$\hat{\beta} = \arg \min_{b \in \mathbb{R}^p} S(b) = \left( \frac{1}{n} \sum_{i=1}^n x_i x_i' \right)^{-1} \cdot \frac{1}{n} \sum_{i=1}^n x_i y_i$$

or equivalently in matrix form,

$$\hat{\beta} = (X^T X)^{-1} X^T y. \quad (12.8)$$

After we have estimated  $\beta$ , the fitted values (or predicted values) from the regression will be

$$\hat{y} = X\hat{\beta} = Py, \quad (12.9)$$

where  $P = X(X^T X)^{-1}X^T$  is the projection matrix onto the space spanned by the columns of  $X$ . This matrix  $P$  is also sometimes called the hat matrix because it “puts a hat” onto the variable  $y$ . Another matrix, closely related to  $P$  is the annihilator matrix  $M = I_n - P$ , this is a projection matrix onto the space orthogonal to  $X$ . Both matrices  $P$  and  $M$  are symmetric and idempotent (meaning that  $P^2 = P$ ), and relate to the data matrix  $X$  via identities  $PX = X$  and  $MX = 0$  (Hayashi, 2000).

Matrix  $M$  creates the residuals from the regression:

$$\hat{\varepsilon} = y - X\hat{\beta} = My = M\varepsilon \quad (12.10)$$

Using these residuals we can estimate the value of  $\sigma^2$ :

$$s^2 = \frac{\hat{\varepsilon}'\hat{\varepsilon}}{n-p} = \frac{y'My}{n-p} = \frac{S(\hat{\beta})}{n-p}, \quad \hat{\sigma}^2 = \frac{n-p}{n}s^2$$

The numerator,  $n - p$ , is the statistical degrees of freedom. The first quantity,  $s^2$ , is the OLS estimate for  $\sigma^2$ , whereas the second,  $\hat{\sigma}^2$ , is the MLE estimate for  $\sigma^2$ . The two estimators are quite similar in large samples; the first one is always unbiased, while the second is biased but minimizes the mean squared error of the estimator. In practice  $s^2$  is used more often, since it is more convenient for the hypothesis testing. The square root of  $s^2$  is called the standard error of the regression (SER), or standard error of the equation (SEE).

It is common to assess the goodness-of-fit of the OLS regression by comparing how much the initial variation in the sample can be reduced by regressing onto  $X$ . The coefficient of determination  $R^2$  is defined as a ratio of “explained” variance to the “total” variance of the dependent variable  $y$  (Hayashi, 2000):

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} = \frac{y'P'LPy}{y'Ly} = 1 - \frac{y'My}{y'Ly} = 1 - \frac{SSR}{TSS} \quad (12.11)$$

where TSS is the total sum of squares for the dependent variable,  $L = I_n - \mathbf{1}\mathbf{1}'/n$ , and  $\mathbf{1}$  is an  $n \times 1$  vector of ones. ( $L$  is a “centering matrix” which is equivalent to regression on a constant; it simply subtracts the mean from a variable.)

In order for  $R^2$  to be meaningful, the matrix  $X$  of data on regressors must contain a column vector of ones to represent the constant whose coefficient is the regression intercept. In that case,  $R^2$  will always be a number between 0 and 1, with values close to 1 indicating a good degree of fit.

### 12.2.1. Simple regression model

If the data matrix  $X$  contains only two variables: a constant, and a scalar regressor  $x_i$ , then this is called the “simple regression model” (Hayashi, 2000). This case is often considered in the beginner statistics classes, as it provides much simpler formulas even suitable for manual calculation. The vectors of parameters in such model is 2-dimensional, and is commonly denoted as  $(\alpha, \beta)$ :

$$y_i = \alpha + \beta x_i + \varepsilon_i. \quad (12.2)$$

The least squares estimates in this case are given by simple formulas

$$\hat{\beta} = \frac{\sum x_i y_i - \frac{1}{n} \sum x_i \sum y_i}{\sum x_i^2 - \frac{1}{n} (\sum x_i)^2} = \frac{\text{Cov}[x, y]}{\text{Var}[x]}, \hat{\alpha} = \bar{y} - \hat{\beta} \bar{x}$$

(12.13)

## 12.3. Alternative derivations

In the previous section the least squares estimator  $\hat{\beta}$  was obtained as a value that minimizes the sum of squared residuals of the model. However it is also possible to derive the same estimator from other approaches. In all cases the formula for OLS estimator remains the same:  $\hat{\beta} = (X'X)^{-1}X'y$ , the only difference is in how we interpret this result.

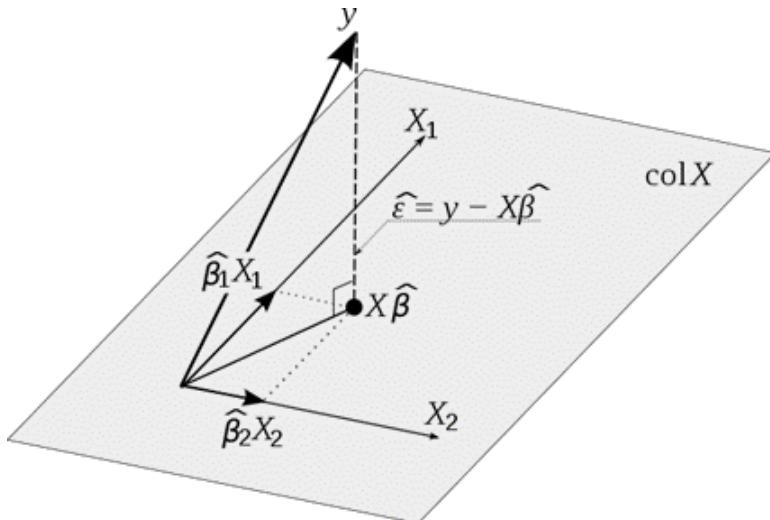
### 12.3.1. Geometric approach

For mathematicians, OLS is an approximate solution to an overdetermined system of linear equations  $X\beta \approx y$ , where  $\beta$  is the unknown. Assuming the system cannot be solved exactly (the number of equations  $n$  is much larger than the number of unknowns  $p$ ), we are looking for a solution that could provide the smallest discrepancy

between the right- and left- hand sides. In other words, we are looking for the solution that satisfies

$$\hat{\beta} = \arg \min_{\beta} \|y - X\beta\|$$

where  $\|\cdot\|$  is the standard  $L^2$  norm in the  $n$ -dimensional Euclidean space  $R^n$ . The predicted quantity  $X\beta$  is just a certain linear combination of the vectors of regressors. Thus, the residual vector  $y - X\beta$  will have the smallest length when  $y$  is projected orthogonally onto the linear subspace spanned by the columns of  $X$ . The OLS estimator in this case can be interpreted as the coefficients of vector decomposition of  $\hat{y} = Py$  along the basis of  $X$ .



**FIGURE 2.** OLS estimation can be viewed as a projection onto the linear space spanned by the regressors.

Another way of looking at it is to consider the regression line to be a weighted average of the lines passing through the combination of any two points in the dataset (Akbarzadeh, 2013). Although this way of calculation is more computationally expensive, it provides a better intuition on OLS.

### 12.3.2. Maximum likelihood

The OLS estimator is identical to the **maximum likelihood estimator** (MLE) under the normality assumption for the error terms (Hayashi,

2000). This normality assumption has historical importance, as it provided the basis for the early work in linear regression analysis by Yule and Pearson. From the properties of MLE, we can infer that the OLS estimator is asymptotically efficient (in the sense of attaining the Cramér–Rao bound for variance) if the normality assumption is satisfied (Hayashi, 2000).

### 12.3.3. Generalized method of moments

In iid case the OLS estimator can also be viewed as a GMM estimator arising from the moment conditions

$$E[x_i(y_i - x_i' \beta)] = 0.$$

These moment conditions state that the regressors should be uncorrelated with the errors. Since  $x_i$  is a  $p$ -vector, the number of moment conditions is equal to the dimension of the parameter vector  $\beta$ , and thus the system is exactly identified. This is the so-called classical GMM case, when the estimator does not depend on the choice of the weighting matrix.

Note that the original strict exogeneity assumption  $E[\varepsilon_i|x] = 0$  implies a far richer set of moment conditions than stated above. In particular, this assumption implies that for any vector-function  $f$ , the moment condition  $E[f(x_i) \cdot \varepsilon_i] = 0$  will hold. However it can be shown using the Gauss–Markov theorem that the optimal choice of function  $f$  is to take  $f(x) = x$ , which results in the moment equation posted above.

## 12.4. Finite sample properties

First of all, under the strict exogeneity assumption the OLS estimators  $\hat{\beta}$  and  $s^2$  are unbiased, meaning that their expected values coincide with the true values of the parameters (Hayashi, 2000):

$$E[\hat{\beta}|X] = \beta, \quad E[s^2|X] = \sigma^2. \quad (12.14 \& 12.15)$$

If the strict exogeneity does not hold (as is the case with many time series models, where exogeneity is assumed only with respect to the past shocks but not the future ones), then these estimators will be biased in finite samples.

The variance-covariance matrix of  $\hat{\beta}$  is equal to (Hayashi, 2000)

$$\text{Var}[\hat{\beta}|X] = \sigma^2(X'X)^{-1}. \quad (12.16)$$

In particular, the standard error of each coefficient  $\hat{\beta}_j$  is equal to square root of the  $j$ -th diagonal element of this matrix. The estimate of this standard error is obtained by replacing the unknown quantity  $\sigma^2$  with its estimate  $s^2$ .

Thus,

$$\widehat{se}(\hat{\beta}_j) = \sqrt{s^2(X'X)_{jj}^{-1}}$$

It can also be easily shown that the estimator is  $\hat{\beta}$  uncorrelated with the residuals from the model (Hayashi, 2000):

$$\text{Cov}[\hat{\beta}, \hat{\varepsilon}|X] = 0. \quad (12.17)$$

The Gauss–Markov theorem states that under the spherical errors assumption (that is, the errors should be uncorrelated and homoscedastic) the estimator is efficient in the class of linear unbiased estimators. This is called the **best linear unbiased estimator** (BLUE). Efficiency should be understood as if we were to find some other estimator  $\tilde{\beta}$  which would be linear in  $y$  and unbiased, then (Hayashi, 2000)

$$\text{Var}[\tilde{\beta}|X] - \text{Var}[\hat{\beta}|X] \quad (12.18)$$

in the sense that this is a nonnegative-definite matrix. This theorem establishes optimality only in the class of linear unbiased estimators, which is quite restrictive. Depending on the distribution of the error terms  $\varepsilon$ , other, non-linear estimators may provide better results than OLS.

### 12.4.1. Assuming normality

The properties listed so far are all valid regardless of the underlying distribution of the error terms. However if you are willing to assume that the normality assumption holds (that is, that  $\varepsilon \sim N(0, \sigma^2 I_n)$ ), then additional properties of the OLS estimators can be stated.

The estimator  $\hat{\beta}$  is normally distributed, with mean and variance as given before (Amemiya, 1985):

$$\hat{\beta} \sim \mathcal{N}(\beta, \sigma^2(X'X)^{-1}) \quad (12.19)$$

This estimator reaches the Cramér–Rao bound<sup>2</sup> for the model, and thus is optimal in the class of all unbiased estimators. Note that unlike the Gauss–Markov theorem<sup>3</sup>, this result establishes optimality among both linear and non-linear estimators, but only in the case of normally distributed error terms.

The estimator  $s^2$  will be proportional to the chi-squared distribution (Amemiya, 1985):

$$s^2 \sim \frac{\sigma^2}{n-p} \cdot \chi_{n-p}^2 \quad (12.20)$$

The variance of this estimator is equal to  $2\sigma^4/(n - p)$ , which does not attain the Cramér–Rao bound of  $2\sigma^4/n$ . However it was shown that there are no unbiased estimators of  $\sigma^2$  with variance smaller than that of the estimator  $s^2$  (Rao, 1973). If we are willing to allow biased estimators, and consider the class of estimators that are proportional to the sum of squared residuals (SSR) of the model, then the best (in the sense of the mean squared error) estimator in this class will be  $\tilde{\sigma}^2 = SSR/(n - p + 2)$ , which even beats the Cramér–Rao bound in case when there is only one regressor ( $p = 1$ ) (Amemiya, 1985).

Moreover, the  $\hat{\beta}$  estimators and  $s^2$  are independent (Amemiya, 1985), the fact which comes in useful when constructing the  $t$ - and  $F$ -tests for the regression.

<sup>2</sup> Cramér–Rao bound stands for an inequality that is the basis of a method for determining a lower bound to the variance of an estimate of a deterministic parameter.

<sup>3</sup> The Gauss–Markov theorem asserts that the ordinary least-squares estimator

$\hat{\beta} = (X'X)^{-1}X'y$  of the parameter  $\beta$  in the classical linear regression model  $y; X\beta\sigma^2I$  is the unbiased linear estimator of least dispersion.

### 12.4.2. Influential observations

As was mentioned before, the estimator  $\hat{\beta}$  is linear in  $y$ , meaning that it represents a linear combination of the dependent variables  $y_i$ 's. The weights in this linear combination are functions of the regressors  $X$ , and generally are unequal. The observations with high weights are called influential because they have a more pronounced effect on the value of the estimator.

To analyze which observations are influential we remove a specific  $j$ -th observation and consider how much the estimated quantities are going to change (similarly to the jackknife method). It can be shown that the change in the OLS estimator for  $\beta$  will be equal to (Davidson & Mackinnon, 1993)

$$\hat{\beta}^{(j)} - \hat{\beta} = -\frac{1}{1-h_j}(X'X)^{-1}x_j'\varepsilon_j \quad (12.21)$$

where  $h = x_j(X'X)^{-1}x_j$  is the  $j$ -th diagonal element of the hat matrix  $P$ , and  $x_j$  is the vector of regressors corresponding to the  $j$ -th observation. Similarly, the change in the predicted value for  $j$ -th observation resulting from omitting that observation from the dataset will be equal to (Davidson & Mackinnon, 1993)

$$\hat{y}_j^{(j)} - \hat{y}_j = x_j'\hat{\beta}^{(j)} - x_j'\hat{\beta} = -\frac{h_j}{1-h_j}\hat{\varepsilon}_j \quad (12.22)$$

From the properties of the hat matrix,  $0 \leq h_j \leq 1$ , and they sum up to  $p$ , so that on average  $h_j \approx p/n$ . These quantities  $h_j$  are called the leverages, **and observations with high  $h_j$ 's — leverage points** (Davidson & Mackinnon, 1993). Usually the observations with high leverage ought to be scrutinized more carefully, in case they are erroneous, or outliers, or in some other way atypical of the rest of the dataset.

### 12.4.3. Partitioned regression

Sometimes the variables and corresponding parameters in the regression can be logically split into two groups, so that the regression takes form

$$y = X_1\beta_1 + X_2\beta_2 + \varepsilon, \quad (12.23)$$

where  $X_1$  and  $X_2$  have dimensions  $n \times p$ ,  $n \times p$ , and  $\beta_1, \beta_2$  are  $p \times 1$  and  $p \times 1$  vectors, with  $p_1 + p_2 = p$ .

The Frisch–Waugh–Lovell theorem states that in this regression the residuals  $\hat{\varepsilon}$  and the OLS estimate  $\hat{\beta}_2$  will be numerically identical to the residuals and the OLS estimate for  $\beta_2$  in the following regression (Davidson & Mackinnon, 1993):

$$M_1 y = M_1 X_2 \beta_2 + \eta, \quad (12.24)$$

where  $M_1$  is the annihilator matrix for regressors  $X_1$ .

The theorem can be used to establish a number of theoretical results. For example, having a regression with a constant and another regressor is equivalent to subtracting the means from the dependent variable and the regressor and then running the regression for the demeaned variables but without the constant term.

#### 12.4.4. Constrained estimation

Suppose it is known that the coefficients in the regression satisfy a system of linear equations

$$H_0: Q'\beta = c$$

where  $Q$  is a  $p \times q$  matrix of full rank, and  $c$  is a  $q \times 1$  vector of known constants, where  $q < p$ . In this case least squares estimation is equivalent to minimizing the sum of squared residuals of the model subject to the constraint  $H_0$ . The **constrained least squares (CLS)** estimator can be given by an explicit formula (Amemiya, 1985):

$$\hat{\beta}^c = \hat{\beta} - (X'X)^{-1}Q(Q'(X'X)^{-1}Q)^{-1}(Q'\hat{\beta} - c) \quad (12.25)$$

This expression for the constrained estimator is valid as long as the matrix  $X'X$  is invertible. It was assumed from the beginning of this article that this matrix is of full rank, and it was noted that when the rank

condition fails,  $\beta$  will not be identifiable. However it may happen that adding the restriction  $H_0$  makes  $\beta$  identifiable, in which case one would like to find the formula for the estimator. The estimator is equal to (Amemiya, 1985)

$$\hat{\beta}^c = R(R'X'XR)^{-1}R'X'y + (I_p - R(R'X'XR)^{-1}R'X'X)Q(Q'Q)^{-1}c, \quad (12.26)$$

where  $R$  is a  $p \times (p - q)$  matrix such that the matrix  $[QR]$  is non-singular, and  $R'Q = 0$ . Such a matrix can always be found, although generally it is not unique. The second formula coincides with the first in case when  $X'X$  is invertible (Amemiya, 1985).

## 12.5. Large sample properties

The least squares estimators are point estimates of the linear regression model parameters  $\beta$ . However, generally we also want to know how close those estimates might be to the true values of parameters. In other words, we want to construct the interval estimates.

Since we haven't made any assumption about the distribution of error term  $\varepsilon_i$ , it is impossible to infer the distribution of the estimators  $\hat{\beta}$  and  $\hat{\sigma}^2$ . Nevertheless, we can apply the law of large numbers and central limit theorem to derive their asymptotic properties as sample size  $n$  goes to infinity. While the sample size is necessarily finite, it is customary to assume that  $n$  is "large enough" so that the true distribution of the OLS estimator is close to its asymptotic limit, and the former may be approximately replaced by the latter.

We can show that under the model assumptions, the least squares estimator for  $\beta$  is consistent (that is  $\hat{\beta}$  converges in probability to  $\beta$ ) and asymptotically normal:

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} \mathcal{N}(0, \sigma^2 Q_{xx}^{-1}), \quad (12.27)$$

where  $Q_{xx} = X'X$ .

Using this asymptotic distribution, approximate two-sided confidence intervals for the  $j$ -th component of the vector  $\hat{\beta}$  can be constructed as at the  $1 - \alpha$  confidence level,

$$\beta_j \in \left[ \hat{\beta}_j \pm q_{1-\alpha/2}^{\mathcal{N}(0,1)} \sqrt{\frac{1}{n} \hat{\sigma}^2 [Q_{xx}^{-1}]_{jj}} \right]$$

where  $q$  denotes the quantile function of standard normal distribution, and  $[\cdot]_{jj}$  is the  $j$ -th diagonal element of a matrix.

Similarly, the least squares estimator for  $\sigma^2$  is also consistent and asymptotically normal (provided that the fourth moment of  $\varepsilon_t$  exists) with limiting distribution

$$\sqrt{n}(\hat{\sigma}^2 - \sigma^2) \xrightarrow{d} \mathcal{N}(0, E[\varepsilon_t^4] - \sigma^4). \quad (12.28)$$

These asymptotic distributions can be used for prediction, testing hypotheses, constructing other estimators, etc. As an example consider the problem of prediction. Suppose  $x_0$  is some point within the domain of distribution of the regressors, and one wants to know what the response variable would have been at that point. The mean response is the quantity  $y_0 = x_0' \beta$ , whereas the predicted response is  $\hat{y}_0 = x_0' \hat{\beta}$ . Clearly the predicted response is a random variable, its distribution can be derived from that of:

$$\sqrt{n}(\hat{y}_0 - y_0) \xrightarrow{d} \mathcal{N}(0, \sigma^2 x_0' Q_{xx}^{-1} x_0), \quad (12.29)$$

which allows construct confidence intervals for mean response  $y_0$  to be constructed:

$$y_0 \in \left[ x_0' \hat{\beta} \pm q_{1-\alpha/2}^{\mathcal{N}(0,1)} \sqrt{\frac{1}{n} \hat{\sigma}^2 x_0' Q_{xx}^{-1} x_0} \right] \quad (12.30)$$

at the  $1 - \alpha$  confidence level.

### EXAMPLE 1. Height and Weight

This example exhibits the common mistake of ignoring the condition of having zero error in the dependent variable.

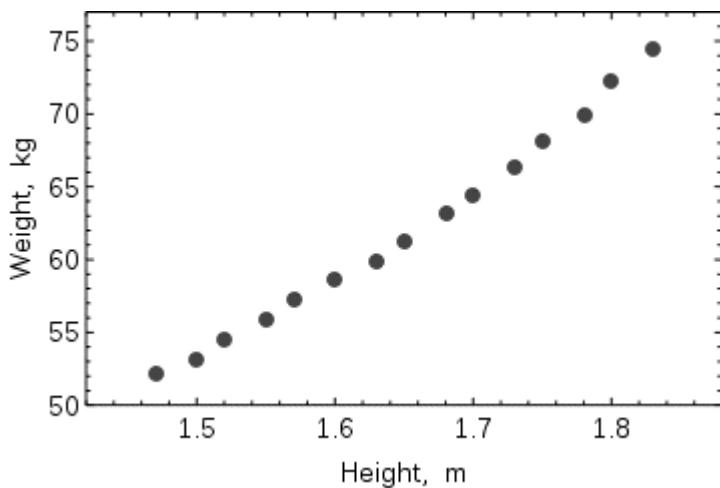
The following data set (Tables 1a and 1b) gives average heights and weights for American women aged 30–39 (source: The World Almanac and Book of Facts, 1975).

**Table 1a.** Height and Weight data, Part I

| n            | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Height (m):  | 1.47  | 1.50  | 1.52  | 1.55  | 1.57  | 1.60  | 1.63  | 1.65  |
| Weight (kg): | 52.21 | 53.12 | 54.48 | 55.84 | 57.20 | 58.57 | 59.93 | 61.29 |

**Table 1b.** Height and Weight data, Part II

| n            | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|--------------|-------|-------|-------|-------|-------|-------|-------|
| Height (m):  | 1.68  | 1.70  | 1.73  | 1.75  | 1.78  | 1.80  | 1.83  |
| Weight (kg): | 63.11 | 64.47 | 66.28 | 68.10 | 69.92 | 72.19 | 74.46 |



**Figure 3.** Scatterplot of Height versus weight

Scatterplot of the data; the relationship is slightly curved but close to linear.

When only one dependent variable is being modeled, a scatterplot will suggest the form and strength of the relationship between the dependent variable and regressors. It might also reveal outliers, heteroscedasticity, and other aspects of the data that may complicate the interpretation of a fitted regression model. The scatterplot suggests that the relationship is strong and can be approximated as a quadratic function. OLS can handle non-linear relationships by introducing the regressor  $HEIGHT^2$ . The regression model then becomes a multiple linear model:

$$w_i = \beta_1 + \beta_2 h_i + \beta_e h_i^2 + \varepsilon_i.$$

The output from most popular statistical packages will look similar to this:

**Table 2.** Method: Least Squares Dependent variable: WEIGHT Included observations: 15

| Variable             | Coefficient | Std. Error         | t-statistic | p-value |
|----------------------|-------------|--------------------|-------------|---------|
| $\beta$              | 128.8128    | 16.3083            | 7.8986      | 0.0000  |
| $h$                  | -143.1620   | 19.8332            | -7.2183     | 0.0000  |
| $h^2$                | 61.9603     | 6.0084             | 10.3122     | 0.0000  |
| $R^2$                | 0.9989      | S.E. of regression |             | 0.2516  |
| Adjusted $R^2$       | 0.9987      | Model sum-of-sq    |             | 692.61  |
| Log-likelihood       | 1.0890      | Residual sum-of-sq |             | 0.7595  |
| Durbin–Watson stats. | 2.1013      | Total sum-of-sq    |             | 693.37  |
| Akaike criterion     | 0.2548      | F-statistic        |             | 5471.2  |
| Schwarz criterion    | 0.3964      | p-value (F-stat)   |             | 0.0000  |

In this table:

- The Coefficient column gives the least squares estimates of parameters  $\beta_j$
- The Std. errors column shows standard errors of each coefficient

estimate:  $\left(\frac{1}{n}\hat{\sigma}^2[Q_{xx}^{-1}]_{jj}\right)^{1/2}$

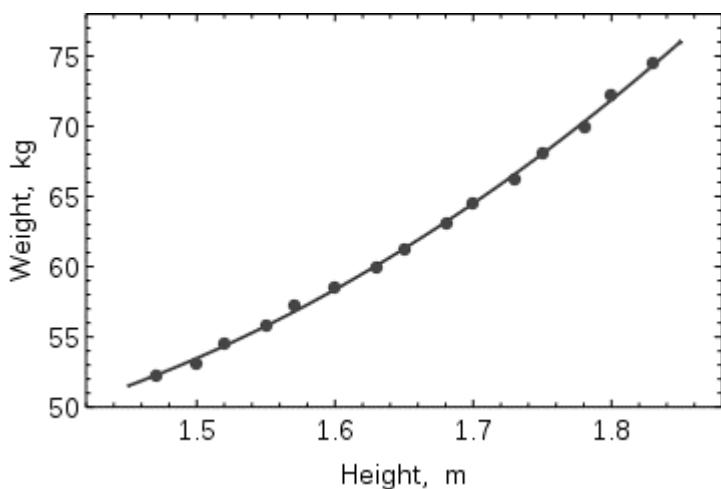
- The  $t$ -statistic and  $p$ -value columns are testing whether any of the coefficients might be equal to zero. The  $t$ -statistic is calculated simply as  $t = \hat{\beta}_j/\hat{\sigma}_j$ . If the errors  $\epsilon$  follow a normal distribution,  $t$  follows a Student-t distribution. Under weaker conditions,  $t$  is asymptotically normal. Large values of  $t$  indicate that the null hypothesis can be rejected and that the corresponding coefficient is not zero. The second column, p-value, expresses the results of the hypothesis test as a significance level. Conventionally,  $p$ -values smaller than 0.05 are taken as evidence that the population coefficient is nonzero.
- $R$ -squared  $R^2$  is the coefficient of determination indicating goodness-of-fit of the regression. This statistic will be equal to one if fit is perfect, and to zero when regressors  $X$  have no explanatory power whatsoever. This is a biased estimate of the population R-squared, and will never decrease if additional regressors are added, even if they are irrelevant.
- Adjusted  $R$ -squared  $\bar{R}^2$  is a slightly modified version of  $R^2$ , designed to penalize for the excess number of regressors which do not add to the explanatory power of the regression. This statistic is always smaller than  $R^2$ , can decrease as new regressors are added, and even be negative for poorly fitting models:

$$\bar{R}^2 = 1 - \frac{n-1}{n-p}(1-R^2)$$

- Log-likelihood is calculated under the assumption that errors follow normal distribution. Even though the assumption is not very reasonable, this statistic may still find its use in conducting LR tests.
- Durbin–Watson statistic tests whether there is any evidence of serial correlation between the residuals. As a rule of thumb, the value smaller than 2 will be an evidence of positive correlation.
- Akaike information criterion and Schwarz criterion are both used for model selection. Generally when comparing two alternative models, smaller values of one of these criteria will indicate a better model

(Burnham & Anderson, 2002).

- Standard error of regression is an estimate of  $\sigma$ , standard error of the error term.
- Total sum of squares, model sum of squared, and residual sum of squares tell us how much of the initial variation in the sample were explained by the regression.
- $F$ -statistic tries to test the hypothesis that all coefficients (except the intercept) are equal to zero. This statistic has  $F(p-1, n-p)$  distribution under the null hypothesis and normality assumption, and its  $p$ -value indicates probability that the hypothesis is indeed true. Note that when errors are not normal this statistic becomes invalid, and other tests such as for example Wald test or LR test should be used.



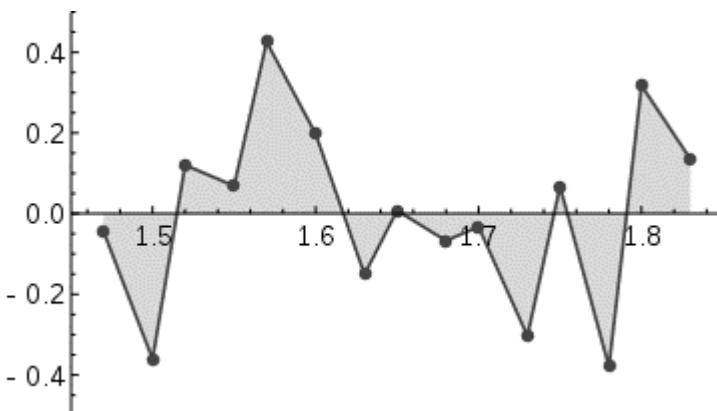
**Figure 4.** Fitted Regression

Ordinary least squares analysis often includes the use of diagnostic plots designed to detect departures of the data from the assumed form of the model. These are some of the common diagnostic plots:

- Residuals against the explanatory variables in the model. A non-linear relation between these variables suggests that the linearity of the conditional mean function may not hold. Different levels of variability in the residuals for different levels of the explanatory

variables suggests possible heteroscedasticity.

- Residuals against explanatory variables not in the model. Any relation of the residuals to these variables would suggest considering these variables for inclusion in the model.
- Residuals against the fitted values,  $\hat{y}$ .
- Residuals against the preceding residual. This plot may identify serial correlations in the residuals.



**Figure 5.** Residuals plot

An important consideration when carrying out statistical inference using regression models is how the data were sampled. In this example, the data are averages rather than measurements on individual women. The fit of the model is very good, but this does not imply that the weight of an individual woman can be predicted with high accuracy based only on her height.

### 12.5.1. Sensitivity to rounding

This example also demonstrates that coefficients determined by these calculations are sensitive to how the data is prepared. The heights were originally given rounded to the nearest inch and have been converted and rounded to the nearest centimeter. Since the conversion factor is one inch to 2.54 cm this is not an exact conversion. The original inches can be recovered by  $\text{Round}(x/0.0254)$  and then re-converted to metric without rounding. If this is done the results become:

**Table 3.** Rounding Errors

| Constant | Height    | Height <sup>2</sup> |                                       |
|----------|-----------|---------------------|---------------------------------------|
| 128.8128 | -143.1620 | 61.96033            | Converted to metric with rounding.    |
| 119.0205 | -131.5076 | 58.50460            | Converted to metric without rounding. |

Using either of these equations to predict the weight of a 5' 6" (1.6764m) woman gives similar values: 62.94 kg with rounding vs. 62.98 kg without rounding. Thus a seemingly small variation in the data has a real effect on the coefficients but a small effect on the results of the equation.

While this may look innocuous in the middle of the data range it could become significant at the extremes or in the case where the fitted model is used to project outside the data range (extrapolation).

This highlights a common error: this example is an abuse of OLS which inherently requires that the errors in the dependent variable (in this case height) are zero or at least negligible. The initial rounding to nearest inch plus any actual measurement errors constitute a finite and non-negligible error. As a result the fitted parameters are not the best estimates they are presumed to be. Though not totally spurious the error in the estimation will depend upon relative size of the  $x$  and  $y$  errors.

## 12.6. Example Using R

### EXAMPLE 2. Tortoises in The Galapagos

Now let's look at an example concerning the number of species of tortoise on the various Galapagos Islands. There are 30 cases (Islands) and 7 variables in the dataset. We start by reading the data into R and examining it. Download the following data and save it as a text file: gala.txt (tab delimited) in a directory, like "C/mydata/".

|           | Species | Endemics | Area  | Elevation | Nearest | Scruz | Adjacent |
|-----------|---------|----------|-------|-----------|---------|-------|----------|
| Baltra    | 58      | 23       | 25.09 | 346       | 0.6     | 0.6   | 1.84     |
| Bartolome | 31      | 21       | 1.24  | 109       | 0.6     | 26.3  | 572.33   |

| Caldwell     | 3   | 3  | 0.21    | 114  | 2.8  | 58.7  | 0.78    |  |
|--------------|-----|----|---------|------|------|-------|---------|--|
| Champion     | 25  | 9  | 0.10    | 46   | 1.9  | 47.4  | 0.18    |  |
| Coamano      | 2   | 1  | 0.05    | 77   | 1.9  | 1.9   | 903.82  |  |
| Daphne.Major | 18  | 11 | 0.34    | 119  | 8.0  | 8.0   | 1.84    |  |
| Daphne.Minor | 24  | 0  | 0.08    | 93   | 6.0  | 12.0  | 0.34    |  |
| Darwin       | 10  | 7  | 2.33    | 168  | 34.1 | 290.2 | 2.85    |  |
| Eden         | 8   | 4  | 0.03    | 71   | 0.4  | 0.4   | 17.95   |  |
| Enderby      | 2   | 2  | 0.18    | 112  | 2.6  | 50.2  | 0.10    |  |
| Espanola     | 97  | 26 | 58.27   | 198  | 1.1  | 88.3  | 0.57    |  |
| Fernandina   | 93  | 35 | 634.49  | 1494 | 4.3  | 95.3  | 4669.32 |  |
| Gardner1     | 58  | 17 | 0.57    | 49   | 1.1  | 93.1  | 58.27   |  |
| Gardner2     | 5   | 4  | 0.78    | 227  | 4.6  | 62.2  | 0.21    |  |
| Genovesa     | 40  | 19 | 17.35   | 76   | 47.4 | 92.2  | 129.49  |  |
| Isabela      | 347 | 89 | 4669.32 | 1707 | 0.7  | 28.1  | 634.49  |  |
| Marchena     | 51  | 23 | 129.49  | 343  | 29.1 | 85.9  | 59.56   |  |
| Onslow       | 2   | 2  | 0.01    | 25   | 3.3  | 45.9  | 0.10    |  |
| Pinta        | 104 | 37 | 59.56   | 777  | 29.1 | 119.6 | 129.49  |  |
| Pinzon       | 108 | 33 | 17.95   | 458  | 10.7 | 10.7  | 0.03    |  |
| Las.Plazas   | 12  | 9  | 0.23    | 94   | 0.5  | 0.6   | 25.09   |  |
| Rabida       | 70  | 30 | 4.89    | 367  | 4.4  | 24.4  | 572.33  |  |
| SanCristobal | 280 | 65 | 551.62  | 716  | 45.2 | 66.6  | 0.57    |  |
| SanSalvador  | 237 | 81 | 572.33  | 906  | 0.2  | 19.8  | 4.89    |  |
| SantaCruz    | 444 | 95 | 903.82  | 864  | 0.6  | 0.0   | 0.52    |  |
| SantaFe      | 62  | 28 | 24.08   | 259  | 16.5 | 16.5  | 0.52    |  |
| SantaMaria   | 285 | 73 | 170.92  | 640  | 2.6  | 49.2  | 0.10    |  |
| Seymour      | 44  | 16 | 1.84    | 147  | 0.6  | 9.6   | 25.09   |  |
| Tortuga      | 16  | 8  | 1.24    | 186  | 6.8  | 50.9  | 17.95   |  |
| wolf         | 21  | 12 | 2.85    | 253  | 34.1 | 254.7 | 2.33    |  |

Change the working directory to “c:/mydata” (do this by choosing “File”->“Change directory”, and then inputting the directory name). And now you can simply read the data into R by:

```
> gala <- read.table("gala.txt") # read the data into R
```

Now we can take a look at the data.

```
> data(gala)
> gala
  Species Endemics Area Elevation Nearest Scruz Adjacent
```

|                       |    |    |       |     |      |       |        |
|-----------------------|----|----|-------|-----|------|-------|--------|
| Baltra                | 58 | 23 | 25.09 | 346 | 0.6  | 0.6   | 1.84   |
| Bartolome             | 31 | 21 | 1.24  | 109 | 0.6  | 26.3  | 572.33 |
| --- cases deleted --- |    |    |       |     |      |       |        |
| Tortuga               | 16 | 8  | 1.24  | 186 | 6.8  | 50.9  | 17.95  |
| Wolf                  | 21 | 12 | 2.85  | 253 | 34.1 | 254.7 | 2.33   |

The variables are

|                  |   |
|------------------|---|
| <b>Species</b>   | The number of species of tortoise found on the island |
| <b>Endemics</b>  | The number of endemic species                         |
| <b>Elevation</b> | The highest elevation of the island (m)               |
| <b>Nearest</b>   | The distance from the nearest island (km)             |
| <b>Scruz</b>     | The distance from Santa Cruz island (km)              |
| <b>Adjacent</b>  | The area of the adjacent island (km <sup>2</sup> )    |

The data were presented by Johnson and Raven (Johnson & Raven, 1973) and also appear in Weisberg (Weisberg, 2005). We have filled in some missing values for simplicity. Fitting a linear model in R is done using the `lm()` command. Notice the syntax for specifying the predictors in the model. This is the so-called Wilkinson-Rogers notation. In this case, since all the variables are in the `gala` data frame, we must use the `data=` argument.

First, we generate a series of plots of various species-area relations. We consider three models: Linear, Gleason, and log-Arrhenius.

1. Linear model:  $S \sim \text{Normal}(\mu, \sigma^2)$  with identity link such that  $\mu = \beta_0 + \beta_1 A$ .
2. Gleason model:  $S \sim \text{Normal}(\mu, \sigma^2)$  with identity link such that  $\mu = \beta_0 + \beta_1 \log A$ .
3. log-Arrhenius model:  $S \sim \text{Normal}(\mu, \sigma^2)$  with identity link such that  $\mu = \beta_0 A^{\beta_1}$ .

Model 1 is a Linear Model. We fit a linear model on the original scale, model 1 above and obtain the log-likelihood (with **logLik**) and the AIC (with **AIC**).

```
> model1<-lm(Species~Area,data=gala)
> logLik(model1)
'log Lik.' -177.0993 (df=3)
> AIC(model1)
[1] 360.1985
```

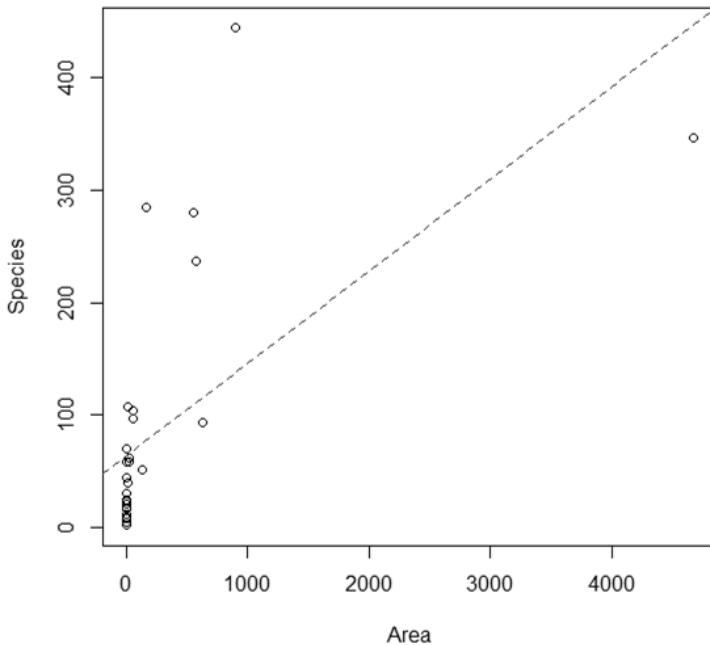
We superimpose the fitted model on a scatter plot of the data.

```
> plot(gala$Area,gala$species, xlab='Area', ylab='Species')
> abline(model1,col=2, lty=2)
> mtext('Model 1: linear model', side=3, line=.5)
```

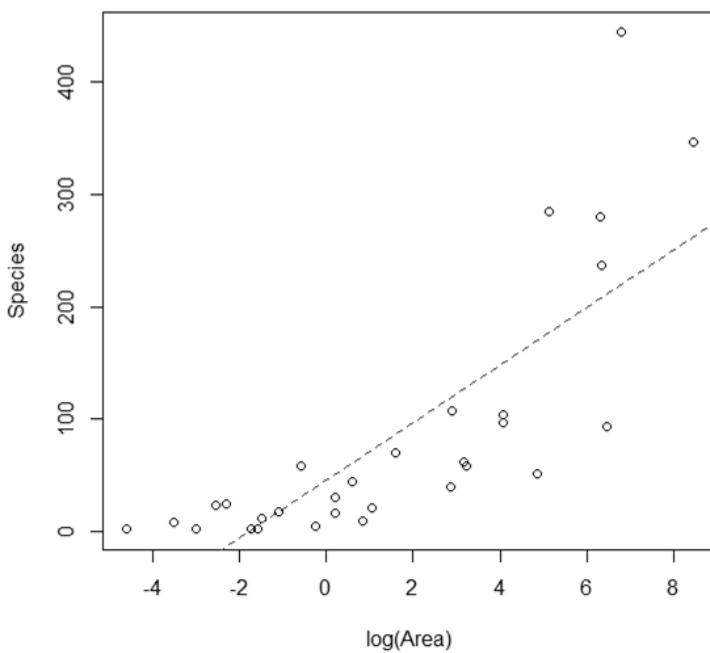
Model 2 is a Gleason Model. The Gleason model requires a log-transformed predictor, but an untransformed response.

```
> model2<-lm(Species~log(Area),data=gala)
> logLik(model2)
'log Lik.' -169.9574 (df=3)
> AIC(model2)
[1] 345.9147
> plot(log(gala$Area), gala$Species, xlab='log(Area)',
+       ylab='Species')
> abline(model2,col=2,lty=2)
> mtext('Model 2: Gleason model', side=3, line=.5)
```

Model 1: linear model



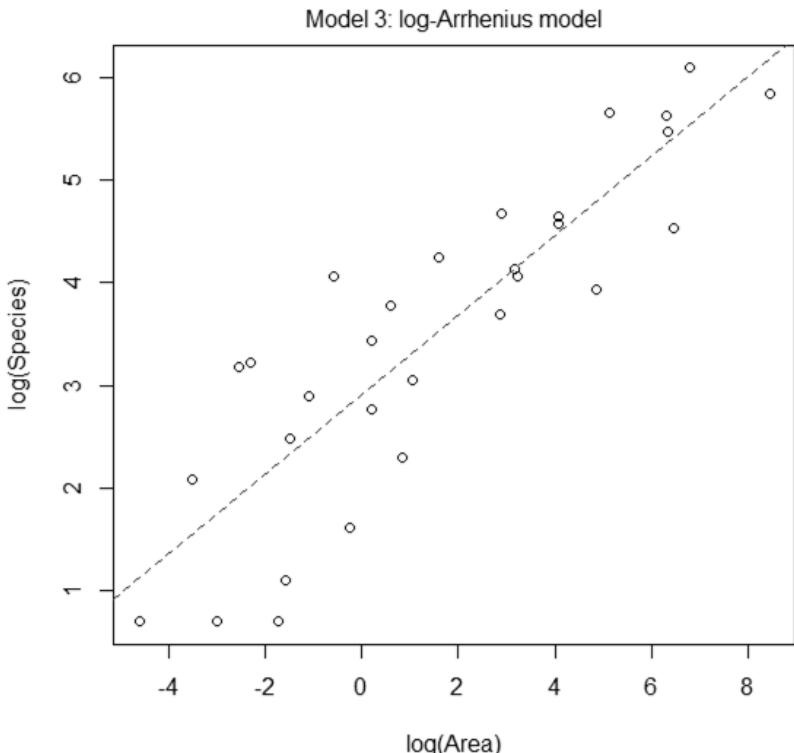
Model 2: Gleason model



Model 3 is a log-Arrhenius. The log-Arrhenius model is just an ordinary regression model in which both the response and predictor are log-transformed.

```
> model3<-lm(log(Species)~log(Area), data=gala)
> logLik(model3)
'log Lik.' -34.23973 (df=3)
> AIC(model3)
[1] 74.47946

> plot(log(gala$Area), log(gala$Species), xlab='log(Area)',
+       ylab='log(Species)')
> abline(model3,col=2, lty=2)
> mtext('Model 3: log-Arrhenius model', side=3, line=.5)
```



Now we want to compare the models. We begin by collecting the names of our models.

```
> model.names<-c('Linear', 'Gleason', 'Log-Arrhenius')
```

Next we concatenate all the calculated log-likelihoods together.

```
> loglike <-c(logLik(model1),logLik(model2),logLik(model3))
```

Next we concatenate a list of the number of parameters estimated for each model. Each model has three parameters— $\beta_0$ ,  $\beta_1$ , and  $\sigma^2$ .

```
> numparms<-c(3,3,3)
```

Finally, we create our function to carry out the calculations.

```
# LL is log likelihood,
# K is number of estimated parameters
# n is the sample size
> AIC.func<-function(LL,K,n,modelnames)
+ {
+   AIC<- -2*LL + 2*K
+   AICC<-AIC + 2*K*(K+1)/(n-K-1)
+   output<-cbind(LL,K,AIC,AICC)
+   colnames(output)<-c('LogL','K','AIC','AICC')
+   minAICC<-min(output[, "AICC"])
+   deltai<-output[, "AICC"]-minAICC
+   rel.like<-exp(-deltai/2)
+   wi<-round(rel.like/sum(rel.like),3)
+   out<-data.frame(modelnames,output,deltai,wi)
+   out
+ }
```

```
> AIC.func(loglike,numparms,dim(gala)[1],model.names)
```

|   | modelnames    | LogL       | K | AIC       | AICC      | deltai   | wi |
|---|---------------|------------|---|-----------|-----------|----------|----|
| 1 | Linear        | -177.09927 | 3 | 360.19853 | 361.12161 | 285.7191 | 0  |
| 2 | Gleason       | -169.95737 | 3 | 345.91473 | 346.83781 | 271.4353 | 0  |
| 3 | Log-Arrhenius | -34.23973  | 3 | 74.47946  | 75.40253  | 0.0000   | 1  |

Based on the output we see there is only one model that has any empirical support, the log-Arrhenius model. For reasons we will not explain here, although the log-Arrhenius model fits well, it is not guaranteed to be optimal. For the Galapagos data, a model that allows there to be heteroscedasticity on the scale of the original response is to be preferred.

The previous models only examine the Species-Area relationship. We now consider additional variables: Endemics, Elevation, Nearest, Scruz, and Adjacent. We will call our model gfit.

```
> gfit <- lm(Species~Area + Elevation + Nearest + Scruz +
+     Adjacent, data=gala)
> summary(gfit)

Call:
> lm(formula = Species~Area + Elevation + Nearest + Scruz +
+     Adjacent, data=gala)
> summary(gfit)

Call:
lm(formula = Species ~ Area + Elevation + Nearest + Scruz +
Adjacent,
  data = gala)

Residuals:
    Min      1Q      Median      3Q      Max 
-111.679 -34.898   -7.862   33.460  182.584 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.068221  19.154198  0.369 0.715351    
Area        -0.023938   0.022422 -1.068 0.296318    
Elevation    0.319465   0.053663  5.953 3.82e-06 ***  
Nearest      0.009144   1.054136  0.009 0.993151    
Scruz        -0.240524   0.215402 -1.117 0.275208    
Adjacent     -0.074805   0.017700 -4.226 0.000297 ***  
---

```

```
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’  
1
```

```
Residual standard error: 60.98 on 24 degrees of freedom  
Multiple R-squared:  0.7658,   Adjusted R-squared:  0.7171  
F-statistic: 15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

We can identify several useful quantities in this output. Other statistical packages tend to produce output quite similar to this. One useful feature of R is that it is possible to directly calculate quantities of interest. Of course, it is not necessary here because the `lm()` function does the job but it is very useful when the statistic you want is not part of the pre-packaged functions.

First we make the  $X$ -matrix

```
> x <- cbind(1,gala[,-c(1,2)])
```

and here's the response  $y$ :

```
> y <- gala$species
```

Now let's construct  $X'X$ : `t()` does transpose and `%*%` does matrix multiplication:

```
> t(x) %*% x  
Error: %*% requires numeric matrix/vector arguments
```

Gives a somewhat cryptic error. The problem is that matrix arithmetic can only be done with numeric values but  $x$  here derives from the data frame type. Data frames are allowed to contain character variables, which would disallow matrix arithmetic. We need to force  $x$  into the matrix form:

```
> x <- as.matrix(x)  
> t(x) %*% x  
           1          Area  Elevation  Nearest  
1      30.00     7851.26    11041.0    301.80  
Area  7851.26  23708665.46  10852798.5  39240.84  
Elevation 11041.00  10852798.53  9218227.0  109139.20  
Nearest  301.80    39240.84    109139.2    8945.30  
Scruz   1709.30   275516.84    616237.8   34527.34  
Adjacent 7832.95  5950313.65  8553187.9   37196.67  
           Scruz     Adjacent
```

```

1          1709.30    7832.95
Area      275516.84  5950313.65
Elevation 616237.80   8553187.95
Nearest    34527.34    37196.67
Scruz     231613.77   534409.98
Adjacent  534409.98  23719568.46

```

Inverses can be taken using the `solve()` command:

```

> xtxi <- solve(t(x) %*% x)
> xtxi

```

|           | 1             | Area          | Elevation     |
|-----------|---------------|---------------|---------------|
| 1         | 9.867829e-02  | 3.778242e-05  | -1.561976e-04 |
| Area      | 3.778242e-05  | 1.352247e-07  | -2.593617e-07 |
| Elevation | -1.561976e-04 | -2.593617e-07 | 7.745339e-07  |
| Nearest   | -2.339027e-04 | 1.294003e-06  | -3.549366e-06 |
| Scruz     | -3.760293e-04 | -4.913149e-08 | 3.080831e-07  |
| Adjacent  | 2.309832e-05  | 4.620303e-08  | -1.640241e-07 |
|           | Nearest       | Scruz         | Adjacent      |
| 1         | -2.339027e-04 | -3.760293e-04 | 2.309832e-05  |
| Area      | 1.294003e-06  | -4.913149e-08 | 4.620303e-08  |
| Elevation | -3.549366e-06 | 3.080831e-07  | -1.640241e-07 |
| Nearest   | 2.988732e-04  | -3.821077e-05 | 1.424729e-06  |
| Scruz     | -3.821077e-05 | 1.247941e-05  | -1.958356e-07 |
| Adjacent  | 1.424729e-06  | -1.958356e-07 | 8.426543e-08  |

A somewhat more direct way to get  $(X'X)^{-1}$  is as follows:

```

> gfit <- lm(Species~Area+Elevation+Nearest+Scruz+Adjacent,
+               data=gala)
> gs <- summary(gfit)
> gs$cov.unscaled


|             | (Intercept)   | Area          | Elevation     |
|-------------|---------------|---------------|---------------|
| (Intercept) | 9.867829e-02  | 3.778242e-05  | -1.561976e-04 |
| Area        | 3.778242e-05  | 1.352247e-07  | -2.593617e-07 |
| Elevation   | -1.561976e-04 | -2.593617e-07 | 7.745339e-07  |
| Nearest     | -2.339027e-04 | 1.294003e-06  | -3.549366e-06 |
| Scruz       | -3.760293e-04 | -4.913149e-08 | 3.080831e-07  |
| Adjacent    | 2.309832e-05  | 4.620303e-08  | -1.640241e-07 |
|             | Nearest       | Scruz         | Adjacent      |
| (Intercept) | -2.339027e-04 | -3.760293e-04 | 2.309832e-05  |


```

|           |               |               |               |
|-----------|---------------|---------------|---------------|
| Area      | 1.294003e-06  | -4.913149e-08 | 4.620303e-08  |
| Elevation | -3.549366e-06 | 3.080831e-07  | -1.640241e-07 |
| Nearest   | 2.988732e-04  | -3.821077e-05 | 1.424729e-06  |
| Scruz     | -3.821077e-05 | 1.247941e-05  | -1.958356e-07 |
| Adjacent  | 1.424729e-06  | -1.958356e-07 | 8.426543e-08  |

The `names()` command is the way to see the components of an Splus object - you can see that there are other useful quantities that are directly available:

```
> names(gs)
> names(gfit)
```

In particular, the fitted (or predicted) values and residuals are

```
> gfit$fit
      Baltra    Bartolome    Caldwell    Champion
 116.7259460   -7.2731544   29.3306594   10.3642660
      Coamano Daphne.Major Daphne.Minor     Darwin
 -36.3839155   43.0877052   33.9196678  -9.0189919
      Eden     Enderby    Espanola Fernandina
 28.3142017   30.7859425   47.6564865  96.9895982
      Gardner1   Gardner2    Genovesa    Isabela
 -4.0332759   64.6337956   -0.4971756  386.4035578
      Marchena    Onslow      Pinta     Pinzon
 88.6945404   4.0372328   215.6794862  150.4753750
      Las.Plazas    Rabida SanCristobal SanSalvador
 35.0758066   75.5531221   206.9518779  277.6763183
      SantaCruz    SantaFe   SantaMaria Seymour
 261.4164131   85.3764857   195.6166286  49.8050946
      Tortuga        wolf
 52.9357316   26.7005735

> gfit$res
      Baltra    Bartolome    Caldwell    Champion
 -58.725946   38.273154  -26.330659   14.635734
      Coamano Daphne.Major Daphne.Minor     Darwin
  38.383916  -25.087705  -9.919668  19.018992
      Eden     Enderby    Espanola Fernandina
 -20.314202  -28.785943  49.343513  -3.989598
      Gardner1   Gardner2    Genovesa    Isabela
```

|            |            |              |             |
|------------|------------|--------------|-------------|
| 62.033276  | -59.633796 | 40.497176    | -39.403558  |
| Marchena   | Onslow     | Pinta        | Pinzon      |
| -37.694540 | -2.037233  | -111.679486  | -42.475375  |
| Las.Plazas | Rabida     | SanCristobal | SanSalvador |
| -23.075807 | -5.553122  | 73.048122    | -40.676318  |
| SantaCruz  | SantaFe    | SantaMaria   | Seymour     |
| 182.583587 | -23.376486 | 89.383371    | -5.805095   |
| Tortuga    | wolf       |              |             |
| -36.935732 | -5.700573  |              |             |

We can get  $\hat{\beta}$  directly:

```
> xtxi %*% t(x) %*% y
      [,1]
1      7.068220709
Area   -0.023938338
Elevation 0.319464761
Nearest  0.009143961
Scruz   -0.240524230
Adjacent -0.074804832
```

or in a computationally efficient and stable manner:

```
> solve(t(x) %*% x, t(x) %*% y)
      [,1]
[1,] 7.068220709
[2,] -0.023938338
[3,] 0.319464761
[4,] 0.009143961
[5,] -0.240524230
[6,] -0.074804832
```

We can estimate  $\sigma$  using the estimator in the text:

```
> root1<-sum((gfit$res)^2)
> sqrt(root1/(30-6))
[1] 60.97519
```

Compare this to the results above (Residual standard error).

We may also obtain the standard errors for the coefficients. Also `diag()` returns the diagonal of a matrix):

```
> sqrt(diag(xtxi))*60.97519
      1       Area   Elevation     Nearest
19.15419834 0.02242235 0.05366281 1.05413598
      Scruz   Adjacent
0.21540225 0.01770019
```

Finally we may compute  $R^2$ :

```
> 1-sum((gfit$res)^2)/sum((y-mean(y))^2)
[1] 0.7658469
```

## 12.7. Least-square data fitting with *SCILAB* function *datafit*

### 12.7.1. The method of least squares

The method of least squares consists in finding a number of parameters,  $a_1, a_2, \dots, a_m$ , that define a function

$$y_p = f(a_1, a_2, \dots, a_m, x_1, x_2, \dots, x_n)$$

to fit to the data set represented in the following table:

| $x_1$     | $x_2$     | ... | $x_n$     | $y$   |
|-----------|-----------|-----|-----------|-------|
| $x_{1,1}$ | $x_{1,2}$ | ... | $x_{1,n}$ | $y_1$ |
| $x_{2,1}$ | $x_{2,2}$ | ... | $x_{2,n}$ | $y_2$ |
| .         | .         | .   | .         | .     |
| .         | .         | .   | .         | .     |
| $x_{k,1}$ | $x_{k,2}$ | ... | $x_{k,n}$ | $y_n$ |

The error,  $e_i$ , incurred in fitting the data to the function is the difference between the actual value  $y_i$  and the fitted value  $y_{pi}$ , i.e.,

$$e_i = y_i - y_{pi}$$

The method of least squares seeks to find the values of the parameters  $\mathbf{a} = [a_1, a_2, \dots, a_m]$  that minimize the Sum of Squared Errors function,  $SSE(p_1, p_2, \dots, p_m)$ , defined as

$$SSE(\mathbf{p}) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - y_{p_i})^2$$

To solve for  $a_1, a_2, \dots, a_m$ , we form the  $m$  equations

$$\frac{\partial(SSE)}{\partial a_1} = 0, \quad \frac{\partial(SSE)}{\partial a_2} = 0, \quad \dots, \quad \frac{\partial(SSE)}{\partial a_m} = 0,$$

and solve them simultaneously for the parameters  $a_1, a_2, \dots, a_m$ .

### 12.7.2. Function *datafit*

*SCILAB* provides function *datafit* to produce least-square fittings for any type of functions. The general call to this function, in its simplest form, is:

```
[a, SSE]=datafit(G, Z, a0)
```

where  $a$  is a column vector containing the fitted parameters;  $SSE$  is the sum of square errors corresponding to the fitted parameters;  $G$  is a function representing the error, i.e.,

```
e = G(a, z) = y - f(a, x) = z(n+1) -  
f(a(1), a(2), ..., a(m), z(1), z(2), ..., z(n)),
```

$z$  is a vector of variables, with  $z(1) = x(1)$ ,  $z(2) = x(2)$ , ...,  $z(n) = x(n)$ , and  $z(n+1) = y$ .  $Z$  is a matrix whose rows are the lists of values of  $x(1)$ ,  $x(2)$ , ...,  $x(n)$ , and  $y$ , i.e., the columns of the table shown above:

```
x1 = [x(1,1), x(2,1), ..., x(k,1)];  
x2 = [x(1,2), x(2,2), ..., x(k,2)];  
. . .  
xn = [x(1,n), x(2,n), ..., x(k,n)];  
y = [y(1), y(2), ..., y(n)];
```

The matrix  $Z$  is put together as

```
z = [x1;x2;...;y];
```

and,  $a_0$  is a column vector with initial guesses of the parameters  $a$ .

To find additional information on function 'datafit' use:

```
help datafit
```

### EXAMPLE 3. Function *datafit* - quadratic function

To illustrate the use of function *datafit*, we first generate data based on the quadratic equation  $Y=20+30*x+50*x^2$ :

```
x = [0:0.1:10]; Y = 20+30*x+50*x^2;
```

Next, we add a random component given by:

```
E = 1000*(rand(1,length(x))-0.5);
```

The y-data to be used is calculated as:

```
Y1 = Y+E;
```

A plot of the data set  $(X, Y_1)$  is shown against the original function  $(X, Y)$ :

```
plot(X,Y,'-',X,Y1,'+')
```

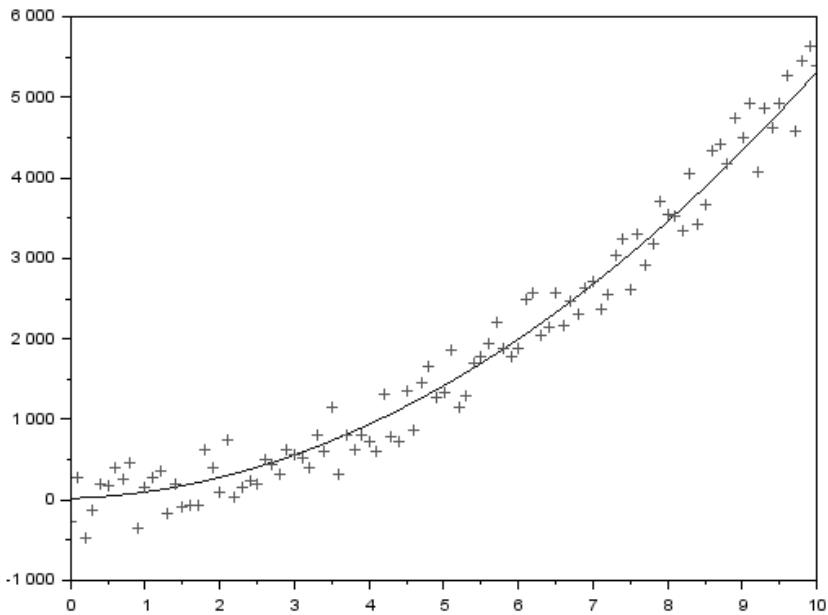
To use '*datafit*', we first form matrix  $Z$ :

```
z = [X;Y1];
```

and define the error function as follows:

```
deff('[e]=G(a,z)', 'e=z(2)-a(1)-a(2)*z(1)-a(3)*z(1)^2')
```

Notice that, in this case, we went from the expression  $y = a_1 + a_{2x} + a_{3x}^2$ , to  $z_2 = a_1 + a_2 z_1 + a_3 z_1^2$ , from where the error function is obtained as  $e = z_2 - a_1 - a_2 z_1 - a_3 z_1^2$ . Use as initial guess of the parameters:



**Figure 1.** Quadratic function with random error

```
a0 = [1;1;1]
a0 =
    1.
    1.
    1.
```

Calling function '*datafit*' (Note - it takes a couple of minutes to produce a result):

```
aa,SSE] = datafit(G,Z,a0)
SSE =
    7051914.9
aa =
    34.407874
    - 5.2058162
    54.785766
```

The result shows the SSE for the fitted parameters. To check how well the fitted data follows the original data put together the function  $f(x)$ :

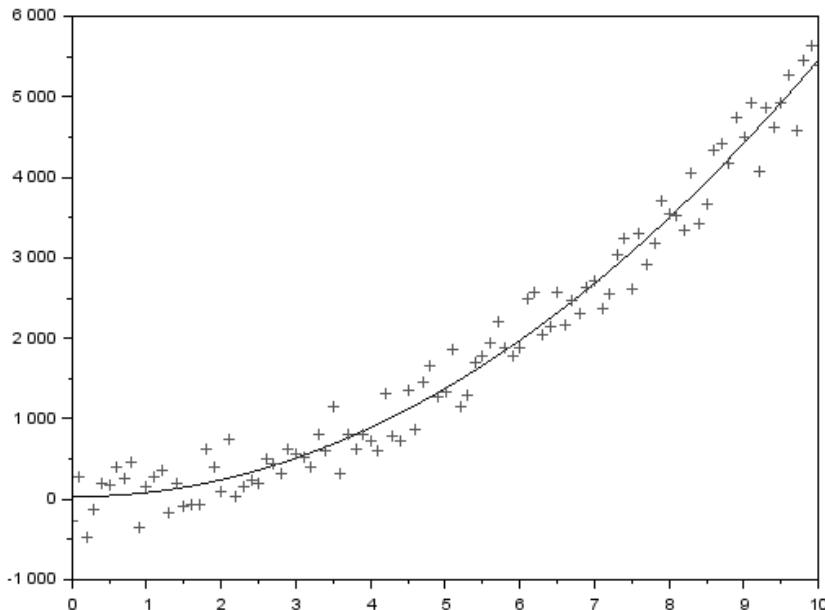
```
deff('[y]=f(x)', 'y=aa(1)+aa(2)*x+aa(3)*x^2')
```

and calculate the corresponding values:

```
YY = f(x);
```

The following plot shows the fitted data and the original data points:

```
fig(2);plot(x,YY,'-',x,Y1,'+')
```



**Figure 2.** Quadratic fitting and original data

#### EXAMPLE 4. Function *datafit* - cubic function

Here we produce a cubic polynomial fitting using the same data  $X, Y_1$ , that we used earlier. The error function is now:

```
deff('[e]=G(a,z)', 'e=z(2)-a(1)-a(2)*z(1)-a(3)*z(1)^2-a(4)*z(1)^3')
```

and the initial guess for the parameters gets changed to:

```
a0 = [15;25; 49; 10]  
a0 =
```

```
15.
```

```
25.  
49.  
10.
```

Calling function 'datafit' produces, after a few minutes, the following results:

```
[aa,SSE] = datafit(G,z,a0)  
SSE =  
  
7047076.3  
aa =  
  
17.149817  
16.033299  
49.449569  
0.3557430
```

Next, we generate data from the fitted function to compare to the original data by defining the function  $f(x)$ :

```
deff('[y]=f(x)', 'y=aa(1)+aa(2)*x+aa(3)*x^2+aa(4)*x^3')
```

and generating vector YYY:

```
YYY = f(X);
```

A plot of the cubic fitted data and the original data is shown next:

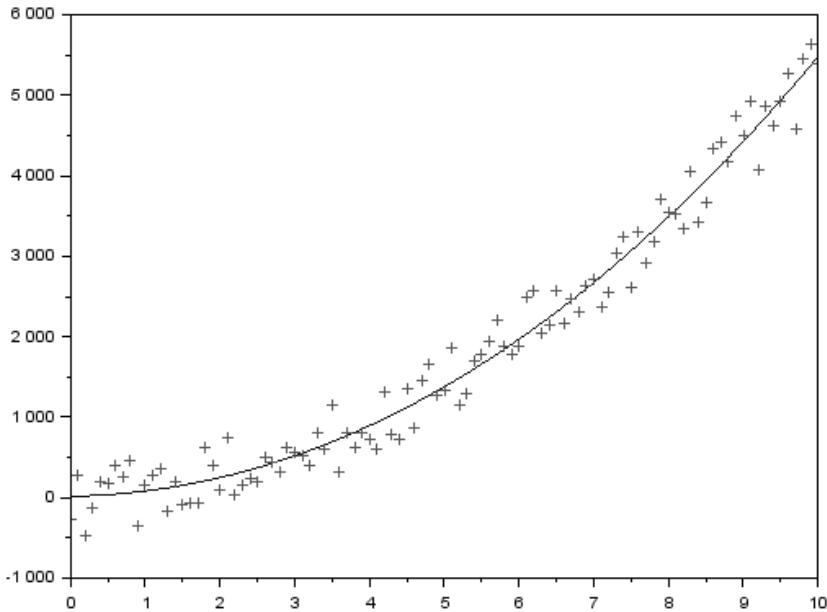
```
fig(3);plot(X,YYY,'-',X,Y1,'+')
```

#### EXAMPLE 5. Function datafit - exponential function

In this exercise we use an exponential function to fit the original data set  $X, Y_1$ . First, define the error function:

```
deff('[e]=G(a,z)', 'e=z(2)-a(1)-a(2)*exp(a(3)*z(1))')
```

Re-define the initial guess for the parameters:



**Figure 3.** Cubic fitting and original data

```
a0 = [5;1;2]
a0 =
    5.
    1.
    2.
```

Call function 'datafit' to obtain the fitted parameters:

```
[aa,SSE] = datafit(G,Z,a0)
SSE =
    12150408.
aa =
    6.454563
    296.382
    0.2999036
```

Define the exponential function using the parameters found with 'datafit':

```
deff(['y']=f(x)', 'y=aa(1)+aa(2)*exp(aa(3)*x)')
```

Calculate the vector of fitted values:

```
YYYY = f(x);
```

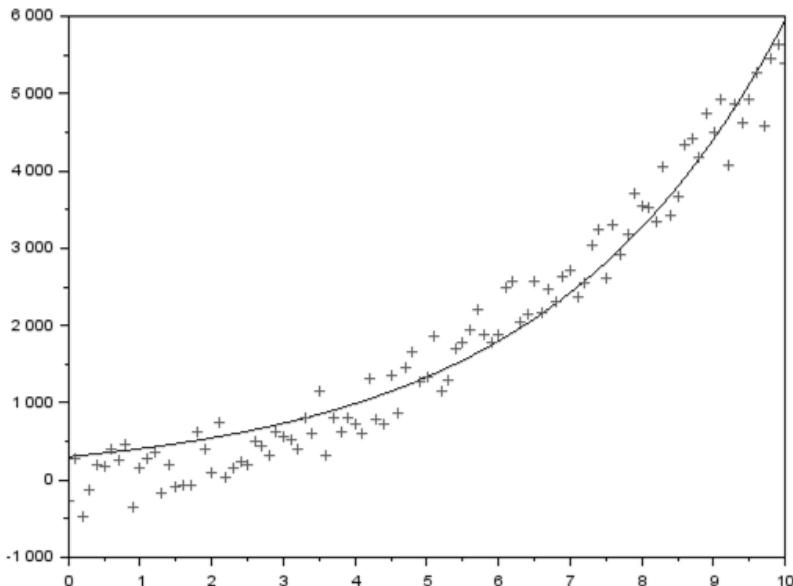
Plot the exponentially-fitted data and the original data in the same set of axes:

```
fig(5);plot(x,YYYY,'-',x,Y1,'+')
```

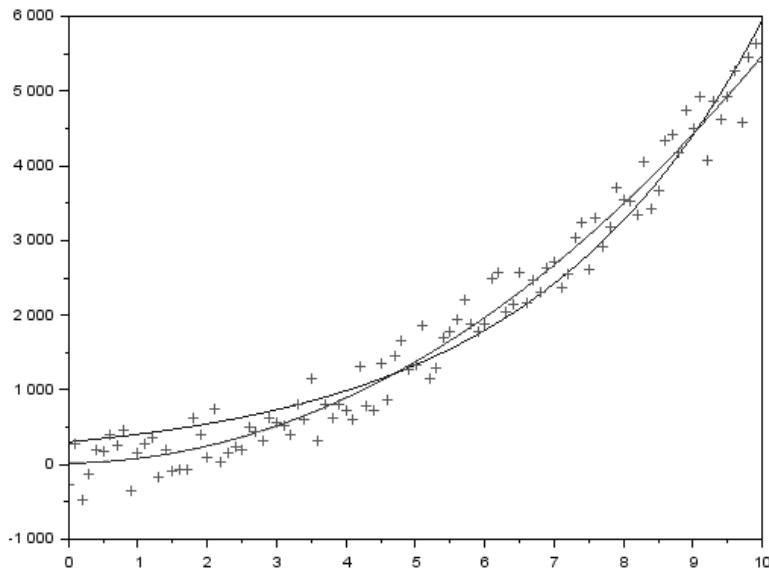
Next, we produce a graph comparing the original data and the three fittings calculated above:

```
fig(6);plot(x,YYYY,'-',x,Y1,'+')  
hold()  
Current plot held  
plot(x,YY,'g--',x,YYY,'r-')
```

The graphs are shown in the next page:



**Figure 4.** Exponential fitting and original data



**Figure 5.** Quadratic, cubic, and exponential fitting and original data

#### EXAMPLE 6. Function datafit– multiple linear regression

Function datafit can be used to determine the coefficients of a multiple linear fitting as illustrated in the example below. The data to be fitted is given in the following table:

| x1 | x2 | x3 | x4  | y   |
|----|----|----|-----|-----|
| 25 | 24 | 91 | 100 | 240 |
| 31 | 21 | 90 | 95  | 236 |
| 45 | 24 | 88 | 110 | 290 |
| 60 | 25 | 87 | 88  | 274 |
| 65 | 25 | 91 | 94  | 301 |
| 72 | 26 | 94 | 99  | 316 |
| 80 | 25 | 87 | 97  | 300 |
| 84 | 25 | 86 | 96  | 296 |
| 75 | 24 | 88 | 110 | 267 |
| 60 | 25 | 91 | 105 | 276 |
| 50 | 25 | 90 | 100 | 288 |
| 38 | 23 | 89 | 98  | 261 |

First, we load the data and prepare the matrix  $X$  for the application of function *datafit*. The matrix  $XX$ , shown below, is loaded following the rows in the table above, and then transposed to make it fit the requirements for matrix  $X$  in the call to function *datafit*:

```
XX =...
[25 24 91 100 240
31 21 90 95 236
45 24 88 110 290
60 25 87 88 274
65 25 91 94 301
72 26 94 99 316
80 25 87 97 300
84 25 86 96 296
75 24 88 110 267
60 25 91 105 276
50 25 90 100 288
38 23 89 98 261];
X=XX';
```

We seek a function of the form  $y = a_1 + a_2x_1 + a_3x_2 + a_4x_3 + a_5x_4$ , thus, we define the error function to be minimized as follows [with  $z(1)=x(1)$ ,  $z(2)=x(2)$ ,  $z(3)=x(3)$ ,  $z(4)=x(4)$ , and  $z(5)=y$ ]:

```
deff('[e]=G(a,z)', 'e=a(1)+a(2)*z(1)+a(3)*z(2)+a(4)*z(3)+a(5)
)*z(4)-z(5)')
```

Next, we call function *datafit*, with an initial guess for  $a$  given by  $a0$ :

```
a0 = [1;2;3;4;5];
[aa,SSE]=datafit(G,X,a0)
SSE =
1699.0093
aa =
- 102.70925
0.6053692
8.9236379
1.4374193
0.0136051
```

The following script file contain commands that are used to produce a table of the original values of  $y$ , the fitted values of  $y(yp)$ , and the error,  $e_i = y_i - yp_i$ :

```
deff('[y]=f(x)',...
'y=aa(1)+aa(2)*x(1)+aa(3)*x(2)+aa(4)*x(3)+aa(5)*x(4)')
yp = [];
[m n] = size(xx);
for i = 1:m
xx = xx(i,1:4);
yp = [yp;f(xx)];
end
y = xx(:,5);
e = y-yp;
mode(-1); //Start command suppression mode
printf("====")
printf("point y yp e ")
printf("====")
for i = 1:m
printf("%4d %6.2f %6.2f %6.2f",i,y(i),yp(i),e(i))
end
printf("====")
mode(1); //End command suppression mode
//End of script
```

The *SCI**LAB* output for this script is shown next:

```
-->deff('[y]=f(x)',...
-->'y=aa(1)+aa(2)*x(1)+aa(3)*x(2)+aa(4)*x(3)+aa(5)*x(4)')
-->yp = [];

-->[m n] = size(xx);

-->for i = 1:m
-->    xx = xx(i,1:4);
-->    yp = [yp;f(xx)];
-->end

-->y = xx(:,5);
-->e = y-yp;
-->mode(-1); //Start command suppression mode
=====

point y yp e
=====
```

```
1 240.00 258.76 -18.76
2 236.00 234.11 1.89
3 290.00 266.69 23.31
4 274.00 282.96 -8.96
5 301.00 291.81 9.19
6 316.00 309.36 6.64
7 300.00 295.19 4.81
8 296.00 296.16 -0.16
9 267.00 284.85 -17.85
10 276.00 288.94 -12.94
11 288.00 281.38 6.62
12 261.00 254.80 6.20
=====
-->//End of script
```

## 12.8. Review Problems

Work Examples 1 – 4 using R.



# 13.What is a Generalized Linear Model?

In statistics, the generalized linear model (GLM)—not to be confused with general linear model or generalized least squares—is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.

Generalized linear models were formulated by John Nelder and Robert Wedderburn as a way of unifying various other statistical models, including linear regression, logistic regression and Poisson regression (Nelder & Wedderburn, 1972). They proposed an iteratively reweighted least squares method for maximum likelihood estimation of the model parameters. Maximum-likelihood estimation remains popular and is the default method on many statistical computing packages. Other approaches, including Bayesian approaches and least squares fits to variance stabilized responses, have been developed.

## 13.1. Intuition

Ordinary linear regression predicts the expected value of a given unknown quantity (the response variable, a random variable) as a linear combination of a set of observed values (predictors). This implies that a constant change in a predictor leads to a constant change in the response variable (i.e. a linear-response model). This is appropriate when the response variable has a normal distribution (intuitively, when a response variable can vary essentially indefinitely in either direction with no fixed “zero value”, or more generally for any quantity that only varies by a relatively small amount, e.g. human heights).

However, these assumptions are inappropriate for many types of response variables. For example, in many cases when the response variable must be positive and can vary over a wide scale, constant input changes lead to geometrically varying rather than constantly varying

output changes. As an example, a model that predicts that each decrease in 10 degrees Fahrenheit leads to 1,000 fewer people going to a given beach is unlikely to generalize well over both small beaches (e.g. those where the expected attendance was 50 at the lower temperature) and large beaches (e.g. those where the expected attendance was 10,000 at the lower temperature). An even worse problem is that, since the model also implies that a drop in 10 degrees leads 1,000 fewer people going to a given beach, a beach whose expected attendance was 50 at the higher temperature would now be predicted to have the impossible attendance value of -950. Logically, a more realistic model would instead predict a constant rate of increased beach attendance (e.g. an increase in 10 degrees leads to a doubling in beach attendance, and a drop in 10 degrees leads to a halving in attendance). Such a model is termed an exponential-response model (or log-linear model, since the logarithm of the response is predicted to vary linearly).

Similarly, a model that predicts a probability of making a yes/no choice (a Bernoulli variable) is even less suitable as a linear-response model, since probabilities are bounded on both ends (they must be between 0 and 1). Imagine, for example, a model that predicts the likelihood of a given person going to the beach as a function of temperature. A reasonable model might predict, for example, that a change in 10 degrees makes a person two times more or less likely to go to the beach. But what does “twice as likely” mean in terms of a probability? It cannot literally mean to double the probability value (e.g. 50% becomes 100%, 75% becomes 150%, etc.). Rather, it is the odds that are doubling: from 2:1 odds, to 4:1 odds, to 8:1 odds, etc. Such a model is a log-odds model.

Generalized linear models cover all these situations by allowing for response variables that have arbitrary distributions (rather than simply normal distributions), and for an arbitrary function of the response variable (the link function) to vary linearly with the predicted values (rather than assuming that the response itself must vary linearly). For example, the case above of predicted number of beach attendees would typically be modeled with a Poisson distribution and a log link, while the case of predicted probability of beach attendance would typically be modeled with a Bernoulli distribution (or binomial distribution,

depending on exactly how the problem is phrased) and a log-odds (or logit) link function.

## 13.2. Overview

In a **generalized linear model (GLM)**, each outcome of the dependent variables,  $\mathbf{Y}$ , is assumed to be generated from a particular distribution in the exponential family, a large range of probability distributions that includes the normal, binomial, Poisson and gamma distributions, among others. The mean,  $\boldsymbol{\mu}$ , of the distribution depends on the independent variables,  $\mathbf{X}$ , through:

$$E(\mathbf{Y}) = \boldsymbol{\mu} = g^{-1}(\mathbf{X}\boldsymbol{\beta}), \quad (13.1)$$

where  $E(\mathbf{Y})$  is the expected value of  $\mathbf{Y}$ ;  $\mathbf{X}\boldsymbol{\beta}$  is the linear predictor, a linear combination of unknown parameters,  $\boldsymbol{\beta}$ ;  $g$  is the link function.

In this framework, the variance is typically a function,  $V$ , of the mean:

$$\text{Var}(\mathbf{Y}) = V(\boldsymbol{\mu}) = V(g^{-1}(\mathbf{X}\boldsymbol{\beta})). \quad (13.2)$$

It is convenient if  $V$  follows from the exponential family distribution, but it may simply be that the variance is a function of the predicted value.

The unknown parameters,  $\boldsymbol{\beta}$ , are typically estimated with maximum likelihood, maximum quasi-likelihood, or Bayesian techniques.

## 13.3. Model components

The GLM consists of three elements:

1. A probability distribution from the exponential family.
2. A linear predictor  $\eta = \mathbf{X}\boldsymbol{\beta}$ .
3. A link function  $g$  such that  $E(\mathbf{Y}) = \boldsymbol{\mu} = g^{-1}(\eta)$ .

### 13.3.1. Probability distribution

The overdispersed exponential family of distributions is a generalization of the exponential family and exponential dispersion model of distributions and includes those probability distributions, parameterized

by  $\boldsymbol{\theta}$  and, whose density functions  $f$  (or probability mass function, for the case of a discrete distribution) can be expressed in the form

$$f_Y(y|\boldsymbol{\theta}, \tau) = h(y, \tau) \exp\left(\frac{\mathbf{b}(\boldsymbol{\theta})\mathbf{T}(y) - A(\boldsymbol{\theta})}{d(\tau)}\right). \quad (13.3a)$$

$\tau$ , called the dispersion parameter, typically is known and is usually related to the variance of the distribution. The functions  $h(y, \tau)$ ,  $\mathbf{b}(\boldsymbol{\theta})$ ,  $\mathbf{T}(y)$ ,  $A(\boldsymbol{\theta})$ , and  $d(\tau)$  are known. Many common distributions are in this family.

For scalar  $Y$  and  $\theta$ , this reduces to

$$f_Y(y|\theta, \tau) = h(y, \tau) \exp\left(\frac{b(\theta)\mathbf{T}(y) - A(\theta)}{d(\tau)}\right). \quad (13.3b)$$

$\boldsymbol{\theta}$  is related to the mean of the distribution. If  $\mathbf{b}(\boldsymbol{\theta})$  is the identity function, then the distribution is said to be in canonical form (or natural form). Note that any distribution can be converted to canonical form by rewriting  $\boldsymbol{\theta}$  as  $\boldsymbol{\theta}'$  and then applying the transformation  $\boldsymbol{\theta} = \mathbf{b}(\boldsymbol{\theta}')$ . It is always possible to convert  $A(\boldsymbol{\theta})$  in terms of the new parameterization, even if  $\mathbf{b}(\boldsymbol{\theta}')$  is not a one-to-one function; see comments in the page on the exponential family. If, in addition,  $\mathbf{T}(y)$  is the identity and  $\tau$  is known, then  $\boldsymbol{\theta}$  is called the canonical parameter (or natural parameter) and is related to the mean through

$$\boldsymbol{\mu} = E(Y) = \nabla A(\boldsymbol{\theta}). \quad (13.4a)$$

For scalar  $Y$  and  $\theta$ , this reduces to

$$\mu = E(Y) = A'(\theta). \quad (13.4b)$$

Under this scenario, the variance of the distribution can be shown to be (McCullagh & Nelder, 1989)

$$\text{Var}(Y) = \nabla \nabla^T A(\boldsymbol{\theta}) d(\tau). \quad (13.5a)$$

For scalar  $Y$  and  $\theta$ , this reduces to

$$\text{Var}(Y) = A''(\theta)d(\tau). \quad (13.5b)$$

### 13.3.2. Linear predictor

The linear predictor is the quantity which incorporates the information about the independent variables into the model. The symbol  $\eta$  (Greek “eta”) denotes a linear predictor. It is related to the expected value of the data (thus, “predictor”) through the link function.

$\eta$  is expressed as linear combinations (thus, “linear”) of unknown parameters  $\beta$ . The coefficients of the linear combination are represented as the matrix of independent variables  $X$ .  $\eta$  can thus be expressed as

$$\eta = X\beta. \quad (13.6)$$

### 13.3.3. Link function

The **link function** provides the relationship between the linear predictor and the mean of the distribution function. There are many commonly used link functions, and their choice can be somewhat arbitrary. It makes sense to try to match the domain of the link function to the range of the distribution function’s mean.

When using a distribution function with a canonical parameter  $\theta$ , the canonical link function is the function that expresses  $\theta$  in terms of  $\mu$ , i.e.  $\theta = b(\mu)$ . For the most common distributions, the mean  $\mu$  is one of the parameters in the standard form of the distribution’s density function, and then  $b(\mu)$  is the function as defined above that maps the density function into its canonical form. When using the canonical link function,  $b(\mu) = \theta = X\beta$ , which allows  $X^T Y$   $X^T Y$  to be a sufficient statistic for  $\beta$ .

Following is a table of several exponential-family distributions in common use and the data they are typically used for, along with the canonical link functions and their inverses (sometimes referred to as the mean function, as done here).

Common distributions with typical uses and canonical link functions

| Distribution     | Support of distribution   | Typical uses  | Link name       | Link function                                | Mean function                    |
|------------------|---|---|-----------------|--|----------------------------------|
| Normal           | real: $(-\infty, \infty)$   | Linear-response data  | Identity        | $X\beta = \mu$                               | $\mu = X\beta$                   |
| Exponential      | real: $(0, \infty)$   | Exponential-response data, scale parameters                                       | Inverse         | $X\beta = -\mu^{-1}$                         | $\mu = (-X\beta)^{-1}$           |
| Gamma            |   |   |                 |  |                                  |
| Inverse Gaussian | real: $(0, \infty)$   |   | Inverse squared | $X\beta = -\mu^{-2}$                         | $\mu = (-X\beta)^{-1/2}$         |
| Poisson          | integer: $(0, \infty)$  | count of occurrences in fixed amount of time/space                                | Log             | $X\beta = \ln(\mu)$                          | $\mu = e^{X\beta}$               |
| Bernoulli        | integer: $[0,1]$  | outcome of single yes/no occurrence   | Logit           | $X\beta = \ln\left(\frac{\mu}{1-\mu}\right)$ | $\mu = \frac{1}{1 + e^{X\beta}}$ |
| Binomial         | integer: $[0, N]$   | count of # of "yes" occurrences out of N yes/no occurrences                       |                 |  |                                  |
| Categorical      | integer: $[0, K]$<br>K-vector of integer: $[0,1]$ , where exactly one element in the vector has the value 1 | outcome of single K-way occurrence  |                 |  |                                  |
| Multinomial      | K-vector of $[0, N]$  | count of occurrences of different types (1 .. K) out of N total K-way occurrences |                 |  |                                  |

In the cases of the exponential and gamma distributions, the domain of the canonical link function is not the same as the permitted range of the

mean. In particular, the linear predictor may be negative, which would give an impossible negative mean. When maximizing the likelihood, precautions must be taken to avoid this. An alternative is to use a noncanonical link function.

Note also that in the case of the Bernoulli, binomial, categorical and multinomial distributions, the support of the distributions is not the same type of data as the parameter being predicted. In all of these cases, the predicted parameter is one or more probabilities, i.e. real numbers in the range [0,1]. The resulting model is known as logistic regression (or multinomial logistic regression in the case that  $K$ -way rather than binary values are being predicted).

For the Bernoulli and binomial distributions, the parameter is a single probability, indicating the likelihood of occurrence of a single event. The Bernoulli still satisfies the basic condition of the generalized linear model in that, even though a single outcome will always be either 0 or 1, the expected value will nonetheless be a real-valued probability, i.e. the probability of occurrence of a “yes” (or 1) outcome. Similarly, in a binomial distribution, the expected value is  $Np$ , i.e. the expected proportion of “yes” outcomes will be the probability to be predicted.

For categorical and multinomial distributions, the parameter to be predicted is a  $K$ -vector of probabilities, with the further restriction that all probabilities must add up to 1. Each probability indicates the likelihood of occurrence of one of the  $K$  possible values. For the multinomial distribution, and for the vector form of the categorical distribution, the expected values of the elements of the vector can be related to the predicted probabilities similarly to the binomial and Bernoulli distributions.

## 13.4. Fitting

### 13.4.1. Maximum likelihood

The maximum likelihood estimates can be found using an iteratively reweighted least squares algorithm using either a Newton–Raphson method with updates of the form:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathcal{J}^{-1}(\boldsymbol{\beta}^{(t)})u(\boldsymbol{\beta}^{(t)}), \quad (13.7)$$

where  $\mathcal{J}(\boldsymbol{\beta}^{(t)})$  is the observed information matrix (the negative of the Hessian matrix) and  $u(\boldsymbol{\beta}^{(t)})$  is the score function; or a Fisher's scoring method:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + I^{-1}(\boldsymbol{\beta}^{(t)})u(\boldsymbol{\beta}^{(t)}), \quad (13.8)$$

where  $I(\boldsymbol{\beta}^{(t)})$  is the Fisher information matrix. Note that if the canonical link function is used, then they are the same.

### 13.4.2. Bayesian methods

In general, the posterior distribution cannot be found in closed form and so must be approximated, usually using Laplace approximations or some type of Markov chain Monte Carlo method such as Gibbs sampling.

## 13.5. Examples

### 13.5.1. General linear models

A possible point of confusion has to do with the distinction between generalized linear models and the general linear model, two broad statistical models. The general linear model may be viewed as a special case of the generalized linear model with identity link and responses normally distributed. As most exact results of interest are obtained only for the general linear model, the general linear model has undergone a somewhat longer historical development. Results for the generalized linear model with non-identity link are asymptotic (tending to work well with large samples).

### 13.5.2. Linear regression

A simple, very important example of a generalized linear model (also an example of a general linear model) is linear regression. In linear regression, the use of the least-squares estimator is justified by the Gauss-Markov theorem, which does not assume that the distribution is normal.

From the perspective of generalized linear models, however, it is useful to suppose that the distribution function is the normal distribution with constant variance and the link function is the identity, which is the canonical link if the variance is known.

For the normal distribution, the generalized linear model has a closed form expression for the maximum-likelihood estimates, which is convenient. Most other GLMs lack closed form estimates.

### 13.5.3. Binomial data

When the response data,  $Y$ , are binary (taking on only values 0 and 1), the distribution function is generally chosen to be the Bernoulli distribution and the interpretation of  $\mu_i$  is then the probability,  $p$ , of  $Y_i$  taking on the value one.

There are several popular link functions for binomial functions; the most typical is the canonical logit link:

$$g(p) = \ln\left(\frac{p}{1-p}\right). \quad (13.9)$$

GLMs with this setup are logistic regression models (or logit models).

In addition, the inverse of any continuous cumulative distribution function (CDF) can be used for the link since the CDF's range is  $[0,1]$ , the range of the binomial mean. The normal CDF  $\Phi$  is a popular choice and yields the probit model. Its link is

$$g(p) = \Phi^{-1}(p). \quad (13.10)$$

The reason for the use of the probit model is that a constant scaling of the input variable to a normal CDF (which can be absorbed through equivalent scaling of all of the parameters) yields a function that is practically identical to the logit function, but probit models are more tractable in some situations than logit models. (In a Bayesian setting in which normally distributed prior distributions are placed on the parameters, the relationship between the normal priors and the normal CDF link function means that a probit model can be computed using Gibbs sampling, while a logit model generally cannot.)

The complementary log-log function  $\log(-\log(1 - p))$  may also be used. This link function is asymmetric and will often produce different results from the probit and logit link functions.

The identity link is also sometimes used for binomial data to yield the linear probability model, but a drawback of this model is that the predicted probabilities can be greater than one or less than zero. In implementation it is possible to fix the nonsensical probabilities outside of [0,1], but interpreting the coefficients can be difficult. The model's primary merit is that near  $p = 0.5$  it is approximately a linear transformation of the probit and logit—econometricians sometimes call this the Harvard model.

The variance function for binomial data is given by:

$$\text{Var}(Y_i) = \tau\mu_i(1 - \mu_i), \quad (13.11)$$

where the dispersion parameter  $\tau$  is typically fixed at exactly one. When it is not, the resulting quasi-likelihood model often described as binomial with overdispersion or quasi-binomial.

### 13.5.4. Multinomial regression

The binomial case may be easily extended to allow for a multinomial distribution as the response (also, a Generalized Linear Model for counts, with a constrained total). There are two ways in which this is usually done:

#### 13.5.4.1 Ordered response

If the response variable is an ordinal measurement, then one may fit a model function of the form:

$$\begin{aligned} g(\mu_m) &= \eta_m = \beta_0 + X_1\beta_1 + \cdots + X_p\beta_p + \gamma_2 + \cdots + \gamma_m \\ &= \eta_1 + \gamma_2 + \cdots + \gamma_m, \end{aligned} \quad (13.11)$$

where  $\mu_m = P(Y \leq m)$ , for  $m > 2$ . Different links  $g$  lead to proportional odds models or ordered probit models.

#### 13.5.4.2 Unordered response

If the response variable is a nominal measurement, or the data do not satisfy the assumptions of an ordered model, one may fit a model of the following form:

$$g(\mu_m) = \eta_m = \beta_{m,0} + X_1\beta_{m,1} + \cdots + X_p\beta_{m,p}, \quad (13.12)$$

where  $\mu_m = P(Y = m|Y \in \{1, m\})$ , for  $m > 2$ . Different links  $g$  lead to multinomial logit or multinomial probit models. These are more general than the ordered response models, and more parameters are estimated.

### 13.5.5. Count data

Another example of generalized linear models includes Poisson regression which models count data using the Poisson distribution. The link is typically the logarithm, the canonical link. The variance function is proportional to the mean

$$\text{var}(Y_i) = \tau\mu_i, \quad (13.13)$$

where the dispersion parameter  $\tau$  is typically fixed at exactly one. When it is not, the resulting quasi-likelihood model is often described as Poisson with overdispersion or quasi-Poisson.

## 13.6. Extensions

### 13.6.1. Correlated or clustered data

The standard GLM assumes that the observations are uncorrelated. Extensions have been developed to allow for correlation between observations, as occurs for example in longitudinal studies and clustered designs:

- **Generalized estimating equations (GEEs)** allow for the correlation between observations without the use of an explicit probability model for the origin of the correlations, so there is no explicit likelihood. They are suitable when the random effects and their variances are not of inherent interest, as they allow for the correlation without explaining its origin. The focus is on estimating the average response over the population (“population-averaged” effects) rather than the regression parameters that would enable prediction of the effect of changing one or more components of  $X$  on a given individual. GEEs are usually used in conjunction with Huber-White standard errors (Zeger, Liang, & Albert, 1988) (Hardin

& Hilbe, 2003).

- **Generalized linear mixed models (GLMMs)** are an extension to GLMs that includes random effects in the linear predictor, giving an explicit probability model that explains the origin of the correlations. The resulting “subject-specific” parameter estimates are suitable when the focus is on estimating the effect of changing one or more components of  $X$  on a given individual. GLMMs are also referred to as multilevel models and as mixed model. In general, fitting GLMMs is more computationally complex and intensive than fitting GEEs.

### 13.6.2. Generalized additive models

**Generalized additive models (GAMs)** are another extension to GLMs in which the linear predictor  $\eta$  is not restricted to be linear in the covariates  $X$  but is the sum of smoothing functions applied to the  $x_i$ 's:

$$\eta = \beta_0 + f_1(x_1) + f_2(x_2) + \dots \quad (13.14)$$

The smoothing functions  $f_i$  are estimated from the data. In general this requires a large number of data points and is computationally intensive (Hastie & Tibshirani, 1990) (Wood, 2006).

The model relates a univariate response variable,  $Y$ , to some predictor variables,  $x_i$ . An exponential family distribution is specified for  $Y$  (for example normal, binomial or Poisson distributions) along with a link function  $g$  (for example the identity or log functions) relating the expected value of  $Y$  to the predictor variables via a structure such as

$$g(E(Y)) = \eta = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_m(x_m). \quad (13.15)$$

The functions  $f_i(x_i)$  may be functions with a specified parametric form (for example a polynomial, or a coefficient depending on the levels of a factor variable) or maybe specified non-parametrically, or semi-parametrically, simply as ‘smooth functions’, to be estimated by non-parametric means. So a typical GAM might use a scatterplot smoothing function, such as a locally weighted mean, for  $f_1(x_1)$ , and then use a factor model for  $f_2(x_2)$ . This flexibility to allow non-parametric fits with relaxed assumptions on the actual relationship between response and

predictor, provides the potential for better fits to data than purely parametric models, but arguably with some loss of interpretability.

### 13.6.3. Generalized additive model for location, scale and shape

The generalized additive model location, scale and shape (GAMLSS) is a class of statistical model that provides extended capabilities compared to the simpler generalized linear models and generalized additive models. These simpler models allow the typical values of a quantity being modelled to be related to whatever explanatory variables are available. Here the “typical value” is more formally a location parameter, which only describes a limited aspect of the probability distribution of the dependent variable. The GAMLSS approach allows other parameters of the distribution to be related to the explanatory variables; where these other parameters might be interpreted as scale and shape parameters of the distribution, although the approach is not limited to such parameters.

In GAMLSS the exponential family distribution assumption for the response variable, ( $Y$ ), (essential in GLMs and GAMs), is relaxed and replaced by a general distribution family, including highly skew and/or kurtotic continuous and discrete distributions.

The systematic part of the model is expanded to allow modeling not only of the mean (or location) but other parameters of the distribution of  $Y$  as linear and/or nonlinear, parametric and/or additive non-parametric functions of explanatory variables and/or random effects.

GAMLSS is especially suited for modeling leptokurtic or platykurtic and/or positive or negative skew response variable. For count type response variable data it deals with over-dispersion by using proper over-dispersed discrete distributions. Heterogeneity also is dealt with by modeling the scale or shape parameters using explanatory variables. There are several packages written in R related to GAMLSS models (Stasinopoulos & Rigby, 2007).

A GAMLSS model assumes independent observations  $y_i$  for  $i = 1, \dots, n$  with probability (density) function  $f(y_i | \mu_i, \sigma_i, \nu_i, \tau_i)$  conditional on  $(\mu_i, \sigma_i, \nu_i, \tau_i)$  a vector of four distribution parameters, each of which can

be a function to the explanatory variables. The first two population distribution parameters  $\mu_i$  and  $\sigma_i$  are usually characterized as location and scale parameters, while the remaining parameter(s), if any, are characterized as shape parameters, e.g. skewness and kurtosis parameters, although the model may be applied more generally to the parameters of any population distribution with up to four distribution parameters, and can be generalized to more than four distribution parameters (Stasinopoulos & Rigby, 2007).

$$g_1(\mu) = \eta_1 = X_1\beta_1 + \sum_{j=1}^{J_1} h_{j1}(x_{j1})$$

$$g_2(\mu) = \eta_2 = X_2\beta_2 + \sum_{j=1}^{J_2} h_{j2}(x_{j2})$$

$$g_3(\mu) = \eta_{23} = X_3\beta_3 + \sum_{j=1}^{J_3} h_{j3}(x_{j3})$$

$$g_4(\mu) = \eta_4 = X_4\beta_4 + \sum_{j=1}^{J_4} h_{j4}(x_{j4}).$$

(13.15a, b, c, d)

Where  $\mu, \sigma, \nu, \tau$  and  $n_k$  are vectors of length  $n$ ,  $\beta_k^T = (\beta_{1k}, \beta_{2k}, \dots, \beta_{J'_k k})$  is a parameter vector of length  $J'_k$ ,  $X_k$  is a fixed known design matrix of order  $n \times J'_k$  and  $h_{jk}$  is a smooth non-parametric function of explanatory variable  $x_{jk}, j = 1, \dots, j_k$  and  $k = 1, 2, 3, 4$  (Nelder & Wedderburn, 1972).

### 13.7. Confusion with general linear models

The term “**generalized linear model**”, and especially its abbreviation **GLM**, can be confused with **general linear model**. John Nelder has expressed regret about this in a conversation with Stephen Senn:

Senn: I must confess to having some confusion when I was a young statistician between general linear models and generalized linear models. Do you regret the terminology?

Nelder: I think probably I do. I suspect we should have found some more fancy name for it that would have stuck and not been confused with the general linear model, although general and generalized are not quite the same. I can see why it might have been better to have thought of something else (Senn, 2003).

## 13.8. Example Using R

### 13.8.1. Setting up the Model

#### EXAMPLE 1. Survival Model

Use data “rats” from “survival” package in R.

Data on 150 rats contain identifying “litter”, ““rx” (indicator of injection of drug after initial administration of carcinogen), time in days on study (ignored initially, and “status” which is indicator of tumor, our binary response variable.

```
> library(survival)

> fitB1 = glm(cbind(status,1-status) ~ rx, family=binomial,
+ data = rats)
```

NOTE you can use the column headers as variable names if you specify the data-frame using “data=“

```
> fitB1

Call: glm(formula = cbind(status, 1 - status) ~ rx, family =
  binomial,
  data = rats)

Coefficients:
(Intercept)          rx
-1.450           1.127

Degrees of Freedom: 149 Total (i.e. Null); 148 Residual
Null Deviance:    174
Residual Deviance: 165.3      AIC: 169.3
```

```

> summary(fitB1)$coef
            Estimate Std. Error   z value    Pr(>|z|)
(Intercept) -1.450010  0.2549063 -5.688404 1.282324e-08
rx           1.127237  0.3835089  2.939272 3.289845e-03

Coefficients fitted by MLE, link="logit" is default for
binomial-family data: this is logistic regression.
The standard error is found as sqrt of diagonal in variance:
> sqrt(diag(summary(fitB1)$cov.scaled))
(Intercept)          rx
0.2549063  0.3835089

"Deviance" = "Residual Deviance" = -2*logLik
> c(fitB1$deviance, -2*logLik(fitB1))
[1] 165.2738 165.2738

```

"Null.deviance" is -2 times the logLik for the same model with only a constant term

```

> c(fitB1>null.dev,
+ -2*logLik(update(fitB1, formula = .~ 1))
+ )
[1] 173.9746 173.9746

```

NOTE the use of the "update" function to refit a model of the same type as the one previously done: in the model formula, the left-hand side term is a "." to indicate that it is the same as before. We have changed the right-hand side from rx (which automatically included intercept along with rx predictor) to one with "1" or intercept alone.

Next use "update" to change the fitB1 model not in its model formula but in its specified link within the binomial "family".

```

> fitB2 = update(fitB1, family=binomial(link="probit"))
> rbind(logit= fitB1$coef, probit= fitB2$coef,
+ rescal.probit = fitB2$coef/0.5513)
            (Intercept)          rx
ogit      -1.4500102 1.1272368
probit     -0.8778963 0.6760028
rescal.probit -1.5924112 1.2261977

```

Recall that we use deviance and differences between them because Wilks' Theorem says that 2 times the difference between `logLik` for a model with  $p$  extra parameter dimensions versus `logLik` for a base model is equal to approximately a chi-square ( $p, df$ ) variate when the base model is the true one (see Chapter 10).

### 13.8.2. Model Quality

LET's use this idea to examine the quality of the model with predictor `log(time)` in the model along with `rx`.

NOTE that we must put the expression `log(time)` within `I()` to have it evaluated and constructed as a new predictor within the `glm` fitting function.

```
> fitB3 = update(fitB1, formula= . ~ . + I(log(time)))
```

The "data" is the same now as in `fitB1`, so "time" is the column in the data-frame, and a `log(time)` predictor is created under "glm"

It is reasonable to use "time" as a predictor, because the range of times is not too different for rats who generate tumors, so "time" is not really a response variable.

```
> summary(rats$time[rats$status==1])
   Min. 1st Qu. Median Mean 3rd Qu. Max.
 34.00  66.75  80.00  77.28  92.50 104.00
> summary(rats$time[rats$status==0])
   Min. 1st Qu. Median Mean 3rd Qu. Max.
 45.00  83.50 104.00  93.85 104.00 104.00

> cbind(rats[1:10,], model.matrix(fitB3)[1:10,])
   litter rx time status (Intercept) rx I(log(time))
1      1  1 101     0          1  1 4.615121
2      1  0  49     1          1  0 3.891820
3      1  0 104     0          1  0 4.644391
4      2  1 104     0          1  1 4.644391
5      2  0 102     0          1  0 4.624973
6      2  0 104     0          1  0 4.644391
7      3  1 104     0          1  1 4.644391
8      3  0 104     0          1  0 4.644391
```

|    |   |   |     |   |   |   |          |
|----|---|---|-----|---|---|---|----------|
| 9  | 3 | 0 | 104 | 0 | 1 | 0 | 4.644391 |
| 10 | 4 | 1 | 77  | 0 | 1 | 1 | 4.343805 |

The first 4 columns are the original “rats” data. The last 3 are the design matrix columns created by `glm`. So we can look at the new model fit for significance of coefficients:

```
> summary(fitB3)$coef
            Estimate Std. Error    z value    Pr(>|z|)
(Intercept) 17.868661  4.4924813 3.977459 6.965559e-05
rx           1.194019  0.4284626 2.786751 5.323935e-03
I(log(time)) -4.355407  1.0180848 -4.278040 1.885462e-05
```

Or alternatively compare deviances or `logLik`'s with `fitB1`

```
> c(2*(logLik(fitB3)-logLik(fitB1)), fitB1$dev-fitB3$dev)
[1] 24.37307 24.37307
```

This is LRT stat to be compared with `chisq` 1 df.

```
> 1-pchisq(24.373,1)
[1] 7.937339e-07
```

This is still highly significant but somewhat different p-value from the `I(log(time))` coef probably because the model is still far from the right one.

We could try to enter additional terms like `log(time)^2` or `rx * log(time)`

```
> fitB4 = update(fitB3, .~. + I(rx*log(time)) +
+ I(log(time)^2))
> summary(fitB4)$coef
            Estimate Std. Error    z value
(Intercept) -9.784968  52.690740 -0.1857056
rx          -13.501206   8.732837 -1.5460274
I(log(time)) 10.179730  24.364938  0.4178024
I(rx * log(time)) 3.324780   1.973389  1.6848069
```

```

I(log(time)^2)      -1.871215  2.819823 -0.6635932
                  Pr(>|z|)
(Intercept)        0.85267560
rx                 0.12209794
I(log(time))       0.67609159
I(rx * log(time)) 0.09202582
I(log(time)^2)     0.50695069

```

This time, the new variables do NOT look significant which we can check also through deviances:

```

> fitB3$dev - fitB4$dev
[1] 3.901065

```

This result is to be compared with  $\chi^2$  2df, so it is not at all significant.

we can also do these deviance comparisons all at once by looking at an “analysis of deviance” table

```

> anova(fitB4)
Analysis of Deviance Table
Model: binomial, link: logit

Response: cbind(status, 1 - status)
Terms added sequentially (first to last)


```

|                   | Df | Deviance | Resid. Df | Resid. Dev |
|-------------------|----|----------|-----------|------------|
| NULL              |    |          | 149       | 173.97     |
| rx                | 1  | 8.7008   | 148       | 165.27     |
| I(log(time))      | 1  | 24.3731  | 147       | 140.90     |
| I(rx * log(time)) | 1  | 3.4784   | 146       | 137.42     |
| I(log(time)^2)    | 1  | 0.4226   | 145       | 137.00     |

As in an ANOVA table, in which RSS replaced Deviance these “Deviance values” are the amounts by which the current model-fitting line decreases the deviance: (recall that `fitB1` uses `rx` only, `fitB3` augments by `log(time)`, and `fitB4` by `log(time)` plus the 2 additional `rx*log(time)` and `log(time)^2` terms.

```

> Devs = c(fitB1>null.dev, fitB1$dev, fitB3$dev,
+ update(fitB3, .~.+I(rx*log(time)))$dev,
+ fitB4$dev)
> Devs
[1] 173.9746 165.2738 140.9007 137.4223 136.9997

> round (-diff(Devs), 3 )### successive differences of 11ks
[1] 8.701 24.373 3.478 0.423      ### give Deviance col
in "anova"

```

## 13.9. Problems

1. Using the `anscombe` data set, perform four simple linear regressions with  $y_1, y_2, y_3$  and  $y_4$  as the dependent variable. This data set requires the `stats` package, i.e., `library(stats)`. Which regression model are not linear?
2. Using the `attitude` data set, build a linear model of overall ratings ( $Y$ ) based on complaints, privileges, learning, raises, critical and advance. Hint: `fit1 <- lm(rating ~ ., data = attitude)`
  - a. Which independent variables are significant in explaining overall ratings (Hint: check p-values)?
  - b. Use the strongest predictor to build another regression model of overall ratings. Use a few diagnostic plots (e.g., residuals, Normal Q-Q) to evaluate the linear fit.
3. A macroeconomic data set, `Longley`, contains factors contributing to the explanation of employment in the US. Construct a linear regression model.
  - a. Build a regression model and determine which factors have the most explanatory power.
  - b. Report values for adjusted R-squared, residual standard error and F-statistic.
  - c. Use various lots to diagnose the fit of your model.

# 14.What is Logistic Regression?

In statistics, **logistic regression**, or **logit regression**, is a type of probabilistic statistical classification model (Bishop, 2006). It is also used to predict a binary response from a binary predictor, used for predicting the outcome of a categorical dependent variable (i.e., a class label) based on one or more predictor variables (features). That is, it is used in estimating the parameters of a qualitative response model. The probabilities describing the possible outcomes of a single trial are modeled, as a function of the explanatory (predictor) variables, using a logistic function. Frequently (and subsequently in this article) “logistic regression” is used to refer specifically to the problem in which the dependent variable is binary—that is, the number of available categories is two—while problems with more than two categories are referred to as multinomial logistic regression or, if the multiple categories are ordered, as ordered logistic regression.

Logistic regression measures the relationship between a **categorical dependent variable** and one or more independent variables, which are usually (but not necessarily) continuous, by using probability scores as the predicted values of the dependent variable (Bhandari & Joensson, 2008). As such it treats the same set of problems as does probit regression using similar techniques.

## 14.1. Fields and examples of applications

Logistic regression was put forth in the 1940s as an alternative to Fisher’s 1936 classification method, **linear discriminant analysis** (James, Witten, Hastie, & Tibshirani, 2013). It is used extensively in numerous disciplines, including the medical and social science fields. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by Boyd et al. using logistic regression (Boyd, Tolson, & Copes, 1987). Logistic regression might be used to predict whether a patient has a given disease (e.g. diabetes), based on observed characteristics of the patient (age, gender, body mass index, results of various blood tests, etc.).

Another example, a **propensity model**, might be to predict whether an American voter will vote Democratic or Republican, based on age, income, gender, race, state of residence, votes in previous elections, etc. (Harrell, 2010). The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product (Strano & Colosimo, 2006) (Palei & Das, 2009). It is also used in marketing applications such as prediction of a customer's propensity to purchase a product or cease a subscription, etc. In economics it can be used to predict the likelihood of a person's choosing to be in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

## 14.2. Basics

Logistic regression can be **binomial** or **multinomial**. Binomial or binary logistic regression deals with situations in which the observed outcome for a dependent variable can have only two possible types (for example, "dead" vs. "alive"). Multinomial logistic regression deals with situations where the outcome can have three or more possible types (e.g., "disease A" vs. "disease B" vs. "disease C"). In binary logistic regression, the outcome is usually coded as "0" or "1", as this leads to the most straightforward interpretation (Hosmer & Lemeshow, 2000). If a particular observed outcome for the dependent variable is the noteworthy possible outcome (referred to as a "success" or a "case") it is usually coded as "1" and the contrary outcome (referred to as a "failure" or a "noncase") as "0". Logistic regression is used to predict the odds of being a case based on the values of the independent variables (predictors). The odds are defined as the probability that a particular outcome is a case divided by the probability that it is a noncase.

Like other forms of regression analysis, logistic regression makes use of one or more predictor variables that may be either continuous or categorical data. Unlike ordinary linear regression, however, logistic regression is used for predicting binary outcomes of the dependent variable (treating the dependent variable as the outcome of a Bernoulli trial) rather than continuous outcomes. Given this difference, it is

necessary that logistic regression take the natural logarithm of the odds of the dependent variable being a case (referred to as the logit or log-odds) to create a continuous criterion as a transformed version of the dependent variable. Thus the **logit transformation** is referred to as the **link function** in logistic regression—although the dependent variable in logistic regression is binomial, the logit is the continuous criterion upon which linear regression is conducted (Hosmer & Lemeshow, 2000).

The logit of success is then fit to the predictors using linear regression analysis. The predicted value of the logit is converted back into predicted odds via the inverse of the natural logarithm, namely the exponential function. Therefore, although the observed dependent variable in logistic regression is a zero-or-one variable, the logistic regression estimates the odds, as a continuous variable, that the dependent variable is a success (a case). In some applications the odds are all that is needed. In others, a specific yes-or-no prediction is needed for whether the dependent variable is or is not a case; this categorical prediction can be based on the computed odds of a success, with predicted odds above some chosen cut-off value being translated into a prediction of a success.

### 14.3. Logistic function, odds ratio, and logit

An explanation of logistic regression begins with an explanation of the logistic function, which always takes on values between zero and one (Hosmer & Lemeshow, 2000):

$$F(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}, \quad (14.1)$$

and viewing  $t$  as a linear function of an explanatory variable  $x$  (or of a linear combination of explanatory variables), the logistic function can be written as:

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}. \quad (14.2)$$

This will be interpreted as the probability of the dependent variable equaling a “success” or “case” rather than a failure or non-case. We also define the inverse of the logistic function, the logit:

$$g(x) = \ln \frac{F(x)}{1 - F(x)} = \beta_0 + \beta_1 x, \quad (14.3)$$

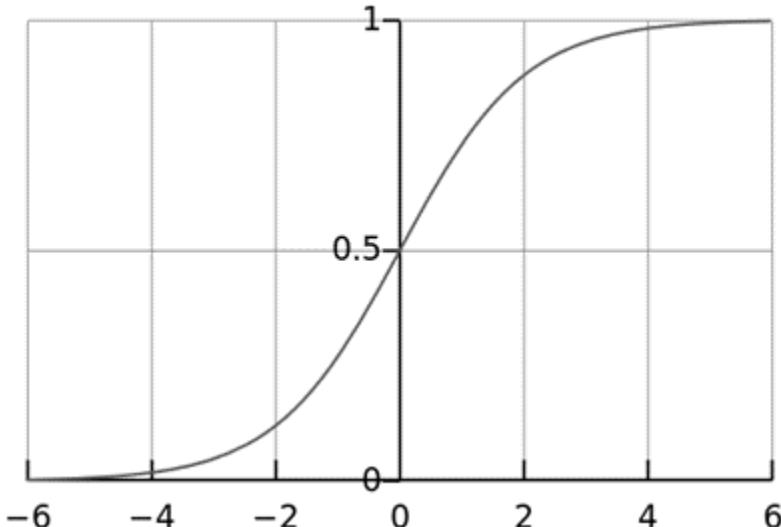
and equivalently:

$$\frac{F(x)}{1 - F(x)} = e^{(\beta_0 + \beta_1 x)}. \quad (14.4)$$

A graph of the logistic function is shown in Figure 1. The input is the value of and the output is  $F(x)$ . The logistic function is useful because it can take an input with any value from negative infinity to positive infinity, whereas the output is confined to values between 0 and 1 and hence is interpretable as a probability. In the above equations, refers to the logit function of some given linear combination of the predictors, denotes the natural logarithm, is the probability that the dependent variable equals a case, is the intercept from the linear regression equation (the value of the criterion when the predictor is equal to zero), is the regression coefficient multiplied by some value of the predictor, and base e denotes the exponential function.

The formula for  $F(x)$  illustrates that the probability of the dependent variable equaling a case is equal to the value of the logistic function of the linear regression expression. This is important in that it shows that the value of the linear regression expression can vary from negative to positive infinity and yet, after transformation, the resulting expression for the probability  $F(x)$  ranges between 0 and 1. The equation for  $g(x)$  illustrates that the logit (i.e., log-odds or natural logarithm of the odds) is equivalent to the linear regression expression. Likewise, the next equation illustrates that the odds of the dependent variable equaling a case is equivalent to the exponential function of the linear regression expression. This illustrates how the logit serves as a link function between the probability and the linear regression expression. Given that

the logit ranges between negative infinity and positive infinity, it provides an adequate criterion upon which to conduct linear regression and the logit is easily converted back into the odds (Hosmer & Lemeshow, 2000)



**Figure 1.** The logistic function, with  $\beta_0 + \beta_1$  on the horizontal axis and  $F(x)$  on the vertical axis

#### 14.3.1. Multiple explanatory variables

If there are multiple explanatory variables, then the above expression  $\beta_0 + \beta_1 x$  can be revised to  $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m$ . Then when this is used in the equation relating the logged odds of a success to the values of the predictors, the linear regression will be a multiple regression with  $m$  explanators; the parameters  $\beta_j$  for all  $j = 0, 1, 2, \dots, m$  are all estimated.

### 14.4. Model fitting

#### 14.4.1. Estimation

#### 14.4.1.1 Maximum likelihood estimation

The regression coefficients are usually estimated using **maximum likelihood estimation** (Menard, 2002). Unlike linear regression with normally distributed residuals, it is not possible to find a closed-form expression for the coefficient values that maximizes the likelihood function, so an iterative process must be used instead, for example Newton's method. This process begins with a tentative solution, revises it slightly to see if it can be improved, and repeats this revision until improvement is minute, at which point the process is said to have converged (Menard, 2002).

In some instances the model may not reach convergence. When a model does not converge this indicates that the coefficients are not meaningful because the iterative process was unable to find appropriate solutions. A failure to converge may occur for a number of reasons: having a large proportion of predictors to cases, multicollinearity, sparseness, or complete separation.

- Having a large proportion of variables to cases results in an overly conservative Wald statistic (discussed below) and can lead to nonconvergence.
- Multicollinearity refers to unacceptably high correlations between predictors. As multicollinearity increases, coefficients remain unbiased but standard errors increase and the likelihood of model convergence decreases. To detect multicollinearity amongst the predictors, one can conduct a linear regression analysis with the predictors of interest for the sole purpose of examining the tolerance statistic used to assess whether multicollinearity is unacceptably high (Menard, 2002).
- Sparseness in the data refers to having a large proportion of empty cells (cells with zero counts). Zero cell counts are particularly problematic with categorical predictors. With continuous predictors, the model can infer values for the zero cell counts, but this is not the case with categorical predictors. The reason the model will not converge with zero cell counts for categorical predictors is because the natural logarithm of zero is an undefined value, so final

solutions to the model cannot be reached. To remedy this problem, researchers may collapse categories in a theoretically meaningful way or may consider adding a constant to all cells (Menard, 2002).

- Another numerical problem that may lead to a lack of convergence is complete separation, which refers to the instance in which the predictors perfectly predict the criterion – all cases are accurately classified. In such instances, one should reexamine the data, as there is likely some kind of error (Hosmer & Lemeshow, 2000).

Although not a precise number, as a general rule of thumb, logistic regression models require a minimum of 10 events per explaining variable (where event denotes the cases belonging to the less frequent category in the dependent variable) (Peduzzi, Concato, Kemper, Holford, & Feinstein, 1996).

#### 14.4.1.2 Minimum chi-squared estimator for grouped data

While individual data will have a dependent variable with a value of zero or one for every observation, with grouped data one observation is on a group of people who all share the same characteristics (e.g., demographic characteristics); in this case the researcher observes the proportion of people in the group for whom the response variable falls into one category or the other. If this proportion is neither zero nor one for any group, the minimum chi-squared estimator involves using weighted least squares to estimate a linear model in which the dependent variable is the logit of the proportion: that is, the log of the ratio of the fraction in one group to the fraction in the other group (Greene, 2011).

#### 14.4.2. Evaluating goodness-of-fit

Goodness-of-fit in linear regression models is generally measured using the  $R^2$ . Since this has no direct analog in logistic regression, various methods (Greene, 2011) including the following can be used instead.

##### 14.4.2.1 Deviance and likelihood ratio tests

In linear regression analysis, one is concerned with partitioning variance via the sum of squares calculations – variance in the criterion is essentially divided into variance accounted for by the predictors and

residual variance. In logistic regression analysis, **deviance** is used in lieu of sum of squares calculations (Cohen, Cohen, West, & Aiken, 2002). Deviance is analogous to the sum of squares calculations in linear regression (Hosmer & Lemeshow, 2000) and is a measure of the lack of fit to the data in a logistic regression model (Cohen, Cohen, West, & Aiken, 2002). Deviance is calculated by comparing a given model with the saturated model—a model with a theoretically perfect fit. This computation is called the **likelihood-ratio test** (Hosmer & Lemeshow, 2000):

$$D = -2\ln\lambda(y_i) = -2 \ln \frac{\text{likelihood of the fitted model}}{\text{likelihood of the saturated model}}$$

In the above equation  $D$  represents the deviance and  $\ln$  represents the natural logarithm, and  $\Lambda = \lambda(y_i)$ . The log of the likelihood ratio (the ratio of the fitted model to the saturated model) will produce a negative value, so the product is multiplied by negative two times its natural logarithm to produce a value with an approximate  $\chi^2$ -squared distribution (Hosmer & Lemeshow, 2000). Smaller values indicate better fit as the fitted model deviates less from the saturated model. When assessed upon a chi-square distribution, nonsignificant chi-square values indicate very little unexplained variance and thus, good model fit. Conversely, a significant chi-square value indicates that a significant amount of the variance is unexplained.

Two measures of deviance are particularly important in logistic regression: **null deviance** and **model deviance**. The null deviance represents the difference between a model with only the intercept (which means “no predictors”) and the saturated model (Cohen, Cohen, West, & Aiken, 2002). And, the model deviance represents the difference between a model with at least one predictor and the saturated model. In this respect, the null model provides a baseline upon which to compare predictor models. Given that deviance is a measure of the difference between a given model and the saturated model, smaller values indicate better fit. Therefore, to assess the contribution of a predictor or set of predictors, one can subtract the model deviance from the null deviance and assess the difference on a  $\chi^2_{S-p}$  chi-square

distribution with degree of freedom (Hosmer & Lemeshow, 2000) equal to the difference in the number of parameters estimated.

Let

$$D_{\text{null}} = -2 \ln \frac{\text{likelihood of the null model}}{\text{likelihood of the saturated model}}$$

$$D_{\text{fitted}} = -2 \ln \frac{\text{likelihood of the fitted model}}{\text{likelihood of the saturated model}}$$

Then

$$\begin{aligned} & D_{\text{fitted}} - D_{\text{null}} \\ &= \left( -2 \ln \frac{\text{likelihood of the fitted model}}{\text{likelihood of the saturated model}} \right) \\ &\quad - \left( 2 \ln \frac{\text{likelihood of the null model}}{\text{likelihood of the saturated model}} \right) \\ &= -2 \left( \ln \frac{\text{likelihood of the fitted model}}{\text{likelihood of the saturated model}} \right. \\ &\quad \left. - \ln \frac{\text{likelihood of the null model}}{\text{likelihood of the saturated model}} \right) \\ &= -2 \ln \frac{\text{likelihood of the saturated model}}{\text{likelihood of the null model}} \\ &\quad \frac{\text{likelihood of the saturated model}}{\text{likelihood of the fitted model}} \\ &= -2 \ln \frac{\text{likelihood of the fitted model}}{\text{likelihood of the null model}}. \end{aligned}$$

If the model deviance is significantly smaller than the null deviance then one can conclude that the predictor or set of predictors significantly improved model fit. This is analogous to the  $F$ -test used in linear regression analysis to assess the significance of prediction (Cohen, Cohen, West, & Aiken, 2002). A convenient result, attributed to Samuel S. Wilks, says that as the sample size  $n$  approaches  $\infty$ , the test statistic  $-2\ln(\Lambda)$  for a nested model will be asymptotically  $\chi^2$ -distributed with degrees of freedom equal to the difference in dimensionality of the saturated model and null model (Wilks, 1938). This means that for a great variety of hypotheses, a practitioner can compute the likelihood ratio  $\Lambda$  for the data and compare  $-2\ln(\Lambda)$  to the  $\chi^2$  value corresponding to a desired statistical significance as an approximate statistical test. This is often referred to a Wilks' Theorem.

## 14.4.2.2

Pseudo- $R^2$ s

In linear regression the squared multiple correlation,  $R^2$  is used to assess goodness-of-fit as it represents the proportion of variance in the criterion that is explained by the predictors (Cohen, Cohen, West, & Aiken, 2002). In logistic regression analysis, there is no agreed upon analogous measure, but there are several competing measures each with limitations. Three of the most commonly used indices are examined on this section beginning with the **likelihood ratio  $R^2$** ,  $R_L^2$  (Cohen, Cohen, West, & Aiken, 2002):

$$R_L^2 \frac{D_{null} - D_{model}}{D_{null}}.$$
(14.5)

This is the most analogous index to the squared multiple correlation in linear regression (Menard, 2002). It represents the proportional reduction in the deviance, wherein the deviance is treated as a measure of variation analogous but not identical to the variance in linear regression analysis (Menard, 2002). One limitation of the likelihood ratio  $R^2$  is that it is not monotonically related to the odds ratio (Cohen, Cohen, West, & Aiken, 2002), meaning that it does not necessarily increase as the odds ratio increases, and does not necessarily decrease as the odds ratio decreases.

The **Cox and Snell  $R^2$**  is an alternative index of goodness-of-fit related to the  $R^2$  value from linear regression. The Cox and Snell index is problematic as its maximum value is 0.75, when the variance is at its maximum (0.25). The **Nagelkerke  $R^2$**  provides a correction to the Cox and Snell  $R^2$  so that the maximum value is equal to one. Nevertheless, the Cox and Snell and likelihood ratio  $R^2$ 's show greater agreement with each other than either does with the Nagelkerke  $R^2$  (Cohen, Cohen, West, & Aiken, 2002). Of course, this might not be the case for values exceeding 0.75 as the Cox and Snell index is capped at this value. The likelihood ratio  $R^2$  is often preferred to the alternatives as it is most analogous to  $R^2$  in linear regression, is independent of the base rate (both Cox and Snell and Nagelkerke  $R^2$ 's increase as the proportion of cases increase from 0 to 0.5) and varies between 0 and 1.

A word of caution is in order when interpreting pseudo- $R^2$  statistics. The reason these indices of fit are referred to as pseudo  $R^2$  is because they do not represent the proportionate reduction in error as the  $R^2$  in linear regression does (Cohen, Cohen, West, & Aiken, 2002). Linear regression assumes homoscedasticity, that the error variance is the same for all values of the criterion. Logistic regression will always be heteroscedastic – the error variances differ for each value of the predicted score. For each value of the predicted score there would be a different value of the proportionate reduction in error. Therefore, it is inappropriate to think of  $R^2$  as a proportionate reduction in error in a universal sense in logistic regression (Cohen, Cohen, West, & Aiken, 2002).

#### 14.4.2.3 Hosmer–Lemeshow test

The **Hosmer–Lemeshow test** uses a test statistic that asymptotically follows a  $\chi^2$  distribution to assess whether or not the observed event rates match expected event rates in subgroups of the model population (Hosmer & Lemeshow, 2000).

#### 14.4.2.4 Evaluating binary classification performance

If the estimated probabilities are to be used to classify each observation of independent variable values as predicting the category that the dependent variable is found in, the various methods below for judging the model's suitability in out-of-sample forecasting can also be used on the data that were used for estimation—accuracy, precision (also called positive predictive value), recall (also called sensitivity), specificity and negative predictive value. In each of these evaluative methods, an aspect of the model's effectiveness in assigning instances to the correct categories is measured.

### 14.5. Coefficients

After fitting the model, it is likely that researchers will want to examine the contribution of individual predictors. To do so, they will want to examine the regression coefficients. In linear regression, the regression coefficients represent the change in the criterion for each unit change in the predictor (Cohen, Cohen, West, & Aiken, 2002). In logistic regression, however, the regression coefficients represent the change in the logit for each unit change in the predictor. Given that the logit is not intuitive,

researchers are likely to focus on a predictor's effect on the exponential function of the regression coefficient – the odds ratio (see definition). In linear regression, the significance of a regression coefficient is assessed by computing a *t*-test. In logistic regression, there are several different tests designed to assess the significance of an individual predictor, most notably the likelihood ratio test and the Wald statistic.

#### 14.5.1. Likelihood ratio test

The **likelihood-ratio test** discussed above to assess model fit is also the recommended procedure to assess the contribution of individual “predictors” to a given model (Cohen, Cohen, West, & Aiken, 2002) (Hosmer & Lemeshow, 2000) (Menard, 2002). In the case of a single predictor model, one simply compares the deviance of the predictor model with that of the null model on a chi-square distribution with a single degree of freedom. If the predictor model has a significantly smaller deviance (c.f chi-square using the difference in degrees of freedom of the two models), then one can conclude that there is a significant association between the “predictor” and the outcome. Although some common statistical packages (e.g. SPSS) do provide likelihood ratio test statistics, without this computationally intensive test it would be more difficult to assess the contribution of individual predictors in the multiple logistic regression case. To assess the contribution of individual predictors one can enter the predictors hierarchically, comparing each new model with the previous to determine the contribution of each predictor (Cohen, Cohen, West, & Aiken, 2002). (There is considerable debate among statisticians regarding the appropriateness of so-called “stepwise” procedures. They do not preserve the nominal statistical properties and can be very misleading (Harrell, 2010).

#### 14.5.2. Wald statistic

Alternatively, when assessing the contribution of individual predictors in a given model, one may examine the significance of the Wald statistic. The **Wald statistic**, analogous to the *t*-test in linear regression, is used to assess the significance of coefficients. The Wald statistic is the ratio of the square of the regression coefficient to the square of the standard

error of the coefficient and is asymptotically distributed as a chi-square distribution (Menard, 2002).

$$W_j = \frac{B_j^2}{SE_{B_j}^2}. \quad (14.6)$$

Although several statistical packages (e.g., SPSS, SAS) report the Wald statistic to assess the contribution of individual predictors, the Wald statistic has limitations. When the regression coefficient is large, the standard error of the regression coefficient also tends to be large increasing the probability of Type-II error. The Wald statistic also tends to be biased when data are sparse (Cohen, Cohen, West, & Aiken, 2002).

#### 14.5.3. Case-control sampling

Suppose cases are rare. Then we might wish to sample them more frequently than their prevalence in the population. For example, suppose there is a disease that affects 1 person in 10,000 and to collect our data we need to do a complete physical. It may be too expensive to do thousands of physicals of healthy people in order to get data on only a few diseased individuals. Thus, we may evaluate more diseased individuals. This is also called unbalanced data. As a rule of thumb, sampling controls at a rate of five times the number of cases is sufficient to get enough control data (Prentice & Pyke, 1979).

If we form a logistic model from such data, if the model is correct, the  $\beta_j$  parameters are all correct except for  $\beta_0$ . We can correct  $\beta_0$  if we know the true prevalence as follows (Prentice & Pyke, 1979):

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log \frac{\pi}{1 - \pi} \log \frac{\hat{\pi}}{1 - \hat{\pi}}, \quad (14.7)$$

where  $\pi$  is the true prevalence and  $\hat{\pi}$  is the prevalence in the sample.

### 14.6. Formal mathematical specification

There are various equivalent specifications of logistic regression, which fit into different types of more general models. These different specifications allow for different sorts of useful generalizations.

### 14.6.1. Setup

The basic setup of logistic regression is the same as for standard linear regression.

It is assumed that we have a series of  $N$  observed data points. Each data point  $i$  consists of a set of  $m$  explanatory variables  $x_{1,i}, \dots, x_{m,i}$  (also called independent variables, predictor variables, input variables, features, or attributes), and an associated binary-valued outcome variable  $Y_i$  (also known as a dependent variable, response variable, output variable, outcome variable or class variable), i.e. it can assume only the two possible values 0 (often meaning “no” or “failure”) or 1 (often meaning “yes” or “success”). The goal of logistic regression is to explain the relationship between the explanatory variables and the outcome, so that an outcome can be predicted for a new set of explanatory variables.

Some examples:

- The observed outcomes are the presence or absence of a given disease (e.g. diabetes) in a set of patients, and the explanatory variables might be characteristics of the patients thought to be pertinent (sex, race, age, blood pressure, body-mass index, etc.).
- The observed outcomes are the votes (e.g. Democratic or Republican) of a set of people in an election, and the explanatory variables are the demographic characteristics of each person (e.g. sex, race, age, income, etc.). In such a case, one of the two outcomes is arbitrarily coded as 1, and the other as 0.

As in linear regression, the outcome variables  $Y$  are assumed to depend on the explanatory variables  $x_{1,i}, \dots, x_{m,i}$ .

#### 14.6.1.1 Explanatory variables

As shown above in the above examples, the explanatory variables may be of any type: real-valued, binary, categorical, etc. The main distinction

is between continuous variables (such as income, age and blood pressure) and discrete variables (such as sex or race). Discrete variables referring to more than two possible choices are typically coded using dummy variables (or indicator variables), that is, separate explanatory variables taking the value 0 or 1 are created for each possible value of the discrete variable, with a 1 meaning “variable does have the given value” and a 0 meaning “variable does not have that value”. For example, a four-way discrete variable of blood type with the possible values “A, B, AB, O” can be converted to four separate two-way dummy variables, “is-A, is-B, is-AB, is-O”, where only one of them has the value 1 and all the rest have the value 0. This allows for separate regression coefficients to be matched for each possible value of the discrete variable. (In a case like this, only three of the four dummy variables are independent of each other, in the sense that once the values of three of the variables are known, the fourth is automatically determined. Thus, it is only necessary to encode three of the four possibilities as dummy variables. This also means that when all four possibilities are encoded, the overall model is not identifiable in the absence of additional constraints such as a regularization constraint. Theoretically, this could cause problems, but in reality almost all logistic regression models are fit with regularization constraints.)

#### 14.6.1.2              Outcome variables

Formally, the outcomes  $Y_i$  are described as being Bernoulli-distributed data, where each outcome is determined by an unobserved probability  $p_i$  that is specific to the outcome at hand, but related to the explanatory variables. This can be expressed in any of the following equivalent forms:

$$\begin{aligned}
 Y_i | x_{1,i}, \dots, x_{m,i} &\sim \text{Bernoulli}(p_i) \\
 E[Y_i | x_{1,i}, \dots, x_{m,i}] &= p_i \\
 \Pr(Y_i = y_i | x_{1,i}, \dots, x_{m,i}) &= \begin{cases} p_i & \text{if } y_i = 1 \\ 1 - p_i & \text{if } y_i = 0 \end{cases} \\
 \Pr(Y_i = y_i | x_{1,i}, \dots, x_{m,i}) &= p_i^{y_i} (1 - p_i)^{(1-y_i)}
 \end{aligned} \tag{14.8}$$

The meanings of these four lines are:

1. The first line expresses the probability distribution of each  $Y_i$ : Conditioned on the explanatory variables, it follows a Bernoulli distribution with parameters  $p_i$ , the probability of the outcome of 1 for trial  $i$ . As noted above, each separate trial has its own probability of success, just as each trial has its own explanatory variables. The probability of success  $p_i$  is not observed, only the outcome of an individual Bernoulli trial using that probability.
2. The second line expresses the fact that the expected value of each  $Y_i$  is equal to the probability of success  $p_i$ , which is a general property of the Bernoulli distribution. In other words, if we run a large number of Bernoulli trials using the same probability of success  $p_i$ , then take the average of all the 1 and 0 outcomes, then the result would be close to  $p_i$ . This is because doing an average this way simply computes the proportion of successes seen, which we expect to converge to the underlying probability of success.
3. The third line writes out the probability mass function of the Bernoulli distribution, specifying the probability of seeing each of the two possible outcomes.
4. The fourth line is another way of writing the probability mass function, which avoids having to write separate cases and is more convenient for certain types of calculations. This relies on the fact that  $Y_i$  can take only the value 0 or 1. In each case, one of the exponents will be 1, “choosing” the value under it, while the other is 0, “canceling out” the value under it. Hence, the outcome is either  $p_i$  or  $1 - p_i$ , as in the previous line.

#### 14.6.1.3 Linear predictor function

The basic idea of logistic regression is to use the mechanism already developed for linear regression by modeling the probability  $p_i$  using a linear predictor function, i.e. a linear combination of the explanatory variables and a set of regression coefficients that are specific to the model at hand but the same for all trials. The linear predictor function  $f(i)$  for a particular data point  $i$  is written as:

$$f(i) = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_m x_{m,i}, \quad (14.9)$$

where  $\beta_0, \dots, \beta_m$  are regression coefficients indicating the relative effect of a particular explanatory variable on the outcome.

The model is usually put into a more compact form as follows:

- The regression coefficients  $\beta_0, \beta_1, \dots, \beta_m$  are grouped into a single vector  $\boldsymbol{\beta}$  of size  $m + 1$ .
- For each data point  $i$ , an additional explanatory pseudo-variable  $x_{0,i}$  is added, with a fixed value of 1, corresponding to the intercept coefficient  $\beta_0$ .
- The resulting explanatory variables  $x_{0,i}, x_{1,i}, \dots, x_{m,i}$  are then grouped into a single vector  $\mathbf{X}_i$  of size  $m + 1$ .

This makes it possible to write the linear predictor function as follows:

$$\mathbf{f}(i) = \boldsymbol{\beta} \cdot \mathbf{X}_i, \quad (14.10)$$

using the notation for a dot product between two vectors.

#### 14.6.2. As a generalized linear model

The particular model used by logistic regression, which distinguishes it from standard linear regression and from other types of regression analysis used for binary-valued outcomes, is the way the probability of a particular outcome is linked to the linear predictor function:

$$\begin{aligned} \text{logit}(E[Y_i | x_{1,i}, \dots, x_{m,i}]) &= \text{logit}(p_i) \\ &= \ln\left(\frac{p_i}{1 - p_i}\right) = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_m x_{m,i} \end{aligned}$$

Written using the more compact notation described above, this is:

$$\text{logit}(E[Y_i | \mathbf{X}_i]) = \text{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = \boldsymbol{\beta} \cdot \mathbf{X}_i. \quad (14.11)$$

This formulation expresses logistic regression as a type of generalized linear model, which predicts variables with various types of probability distributions by fitting a linear predictor function of the above form to some sort of arbitrary transformation of the expected value of the variable.

The intuition for transforming using the logit function (the natural log of the odds) was explained above. It also has the practical effect of converting the probability (which is bounded to be between 0 and 1) to a variable that ranges over  $(-\infty, +\infty)$  — thereby matching the potential range of the linear prediction function on the right side of the equation.

Note that both the probabilities  $p_i$  and the regression coefficients are unobserved, and the means of determining them is not part of the model itself. They are typically determined by some sort of optimization procedure, e.g. maximum likelihood estimation, which finds values that best fit the observed data (i.e. that give the most accurate predictions for the data already observed), usually subject to regularization conditions that seek to exclude unlikely values, e.g. extremely large values for any of the regression coefficients. The use of a regularization condition is equivalent to doing maximum a posteriori (MAP) estimation, an extension of maximum likelihood. (Regularization is most commonly done using a squared regularizing function, which is equivalent to placing a zero-mean Gaussian prior distribution on the coefficients, but other regularizers are also possible.) Whether or not regularization is used, it is usually not possible to find a closed-form solution; instead, an iterative numerical method must be used, such as iteratively reweighted least squares (IRLS) or, more commonly these days, a quasi-Newton method such as the L-BFGS method.

The interpretation of the  $\beta_j$  parameter estimates is as the additive effect on the log of the odds for a unit change in the  $j$ th explanatory variable. In the case of a dichotomous explanatory variable, for instance gender,  $e^\beta$  is the estimate of the odds of having the outcome for, say, males compared with females.

An equivalent formula uses the inverse of the logit function, which is the logistic function, i.e.:

$$E[Y_i | \mathbf{X}_i] = p_i = (\boldsymbol{\beta} \cdot \mathbf{X}_i) = \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{X}_i}}. \quad (14.12)$$

The formula can also be written (somewhat awkwardly) as a probability distribution (specifically, using a probability mass function):

$$\begin{aligned}\Pr(Y_i = y_i | \mathbf{X}_i) &= p_i^{y_i} (1 - p_i)^{(1-y_i)} \\ &= \left( \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{X}_i}} \right)^{y_i} \left( 1 - \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{X}_i}} \right)^{1-y_i}.\end{aligned}$$

### 14.6.3. As a latent-variable model

The above model has an equivalent formulation as a latent-variable model. This formulation is common in the theory of discrete choice models, and makes it easier to extend to certain more complicated models with multiple, correlated choices, as well as to compare logistic regression to the closely related probit model.

Imagine that, for each trial  $i$ , there is a continuous latent variable  $Y_i^*$  (i.e. an unobserved random variable) that is distributed as follows:

$$Y_i^* = \boldsymbol{\beta} \cdot \mathbf{X}_i + \varepsilon, \quad (14.13)$$

Where

$$\varepsilon \sim \text{Logistic}(0,1),$$

i.e., the latent variable can be written directly in terms of the linear predictor function and an additive random error variable that is distributed according to a standard logistic distribution.

Then  $Y_i$  can be viewed as an indicator for whether this latent variable is positive:

$$Y_i = \begin{cases} 1 & \text{if } Y_i^* > 0, \text{ i.e. } -\varepsilon < \boldsymbol{\beta} \cdot \mathbf{X}_i \\ 0 & \text{otherwise} \end{cases} \quad (14.14)$$

The choice of modeling the error variable specifically with a standard logistic distribution, rather than a general logistic distribution with the location and scale set to arbitrary values, seems restrictive, but in fact it is not. It must be kept in mind that we can choose the regression coefficients ourselves, and very often can use them to offset changes in the parameters of the error variable's distribution. For example, a logistic error-variable distribution with a non-zero location parameter  $\mu$  (which sets the mean) is equivalent to a distribution with a zero location

parameter, where  $\mu$  has been added to the intercept coefficient. Both situations produce the same value for  $Y_i^*$  regardless of settings of explanatory variables. Similarly, an arbitrary scale parameter  $s$  is equivalent to setting the scale parameter to 1 and then dividing all regression coefficients by  $s$ . In the latter case, the resulting value of  $Y_i^*$  will be smaller by a factor of  $s$  than in the former case, for all sets of explanatory variables — but critically, it will always remain on the same side of 0, and hence lead to the same  $Y_i$  choice.

(Note that this predicts that the irrelevancy of the scale parameter may not carry over into more complex models where more than two choices are available.)

It turns out that this formulation is exactly equivalent to the preceding one, phrased in terms of the generalized linear model and without any latent variables. This can be shown as follows, using the fact that the cumulative distribution function (CDF) of the standard logistic distribution is the logistic function, which is the inverse of the logit function, i.e.

$$\Pr(\varepsilon < x) = (x).$$

Then: (14.15)

$$\begin{aligned}\Pr(Y_i = 1 | \mathbf{X}_i) &= \Pr((Y_i^* > 0) | \mathbf{X}_i) \\ &= \Pr(\boldsymbol{\beta} \cdot \mathbf{X}_i + \varepsilon > 0) \\ &= \Pr(\varepsilon > -\boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= \Pr(\varepsilon < \boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= (\boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= p_i.\end{aligned}$$

This formulation — which is standard in discrete choice models — makes clear the relationship between logistic regression (the “logit model”) and the probit model, which uses an error variable distributed according to a standard normal distribution instead of a standard logistic distribution. Both the logistic and normal distributions are symmetric with a basic unimodal, “bell curve” shape. The only difference is that the logistic distribution has somewhat heavier tails, which means that it is less sensitive to outlying data (and hence somewhat more robust to model misspecifications or erroneous data).

#### 14.6.4. As a two-way latent-variable model

Yet another formulation uses two separate latent variables:

$$Y_i^{0*} = \beta_0 \cdot X_i + \varepsilon_0,$$

$$Y_i^{1*} = \beta_{01} \cdot X_i + \varepsilon_1,$$

Where

(14.16)

$$\varepsilon_0 \sim EV_1(0,1),$$

$$\varepsilon_1 \sim EV_1(0,1),$$

where  $EV_1(0,1)$  is a standard type-1 extreme value distribution: i.e.

$$Pr(\varepsilon_0 = x) = Pr(\varepsilon_1 = x) = e^{-x}e^{-e^{-x}},$$

Then

(14.17)

$$Y_i = \begin{cases} 1 & \text{if } Y_i^{1*} > Y_i^{0*} \\ 0 & \text{otherwise} \end{cases}.$$

This model has a separate latent variable and a separate set of regression coefficients for each possible outcome of the dependent variable. The reason for this separation is that it makes it easy to extend logistic regression to multi-outcome categorical variables, as in the multinomial logit model. In such a model, it is natural to model each possible outcome using a different set of regression coefficients. It is also possible to motivate each of the separate latent variables as the theoretical utility associated with making the associated choice, and thus motivate logistic regression in terms of utility theory. (In terms of utility theory, a rational actor always chooses the choice with the greatest associated utility.) This is the approach taken by economists when formulating discrete choice models, because it both provides a theoretically strong foundation and facilitates intuitions about the model, which in turn makes it easy to consider various sorts of extensions. (See the example below.)

The choice of the type-1 extreme value distribution seems fairly arbitrary, but it makes the mathematics work out, and it may be possible to justify its use through rational choice theory.

It turns out that this model is equivalent to the previous model, although this seems non-obvious, since there are now two sets of regression coefficients and error variables, and the error variables have a different distribution. In fact, this model reduces directly to the previous one with the following substitutions:

$$\beta = \beta_1 - \beta_0,$$

$$\varepsilon = \varepsilon_1 - \varepsilon_0.$$

An intuition for this comes from the fact that, since we choose based on the maximum of two values, only their difference matters, not the exact values — and this effectively removes one degree of freedom. Another critical fact is that the difference of two type-1 extreme-value-distributed variables is a logistic distribution, i.e. if

$$\varepsilon = \varepsilon_1 - \varepsilon_0 \sim \text{Logistic}(0,1).$$

We can demonstrate the equivalent as follows: (14.18)

$$\begin{aligned} \Pr(Y_i = 1 | \mathbf{X}_i) &= \Pr(Y_i^{1*} > Y_i^{0*} | \mathbf{X}_i) \\ &= \Pr(Y_i^{1*} - Y_i^{0*} > 0 | \mathbf{X}_i) \\ &= \Pr(\boldsymbol{\beta}_1 \cdot \mathbf{X}_i + \varepsilon_1 - (\boldsymbol{\beta}_0 \cdot \mathbf{X}_i + \varepsilon_0) > 0) \\ &= \Pr((\boldsymbol{\beta}_1 \cdot \mathbf{X}_i - \boldsymbol{\beta}_0 \cdot \mathbf{X}_i) + (\varepsilon_1 - \varepsilon_0) > 0) \\ &= \Pr((\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) \cdot \mathbf{X}_i + (\varepsilon_1 - \varepsilon_0) > 0) \\ &= \Pr((\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) \cdot \mathbf{X}_i + \varepsilon > 0) \\ &= \Pr(\boldsymbol{\beta} \cdot \mathbf{X}_i + \varepsilon > 0) \\ &= \Pr(\varepsilon > -\boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= \Pr(\varepsilon < \boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= (\boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= p_i. \end{aligned}$$

### EXAMPLE 1. Province-level Election

As an example, consider a province-level election where the choice is between a right-of-center party, a left-of-center party, and a secessionist party (e.g. the Parti Québécois, which wants Quebec to secede from Canada (Hale & Hale, 2006)). We would then use three latent variables, one for each choice. Then, in accordance with utility theory, we can then interpret the latent variables as expressing the utility that results from making each of the choices. We can also interpret the regression

coefficients as indicating the strength that the associated factor (i.e. explanatory variable) has in contributing to the utility — or more correctly, the amount by which a unit change in an explanatory variable changes the utility of a given choice. A voter might expect that the right-of-center party would lower taxes, especially on rich people. This would give low-income people no benefit, i.e. no change in utility (since they usually don't pay taxes); would cause moderate benefit (i.e. somewhat more money, or moderate utility increase) for middle-income people; and would cause significant benefits for high-income people. On the other hand, the left-of-center party might be expected to raise taxes and offset it with increased welfare and other assistance for the lower and middle classes. This would cause significant positive benefit to low-income people, perhaps weak benefit to middle-income people, and significant negative benefit to high-income people. Finally, the secessionist party would take no direct actions on the economy, but simply secede. A low-income or middle-income voter might expect basically no clear utility gain or loss from this, but a high-income voter might expect negative utility, since he/she is likely to own companies, which will have a harder time doing business in such an environment and probably lose money.

These intuitions can be expressed as follows:

Estimated strength of regression coefficient for different outcomes (party choices) and different values of explanatory variables

|               | Center-right | Center-left | Secessionist |
|---------------|--------------|-------------|--------------|
| High-income   | strong +     | strong -    | strong -     |
| Middle-income | moderate +   | weak +      | none         |
| Low-income    | none         | strong +    | none         |

This clearly shows that

1. Separate sets of regression coefficients need to exist for each choice. When phrased in terms of utility, this can be seen very easily. Different choices have different effects on net utility; furthermore, the effects vary in complex ways that depend on the characteristics of each

individual, so there need to be separate sets of coefficients for each characteristic, not simply a single extra per-choice characteristic.

2. Even though income is a continuous variable, its effect on utility is too complex for it to be treated as a single variable. Either it needs to be directly split up into ranges, or higher powers of income need to be added so that polynomial regression on income is effectively done.

#### 14.6.5. As a “log-linear” model

Yet another formulation combines the two-way latent variable formulation above with the original formulation higher up without latent variables, and in the process provides a link to one of the standard formulations of the multinomial logit (Greene, 2011).

Here, instead of writing the logit of the probabilities  $p_i$  as a linear predictor, we separate the linear predictor into two, one for each of the two outcomes:

$$\ln \Pr(Y_i = 0) = \beta_0 \cdot X_i - \ln Z \quad (14.19)$$

$$\ln \Pr(Y_i = 1) = \beta_1 \cdot X_i - \ln Z. \quad (14.20)$$

Note that two separate sets of regression coefficients have been introduced, just as in the two-way latent variable model, and the two equations appear a form that writes the logarithm of the associated probability as a linear predictor, with an extra term  $-\ln Z$  at the end. This term, as it turns out, serves as the normalizing factor ensuring that the result is a distribution. This can be seen by exponentiating both sides:

$$\Pr(Y_i = 0) = \frac{1}{Z} e^{\beta_0 \cdot X_i} \quad (14.21)$$

$$\Pr(Y_i = 1) = \frac{1}{Z} e^{\beta_1 \cdot X_i}. \quad (14.22)$$

In this form it is clear that the purpose of  $Z$  is to ensure that the resulting distribution over  $Y_i$  is in fact a probability distribution, i.e. it sums to 1. This means that  $Z$  is simply the sum of all un-normalized probabilities, and by dividing each probability by  $Z$ , the probabilities become “normalized”. That is:

$$Z = e^{\beta_0 \cdot X_i} + e^{\beta_1 \cdot X_i},$$

and the resulting equations are

$$\Pr(Y_i = 0) = \frac{e^{\beta_0 \cdot X_i}}{e^{\beta_0 \cdot X_i} + e^{\beta_1 \cdot X_i}}$$

$$\Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot X_i}}{e^{\beta_0 \cdot X_i} + e^{\beta_1 \cdot X_i}}.$$

Or generally:

$$\Pr(Y_i = c) = \frac{e^{\beta_c \cdot X_i}}{\sum_h e^{\beta_h \cdot X_i}}. \quad (14.25)$$

This shows clearly how to generalize this formulation to more than two outcomes, as in multinomial logit (Greene, 2011).

In order to prove that this is equivalent to the previous model, note that the above model is over specified, in that  $\Pr(Y_i = 0)$  and  $\Pr(Y_i = 1)$  cannot be independently specified:  $\Pr(Y_i = 0) + \Pr(Y_i = 1) = 1$  rather so knowing one automatically determines the other. As a result, the model is nonidentifiable, in that multiple combinations of  $\beta_0$  and  $\beta_1$  will produce the same probabilities for all possible explanatory variables. In fact, it can be seen that adding any constant vector to both of them will produce the same probabilities:

$$\Pr(Y_i = 1) = \frac{e^{(\beta_1 + C) \cdot X_i}}{e^{(\beta_0 + C) \cdot X_i} + e^{(\beta_1 + C) \cdot X_i}}$$

$$= \frac{e^{\beta_1 \cdot X_i} e^{C \cdot X_i}}{e^{\beta_0 \cdot X_i} e^{C \cdot X_i} + e^{\beta_1 \cdot X_i} e^{C \cdot X_i}}$$

$$= \frac{e^{\beta_1 \cdot X_i} e^{C \cdot X_i}}{e^{C \cdot X_i} (e^{\beta_0 \cdot X_i} + e^{\beta_1 \cdot X_i})}$$

$$= \frac{e^{\beta_1 \cdot X_i}}{(e^{\beta_0 \cdot X_i} + e^{\beta_1 \cdot X_i})}$$

As a result, we can simplify matters, and restore identifiability, by picking an arbitrary value for one of the two vectors. We choose to set  $\beta_0 = 0$ . Then,

$$e^{\beta_0 \cdot X_i} = e^{0 \cdot X_i} = 1,$$

and so

$$\Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot X_i}}{1 + e^{\beta_1 \cdot X_i}} = \frac{1}{1 + e^{-\beta_1 \cdot X_i}} = p_i,$$
(14.26)

which shows that this formulation is indeed equivalent to the previous formulation. (As in the two-way latent variable formulation, any settings where  $\beta = \beta_1 - \beta_0$  will produce equivalent results.)

Note that most treatments of the multinomial logit model start out either by extending the “log-linear” formulation presented here or the two-way latent variable formulation presented above, since both clearly show the way that the model could be extended to multi-way outcomes (Greene, 2011). In general, the presentation with latent variables is more common in econometrics and political science, where discrete choice models and utility theory reign, while the “log-linear” formulation here is more common in computer science, e.g. machine learning and natural language processing.

#### 14.6.6. As a single-layer perceptron

The model has an equivalent formulation

$$p_i = \frac{1}{1 + e^{-(\beta_{10} + \beta_1 \cdot X_{1,i} + \dots + k \cdot X_{k,i})}}.$$
(14.27)

This functional form is commonly called a single-layer perceptron or single-layer artificial neural network (Da & Xiurun, 2005). A single-layer neural network computes a continuous output instead of a step function. The derivative of  $p_i$  with respect to  $X = (x_1, \dots, x_k)$  is computed from the general form:

$$y = \frac{1}{1 + e^{-f(x)}},$$
(14.28)

where  $f(X)$  is an analytic function in  $X$ . With this choice, the single-layer neural network is identical to the logistic regression model. This function

has a continuous derivative, which allows it to be used in backpropagation. This function is also preferred because its derivative is easily calculated:

$$\frac{dy}{dX} = y(1 - 4) \frac{df}{dX}.$$

#### 14.6.7. In terms of binomial data

A closely related model assumes that each  $i$  is associated not with a single Bernoulli trial but with  $n_i$  independent identically distributed trials, where the observation  $Y_i$  is the number of successes observed (the sum of the individual Bernoulli-distributed random variables), and hence follows a binomial distribution:

$$Y_i \sim \text{Bin}(n_i, p_i), \text{ for } i = 1, \dots, n.$$

An example of this distribution is the fraction of seeds ( $p_i$ ) that germinate after  $n_i$  are planted. In terms of expected values, this model is expressed as follows:

$$p_i = E \left[ \frac{Y_i}{n_i} \middle| \mathbf{X}_i \right],$$

so that

$$\text{logit} \left( E \left[ \frac{Y_i}{n_i} \middle| \mathbf{X}_i \right] \right) = \text{logit}(p_i) = \ln \left( \frac{p_i}{1 - p_i} \right) = \boldsymbol{\beta} \cdot \mathbf{X}_i,$$

Or equivalently: (14.29)

$$\begin{aligned} \Pr(Y_i = y_i | \mathbf{X}_i) &= \binom{n_i}{k_i} p_i^{y_i} (1 - p_i)^{n_i - y_i} \\ &= \binom{n_i}{k_i} \left( \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{X}_i}} \right)^{y_i} \left( 1 - \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{X}_i}} \right)^{1 - y_i}. \end{aligned} \quad (14.30)$$

This model can be fit using the same sorts of methods as the above more basic model.

### 14.7. Bayesian logistic regression

In a Bayesian statistics context, prior distributions are normally placed on the regression coefficients, usually in the form of Gaussian distributions. Unfortunately, the Gaussian distribution is not the conjugate prior of the likelihood function in logistic regression; in fact, the likelihood function is not an exponential family and thus does not have a conjugate prior at all. As a result, the posterior distribution is difficult to calculate, even using standard simulation algorithms (e.g. Gibbs sampling).

There are various possibilities:

- Don't do a proper Bayesian analysis, but simply compute a maximum a posteriori point estimate of the parameters. This is common, for example, in "maximum entropy" classifiers in machine learning.
- Use a more general approximation method such as the Metropolis–Hastings algorithm.
- Draw a Markov chain Monte Carlo sample from the exact posterior by using the Independent Metropolis–Hastings algorithm with heavy-tailed multivariate candidate distribution found by matching the mode and curvature at the mode of the normal approximation to the posterior and then using the Student's  $t$  shape with low degrees of freedom. This is shown to have excellent convergence properties.
- Use a latent variable model and approximate the logistic distribution using a more tractable distribution, e.g. a Student's  $t$ -distribution or a mixture of normal distributions.
- Do probit regression instead of logistic regression. This is actually a special case of the previous situation, using a normal distribution in place of a Student's  $t$ , mixture of normals, etc. This will be less accurate but has the advantage that probit regression is extremely common, and a ready-made Bayesian implementation may already be available.
- Use the Laplace approximation of the posterior distribution. This approximates the posterior with a Gaussian distribution. This is not a terribly good approximation, but it suffices if all that is desired is an estimate of the posterior mean and variance. In such a case, an

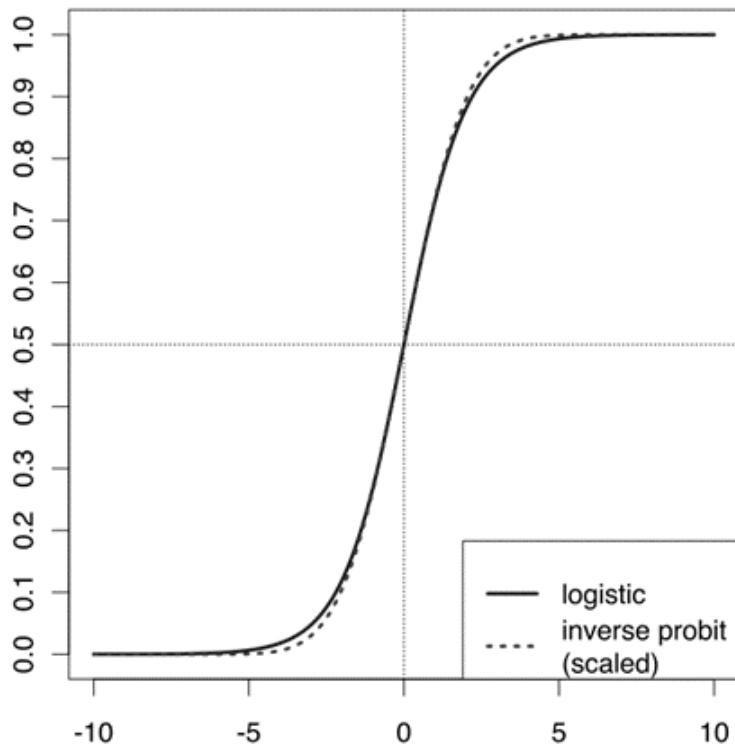
approximation scheme such as variational Bayes can be used (Bishop, 2006).

#### 14.7.1. Gibbs sampling with an approximating distribution

As shown above, logistic regression is equivalent to a latent variable model with an error variable distributed according to a standard logistic distribution. The overall distribution of the latent variable  $Y_i^*$  is also a logistic distribution, with the mean equal to  $\beta \cdot X_i$  (i.e. the fixed quantity added to the error variable). This model considerably simplifies the application of techniques such as Gibbs sampling (George & McCulloch, 1993). However, sampling the regression coefficients is still difficult, because of the lack of conjugacy between the normal and logistic distributions. Changing the prior distribution over the regression coefficients is of no help, because the logistic distribution is not in the exponential family and thus has no conjugate prior.

One possibility is to use a more general Markov chain Monte Carlo technique (Walsh, 2004), such as the **Metropolis–Hastings algorithm** (Chib & Greenberg, 1995), which can sample arbitrary distributions. Another possibility, however, is to replace the logistic distribution with a similar-shaped distribution that is easier to work with using Gibbs sampling. In fact, the logistic and normal distributions have a similar shape, and thus one possibility is simply to have normally distributed errors. Because the normal distribution is conjugate to itself, sampling the regression coefficients becomes easy. In fact, this model is exactly the model used in probit regression.

However, the normal and logistic distributions differ in that the logistic has heavier tails. As a result, it is more robust to inaccuracies in the underlying model (which are inevitable, in that the model is essentially always an approximation) or to errors in the data. Probit regression loses some of this robustness.



**Figure 1.** Comparison of logistic function with a scaled inverse probit function (i.e. the CDF of the normal distribution), comparing  $\sigma(x)$  vs.  $\Phi = \left(\sqrt{\frac{\pi}{8}}x\right)$ , which makes the slopes the same at the origin. This shows the heavier tails of the logistic distribution.

Another alternative is to use errors distributed as a **Student's *t*-distribution**. The Student's *t*-distribution has heavy tails, and is easy to sample from because it is the compound distribution of a normal distribution with variance distributed as an inverse gamma distribution. In other words, if a normal distribution is used for the error variable, and another latent variable, following an inverse gamma distribution, is added corresponding to the variance of this error variable, the marginal distribution of the error variable will follow a Student's *t*-distribution. Because of the various conjugacy relationships, all variables in this model are easy to sample from.

The Student's  $t$ -distribution that best approximates a standard logistic distribution can be determined by matching the moments of the two distributions. The Student's  $t$ -distribution has three parameters, and since the skewness of both distributions is always 0, the first four moments can all be matched, using the following equations:

$$\begin{aligned}\mu &= 0 \\ \frac{\nu}{\nu - 2} s^2 &= \frac{\pi^2}{3} \\ \frac{6}{\nu - 4} &= \frac{6}{5}.\end{aligned}$$

This yields the following values:

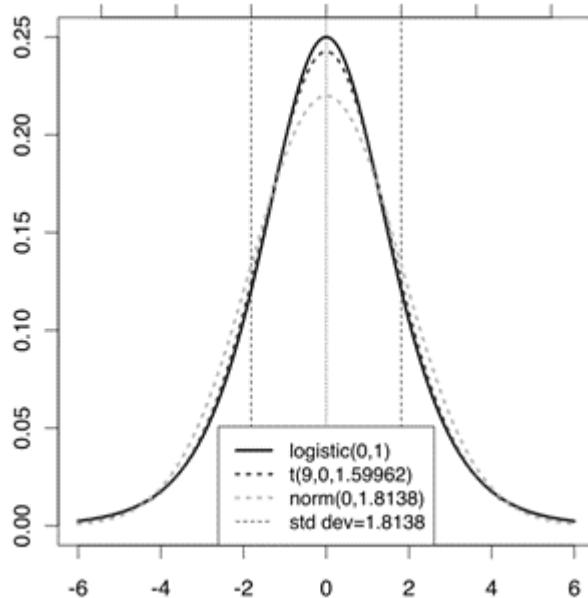
$$\mu = 0$$

$$s = \sqrt{\frac{7\pi^2}{9}} = \sqrt{\frac{7}{9}}\pi = \sqrt{\frac{7}{3}}\pi$$

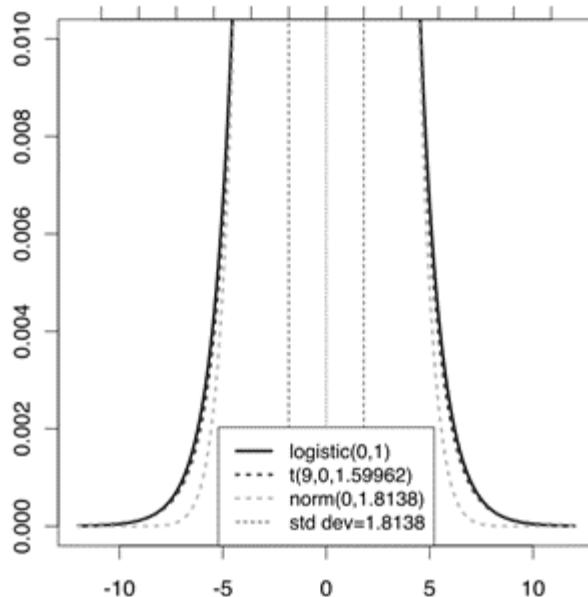
$$\nu = 9.$$

The following graphs compare the standard logistic distribution with the Student's  $t$ -distribution that matches the first four moments using the above-determined values, as well as the normal distribution that matches the first two moments. Note how much closer the Student's  $t$ -distribution agrees, especially in the tails. Beyond about two standard deviations from the mean, the logistic and normal distributions diverge rapidly, but the logistic and Student's  $t$ -distributions don't start diverging significantly until more than 5 standard deviations away.

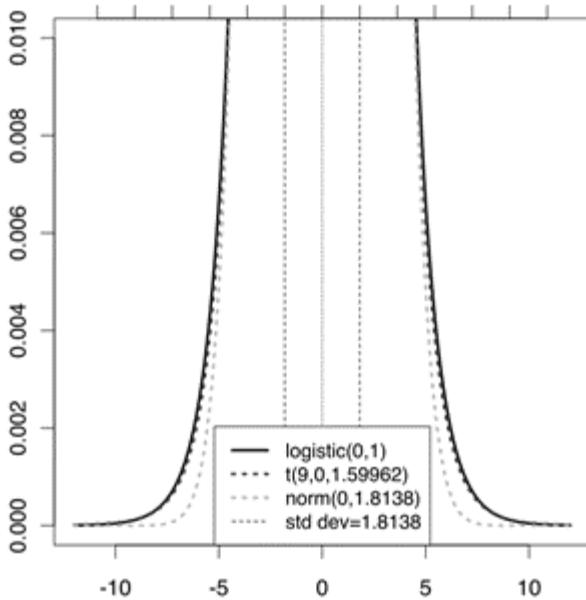
(Another possibility, also amenable to Gibbs sampling, is to approximate the logistic distribution using a mixture density of normal distributions (Chen, Zhu, Wang, Zheng, & Zhang, 2013).)



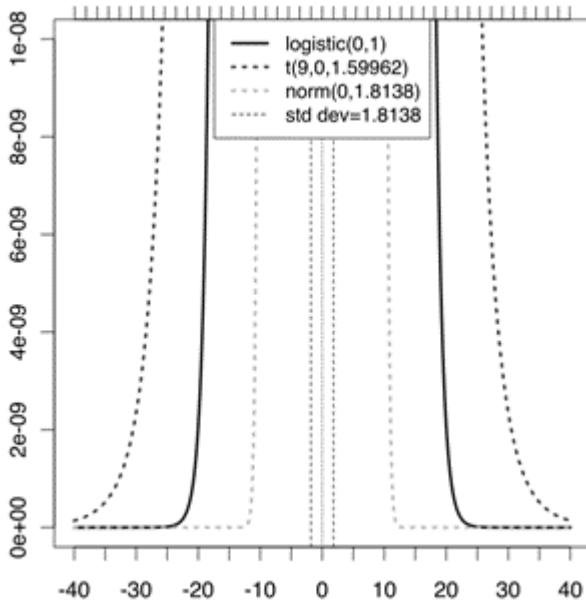
**Figure 2.** Comparison of logistic and approximating distributions ( $t$ , normal).



**Figure 4.** Tails of distributions.



**Figure 5.** Further tails of distributions.



**Figure 6.** Extreme tails of distributions.

## 14.8. Extensions

There are large numbers of extensions:

- Multinomial logistic regression (or multinomial logit) handles the case of a multi-way categorical dependent variable (with unordered values, also called “classification”). Note that the general case of having dependent variables with more than two values is termed polytomous regression.
- Ordered logistic regression (or ordered logit) handles ordinal dependent variables (ordered values).
- Mixed logit is an extension of multinomial logit that allows for correlations among the choices of the dependent variable.
- An extension of the logistic model to sets of interdependent variables is the conditional random field.

## 14.9. Model suitability

A way to measure a model’s suitability is to assess the model against a set of data that was not used to create the model (Mark & Goldberg, 2001). The class of techniques is called cross-validation. This holdout model assessment method is particularly valuable when data are collected in different settings (e.g., at different times or places) or when models are assumed to be generalizable.

To measure the suitability of a binary regression model, one can classify both the actual value and the predicted value of each observation as either 0 or 1 (Myers & Forgy, 1963). The predicted value of an observation can be set equal to 1 if the estimated probability that the observation equals 1 is above 1/2, and set equal to 0 if the estimated probability is below 1/2. Here logistic regression is being used as a binary classification model. There are four possible combined classifications:

1. prediction of 0 when the holdout sample has a 0 (True Negatives, the number of which is TN)
2. prediction of 0 when the holdout sample has a 1 (False Negatives, the number of which is FN)

3. prediction of 1 when the holdout sample has a 0 (False Positives, the number of which is FP)

4. prediction of 1 when the holdout sample has a 1 (True Positives, the number of which is TP)

These classifications are used to calculate accuracy, precision (also called positive predictive value), recall (also called sensitivity), specificity and negative predictive value:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

= fraction of observations with correct predicted classification

$$\text{Precision} = \text{PositivePredictiveValue} = \frac{TP}{TP + FP}$$

= Fraction of predicted positives that are correct

$$\text{NegativePredictiveValue} = \frac{TN}{TN + FN}$$

= fraction of predicted negatives that are correct

$$\text{Recall} = \text{Sensitivity} = \frac{TP}{TP + FN}$$

= fraction of observations that are actually 1 with a correct predicted classification

$$\text{Specificity} = \frac{TN}{TN + FP}$$

= fraction of observations that are actually 0 with a correct predicted

## 14.10. Examples Using R

### EXAMPLE 2. Logistic Regression: Multiple Numerical Predictors

Inattentional Blindness (IB) refers to situations in which a person fails to see an obvious stimulus right in front of his eyes. It is hypothesized that IB could be predicted from performance on the Stroop Color Word test. This test produces three scores: "W" (word alone, i.e., a score derived from reading a list of color words such as red, green, black), "C" (color alone, in which a score is derived from naming the color in which a series of Xs are printed), and "CW" (the Stroop task, in which a score is derived

from the subject's attempt to name the color in which a color word is printed when the word and the color do not agree). The data are in the following table, in which the response, "seen", is coded as 0=no and 1=yes The R code follows.

|    | seen | w   | c   | CW |
|----|------|-----|-----|----|
| 1  | 0    | 126 | 86  | 64 |
| 2  | 0    | 118 | 76  | 54 |
| 3  | 0    | 61  | 66  | 44 |
| 4  | 0    | 69  | 48  | 32 |
| 5  | 0    | 57  | 59  | 42 |
| 6  | 0    | 78  | 64  | 53 |
| 7  | 0    | 114 | 61  | 41 |
| 8  | 0    | 81  | 85  | 47 |
| 9  | 0    | 73  | 57  | 33 |
| 10 | 0    | 93  | 50  | 45 |
| 11 | 0    | 116 | 92  | 49 |
| 12 | 0    | 156 | 70  | 45 |
| 13 | 0    | 90  | 66  | 48 |
| 14 | 0    | 120 | 73  | 49 |
| 15 | 0    | 99  | 68  | 44 |
| 16 | 0    | 113 | 110 | 47 |
| 17 | 0    | 103 | 78  | 52 |
| 18 | 0    | 123 | 61  | 28 |
| 19 | 0    | 86  | 65  | 42 |
| 20 | 0    | 99  | 77  | 51 |
| 21 | 0    | 102 | 77  | 54 |
| 22 | 0    | 120 | 74  | 53 |
| 23 | 0    | 128 | 100 | 56 |
| 24 | 0    | 100 | 89  | 56 |
| 25 | 0    | 95  | 61  | 37 |
| 26 | 0    | 80  | 55  | 36 |
| 27 | 0    | 98  | 92  | 51 |
| 28 | 0    | 111 | 90  | 52 |
| 29 | 0    | 101 | 85  | 45 |
| 30 | 0    | 102 | 78  | 51 |
| 31 | 1    | 100 | 66  | 48 |
| 32 | 1    | 112 | 78  | 55 |
| 33 | 1    | 82  | 84  | 37 |
| 34 | 1    | 72  | 63  | 46 |
| 35 | 1    | 72  | 65  | 47 |
| 36 | 1    | 89  | 71  | 49 |
| 37 | 1    | 108 | 46  | 29 |
| 38 | 1    | 88  | 70  | 49 |
| 39 | 1    | 116 | 83  | 67 |
| 40 | 1    | 100 | 69  | 39 |
| 41 | 1    | 99  | 70  | 43 |
| 42 | 1    | 93  | 63  | 36 |
| 43 | 1    | 100 | 93  | 62 |
| 44 | 1    | 110 | 76  | 56 |
| 45 | 1    | 100 | 83  | 36 |
| 46 | 1    | 106 | 71  | 49 |

```

47     1 115 112 66
48     1 120  87 54
49     1   97  82 41

To get them into R, try this first...
> file = "http://ww2.coastal.edu/kingw/statistics/R-
+ tutorials/text/gorilla.csv"
> read.csv(file) -> gorilla
> str(gorilla)
'data.frame': 49 obs. of 4 variables:
 $ seen: int 0 0 0 0 0 0 0 0 ...
 $ W   : int 126 118 61 69 57 78 114 81 73 93 ...
 $ C   : int 86 76 66 48 59 64 61 85 57 50 ...
 $ CW  : int 64 54 44 32 42 53 41 47 33 45 ...

If that doesn't work (and it should), try copying and pasting
this script into R at the command prompt...
### Begin copying here.
gorilla = data.frame(rep(c(0,1),c(30,19)),
c(126,118,61,69,57,78,114,81,73,93,116,156,90,120,99,113,10
3,123,
86,99,102,120,128,100,95,80,98,111,101,102,100,112,82,72,72
,
89,108,88,116,100,99,93,100,110,100,106,115,120,97),
c(86,76,66,48,59,64,61,85,57,50,92,70,66,73,68,110,78,61,65
,
77,77,74,100,89,61,55,92,90,85,78,66,78,84,63,65,71,46,70,
83,69,70,63,93,76,83,71,112,87,82),
c(64,54,44,32,42,53,41,47,33,45,49,45,48,49,44,47,52,28,42,
51,54,
53,56,56,37,36,51,52,45,51,48,55,37,46,47,49,29,49,67,39,43
,36,
62,56,36,49,66,54,41))
colnames(gorilla) = c("seen","W","C","CW")
str(gorilla)
### End copying here.

```

And if that does not work, well, you know what you have to do! We might begin like this...

```

> cor(gorilla)           ### a correlation matrix
      seen          W          C          CW
seen  1.00000000 -0.03922667 0.05437115 0.06300865
W    -0.03922667  1.00000000 0.43044418 0.35943580
C    0.05437115  0.43044418 1.00000000 0.64463361
CW   0.06300865  0.35943580 0.64463361 1.00000000

```

```

...or like this...
> with(gorilla, tapply(w, seen, mean))
   0      1
100.40000 98.89474
> with(gorilla, tapply(c, seen, mean))
   0      1
73.76667 75.36842
> with(gorilla, tapply(cw, seen, mean))
   0      1
46.70000 47.84211

```

The Stroop scale scores are moderately positively correlated with each other, but none of them appears to be related to the “seen” response variable, at least not to any impressive extent. There doesn’t appear to be much here to look at. Let’s have a go at it anyway.

Since the response is a binomial variable, a logistic regression can be done as follows...

```

> glm.out = glm(seen ~ w * c * cw, family=binomial(logit),
+ data=gorilla)
> summary(glm.out)
Call:
glm(formula = seen ~ w * c * cw, family = binomial(logit),
data = gorilla)


```

Deviance Residuals:

|  | Min     | 1Q      | Median  | 3Q     | Max    |
|--|---------|---------|---------|--------|--------|
|  | -1.8073 | -0.9897 | -0.5740 | 1.2368 | 1.7362 |

Coefficients:

|                | Estimate   | Std. Error | z value  | Pr(> z ) |
|----------------|------------|------------|----------|----------|
| (Intercept)    | -1.323e+02 | 8.037e+01  | -1.646   | 0.0998 . |
| w              | 1.316e+00  | 7.514e-01  | 1.751    | 0.0799 . |
| c              | 2.129e+00  | 1.215e+00  | 1.753    | 0.0797 . |
| cw             | 2.206e+00  | 1.659e+00  | 1.329    | 0.1837   |
| w:c            | -2.128e-02 | 1.140e-02  | -1.866   | 0.0621 . |
| w:cw           | -2.201e-02 | 1.530e-02  | -1.439   | 0.1502   |
| c:cw           | -3.582e-02 | 2.413e-02  | -1.485   | 0.1376   |
| w:c:cw         | 3.579e-04  | 2.225e-04  | 1.608    | 0.1078   |
| ---            |            |            |          |          |
| Signif. codes: | 0 ‘***’    | 0.001 ‘**’ | 0.01 ‘*’ | 0.05 ‘.’ |
|                | 0.1 ‘ ’    | 1          |          |          |

```
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 65.438 on 48 degrees of freedom
Residual deviance: 57.281 on 41 degrees of freedom
AIC: 73.281
Number of Fisher Scoring iterations: 5

> anova(glm.out, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: seen

Terms added sequentially (first to last)

          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL              48    65.438
W      1   0.0755     47    65.362  0.78351
C      1   0.3099     46    65.052  0.57775
CW     1   0.1061     45    64.946  0.74467
W:C    1   2.3632     44    62.583  0.12423
W:CW   1   0.5681     43    62.015  0.45103
C:CW   1   1.4290     42    60.586  0.23193
W:C:CW 1   3.3053     41    57.281  0.06906 .
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Two different extractor functions have been used to see the results of our analysis. What do they mean?

The first gives us what amount to regression coefficients with standard errors and a z-test, as we saw in the single variable example above. None of the coefficients are significantly different from zero (but a few are close). The deviance was reduced by 8.157 points on 7 degrees of freedom, for a p-value of...

```
> 1 - pchisq(8.157, df=7)
[1] 0.3189537
```

Overall, the model appears to have performed poorly, showing no significant reduction in deviance (no significant difference from the null model).

The second print out shows the same overall reduction in deviance, from 65.438 to 57.281 on 7 degrees of freedom. In this print out, however, the reduction in deviance is shown for each term, added sequentially first to last. Of note is the three-way interaction term, which produced a nearly significant reduction in deviance of 3.305 on 1 degree of freedom ( $p = 0.069$ ).

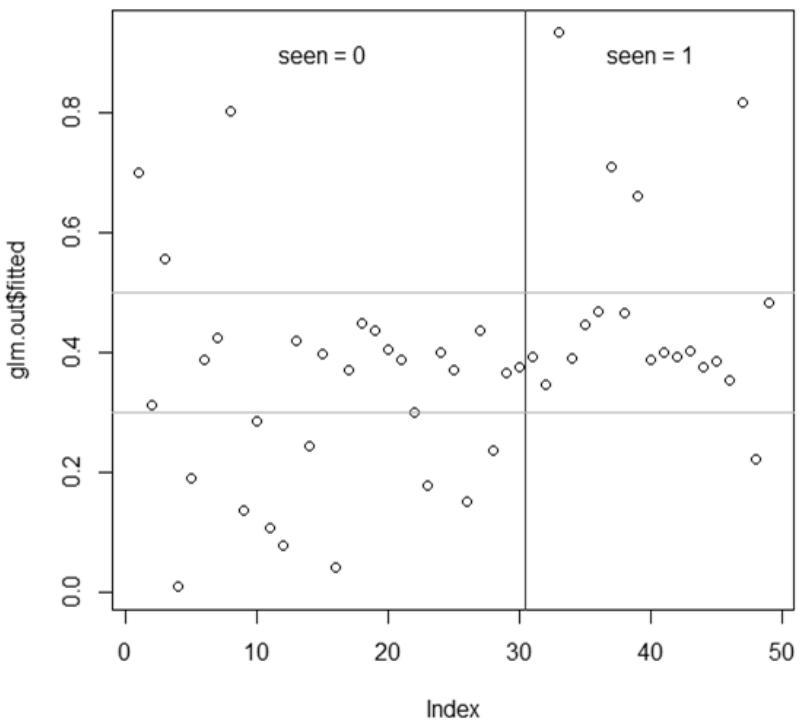
In the event you are encouraged by any of this, the following graph might be revealing...

```
> plot(glm.out$fitted)
> abline(v=30.5,col="red")
> abline(h=.3,col="green")
> abline(h=.5,col="green")
> text(15,.9,"seen = 0")
> text(40,.9,"seen = 1")
```

We leave it up to you to interpret this.

### EXAMPLE 3. Logistic Regression: Categorical Predictors

Categorical data are commonly encountered in three forms: a frequency table or cross-tabulation, a flat table, or a case-by-case data frame. Let's begin with the last of these. Copy and paste the following lines ALL AT ONCE into R. That is, highlight these lines with your mouse, hit Ctrl-C on your keyboard, click at a command prompt in R, and hit Ctrl-V on your keyboard, and hit Enter if necessary, i.e., if R hasn't returned to a command prompt. On the Mac, use Command-C and Command-V. This will execute these lines as a script and create a data frame called "ucb" in your workspace. WARNING: Your workspace will also be cleared, so save anything you don't want to lose first.



```

# Begin copying here.
rm(list=ls())
gender = rep(c("female","male"),c(1835,2691))
admitted
rep(c("yes","no","yes","no"),c(557,1278,1198,1493)) =
dept
rep(c("A","B","C","D","E","F","A","B","C","D","E","F"),
      c(89,17,202,131,94,24,19,8,391,244,299,317)) =
dept2
rep(c("A","B","C","D","E","F","A","B","C","D","E","F"),
      c(512,353,120,138,53,22,313,207,205,279,138,351)) =
department = c(dept,dept2)
ucb = data.frame(gender,admitted,department)
rm(gender,admitted,dept,dept2,department)
ls()
# End copying here.
[1] "ucb"

```

Data sets that are purely categorical are not economically represented in case-by-case data frames, and so the built-in data sets that are purely categorical come in the form of tables (contingency tables or cross-tabulations). We have just taken the data from one of these (the “UCBAdmissions” built-in data set) and turned it into a case-by-case data frame. It’s the classic University of California, Berkeley, admissions data from 1973 describing admissions into six different graduate programs broken down by gender. Let’s examine the “UCBAdmissions” data set.

```
> ftable(UCBAdmissions, col.vars="Admit")
          Admit Admitted Rejected
Gender Dept
Male   A        512     313
      B        353     207
      C        120     205
      D        138     279
      E         53     138
      F         22     351
Female A        89      19
      B        17       8
      C       202     391
      D       131     244
      E       94      299
      F        24     317
```

The data are from 1973 and show admissions by gender to the top six grad programs at the University of California, Berkeley. Looked at as a two-way table, there appears to be a bias against admitting women...

```
> dimnames(UCBAdmissions)
$Admit
[1] "Admitted" "Rejected"

$Gender
[1] "Male"     "Female"

$Dept
[1] "A" "B" "C" "D" "E" "F"

> margin.table(UCBAdmissions, c(2,1))
          Admit
```

| Gender | Admitted | Rejected |
|--------|----------|----------|
| Male   | 1198     | 1493     |
| Female | 557      | 1278     |

However, there are also relationships between “Gender” and “Dept” as well as between “Dept” and “Admit”, which means the above relationship may be confounded by “Dept” (or “Dept” might be a lurking variable, in the language of traditional regression analysis). Perhaps a logistic regression with the binomial variable “Admit” as the response can tease these variables apart.

If there is a way to conveniently get that flat table into a data frame (without splitting an infinitive), I do not know it. So I had to do this...

```
> ucb.df =
+ data.frame(gender=rep(c("Male","Female"),c(6,6)),
+ dept=rep(LETTERS[1:6],2),
+ yes=c(512,353,120,138,53,22,89,17,202,131,94,24),
+ no=c(313,207,205,279,138,351,19,8,391,244,299,317))
> ucb.df
   gender dept yes no
1    Male     A  512 313
2    Male     B  353 207
3    Male     C  120 205
4    Male     D  138 279
5    Male     E   53 138
6    Male     F   22 351
7 Female     A   89  19
8 Female     B   17   8
9 Female     C  202 391
10 Female    D  131 244
11 Female    E   94 299
12 Female    F   24 317
```

Once again, we do not have a binary coded response variable, so the last two columns of this data frame will have to be bound into the columns of a table to serve as the response in the model formula...

```
> mod.form = "cbind(yes,no) ~ gender * dept"
```

```
> glm.out = glm(mod.form, family=binomial(logit),  
+ data=ucb.df)
```

We used a trick here of storing the model formula in a data object, and then entering the name of this object into the `glm()` function. That way, if we made a mistake in the model formula (or want to run an alternative model), we have only to edit the “`mod.form`” object to do it.

Let’s see what we have found...

```
> options(show.signif.stars=F) # turn off significance  
                                # stars (optional)  
> anova(glm.out, test="Chisq")  
Analysis of Deviance Table  
  
Model: binomial, link: logit  
  
Response: cbind(yes, no)  
  
Terms added sequentially (first to last)  
  
          Df Deviance Resid. Df Resid. Dev Pr(>Chi)  
NULL           11     877.06  
gender        1      93.45      10     783.61 < 2.2e-16  
dept         5      763.40       5      20.20 < 2.2e-16  
gender:dept  5      20.20       0      0.00  0.001144
```

This is a saturated model, meaning we have used up all our degrees of freedom, and there is no residual deviance left over at the end. Saturated models always fit the data perfectly. In this case, it appears the saturated model is required to explain the data adequately. If we leave off the interaction term, for example, we will be left with a residual deviance of 20.2 on 5 degrees of freedom, and the model will be rejected ( $p = 0.001144$ ). It appears all three terms are making a significant contribution to the model.

How they are contributing appears if we use the other extractor...

```

> summary(glm.out)

Call:
glm(formula = mod.form, family = binomial(logit), data =
ucb.df)

Deviance Residuals:
[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.5442    0.2527   6.110 9.94e-10
genderMale     -1.0521    0.2627  -4.005 6.21e-05
deptB          -0.7904    0.4977  -1.588 0.11224
deptC          -2.2046    0.2672  -8.252 < 2e-16
deptD          -2.1662    0.2750  -7.878 3.32e-15
deptE          -2.7013    0.2790  -9.682 < 2e-16
deptF          -4.1250    0.3297 -12.512 < 2e-16
genderMale:deptB 0.8321    0.5104   1.630 0.10306
genderMale:deptC 1.1770    0.2996   3.929 8.53e-05
genderMale:deptD 0.9701    0.3026   3.206 0.00135
genderMale:deptE 1.2523    0.3303   3.791 0.00015
genderMale:deptF 0.8632    0.4027   2.144 0.03206

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 8.7706e+02 on 11 degrees of freedom
Residual deviance: -1.6676e-13 on 0 degrees of freedom
AIC: 92.94

Number of Fisher Scoring iterations: 3

```

These are the regression coefficients for each predictor in the model, with the base level of each factor being suppressed. Remember, we are predicting log odd”...

```
> exp(-1.0521)      # antilog of the genderMale coefficient  
[1] 0.3492037  
> 1/exp(-1.0521)
```

```
[1] 2.863658
```

This shows that men were actually at a significant *disadvantage* when department and the interaction are controlled. The odds of a male being admitted were only 0.35 times the odds of a female being admitted. The reciprocal of this turns it on its head. All else being equal, the odds of female being admitted were 2.86 times the odds of a male being admitted.

Each coefficient compares the corresponding predictor to the base level. So...

```
> exp(-2.2046)  
[1] 0.1102946
```

...the odds of being admitted to department C were only about 1/9th the odds of being admitted to department A, all else being equal. If you want to compare, for example, department C to department D, do this...

```
> exp(-2.2046) / exp(-2.1662) # C:A / D:A leaves C:D  
[1] 0.9623279
```

All else equal, the odds of being admitted to department C were 0.96 times the odds of being admitted to department D. (To be honest, I am not sure I am comfortable with the interaction in this model. You might want to examine the interaction, and if you think it doesn't merit inclusion, run the model again without it. Statistics are nice, but in the end it's what makes sense that should rule the day.)

## 14.11. Problems

1. The data set `infert` contains factors bearing on the human female fertility. The dependent variable is `case` (`yes=1`, `no =0`).
  - a. Construct a logistic regression that will predict `case` based on the factors `age`, `parity`, `education`, spontaneous (abortions) and induced (abortions).
  - b. Which factors are most predictive for `case`?

- c. Use the survival package and the conditional logistic regression model below to analyze the indicated factors.

```
model3 <- clogit(case ~ spontaneous + induced + strata(stratum), data = infert)
```

2. The following R program and output (including plots) represent a fitted logistic model containing the predictors age, blood.pressure, sex and cholesterol, with age fitted with a smooth 5-knot restricted cubic spline function and a different shape of the age relationship for males and females. There are three different treatments: a, b and c. The binary dependent variable  $y$  is simulated and

$$\text{Prob}(y = 1) = \frac{1}{[1 + \exp(-L)]}$$

As an intermediate step, predict mean cholesterol from age using a proportional odds ordinal logistic model. Use the output to analyze the fit of the model (ensure you analyze all output and comment on each).

```
require(rms)
n <- 1000 # define sample size
set.seed(17) # so can reproduce the results
age <- rnorm(n, 50, 10)
blood.pressure <- rnorm(n, 120, 15)
cholesterol <- rnorm(n, 200, 25)
sex <- factor(sample(c('female','male'), n,TRUE))
treat <- factor(sample(c('a','b','c'), n,TRUE))

# Specify population model for log odds that Y=1

L <- .4*(sex=='male') + .045*(age-50) +
  (log(cholesterol - 10)-5.2)*(-2*(sex=='female')) +
  2*(sex=='male') +
  .3*sqrt(blood.pressure-60)-2.3 + 1*(treat=='b')

# Simulate binary y to have Prob(y=1) = 1/[1+exp(-L)]
y <- ifelse(runif(n) < plogis(L), 1, 0)

ddist <- datadist(age, blood.pressure, cholesterol, sex, treat)
options(datadist='ddist')

fit <- lrm(y ~ rcs(blood.pressure,4) +
            sex * (age + rcs(cholesterol,4)) + sex*treat*age)
summary(fit)

# Use xyplot to display predictions in 9 panels, with error bars,
# with superposition of two treatments
```

```

dat <- expand.grid(treat=levels(treat), sex=levels(sex),
                    age=c(20,40,60), blood.pressure=120,
                    cholesterol=seq(100,300, length=10))

# Add variables linear.predictors and se.fit to dat

dat <- cbind(dat, predict(fit, dat, se.fit=TRUE))

# This is much easier with Predict
# xYplot in Hmisc extends xyplot to allow error bars

xYplot(Cbind(linear.predictors, linear.predictors-1.96*se.fit,
              linear.predictors+1.96*se.fit) ~ cholesterol | sex*age,
        groups=treat, data=dat, type='b')

# Since blood.pressure doesn't interact with anything, we can quickly and
# interactively try various transformations of blood.pressure, taking
# the fitted spline function as the gold standard. We are seeking a
# linearizing transformation even though this may lead to falsely
# narrow confidence intervals if we use this data-dredging-based
# transformation

bp <- 70:160
logit <- predict(fit, expand.grid(treat="a", sex='male',
                                    age=median(age),
                                    cholesterol=median(cholesterol),
                                    blood.pressure=bp),
                  type="terms")[, "blood.pressure"]
summary(logit)

#Note: if age interacted with anything, this would be the age
#      "main effect" ignoring interaction terms
#Could also use Predict(f, age=ag)$yhat
#which allows evaluation of the shape for any level of interacting
#factors. When age does not interact with anything, the result from
#predict(f, \dots, type="terms") would equal the result from
#plot if all other terms were ignored

plot(bp^.5, logit) # try square root vs. spline transform.
plot(bp^1.5, logit) # try 1.5 power
plot(sqrt(bp-60), logit)

#Some approaches to making a plot showing how predicted values
#vary with a continuous predictor on the x-axis, with two other
#predictors varying

combos <- gendata(fit, age=seq(10,100,by=10),
                   cholesterol=c(170,200,230),
                   blood.pressure=c(80,120,160))

#treat, sex not specified -> set to mode
#can also used expand.grid

```

```

 combos$pred <- predict(fit, combos)
 xyplot(pred ~ age | cholesterol*blood.pressure, data=combos,
 type='l')
 xyplot(pred ~ age | cholesterol, groups=blood.pressure,
 data=combos, type='l')
 key() # Key created by xyplot
 xyplot(pred ~ age, groups=interaction(cholesterol,blood.pressure),
 data=combos, type='l', lty=1:9)
 key()

# Add upper and lower 0.95 confidence limits for individuals

 combos <- cbind(combos, predict(fit, combos, conf.int=.95))
 xyplot(cbind(linear.predictors, lower, upper) ~ age | cholesterol,
 groups=blood.pressure, data=combos, type='b')
 key()

# Plot effects of treatments (all pairwise comparisons) vs.
# levels of interacting factors (age, sex)

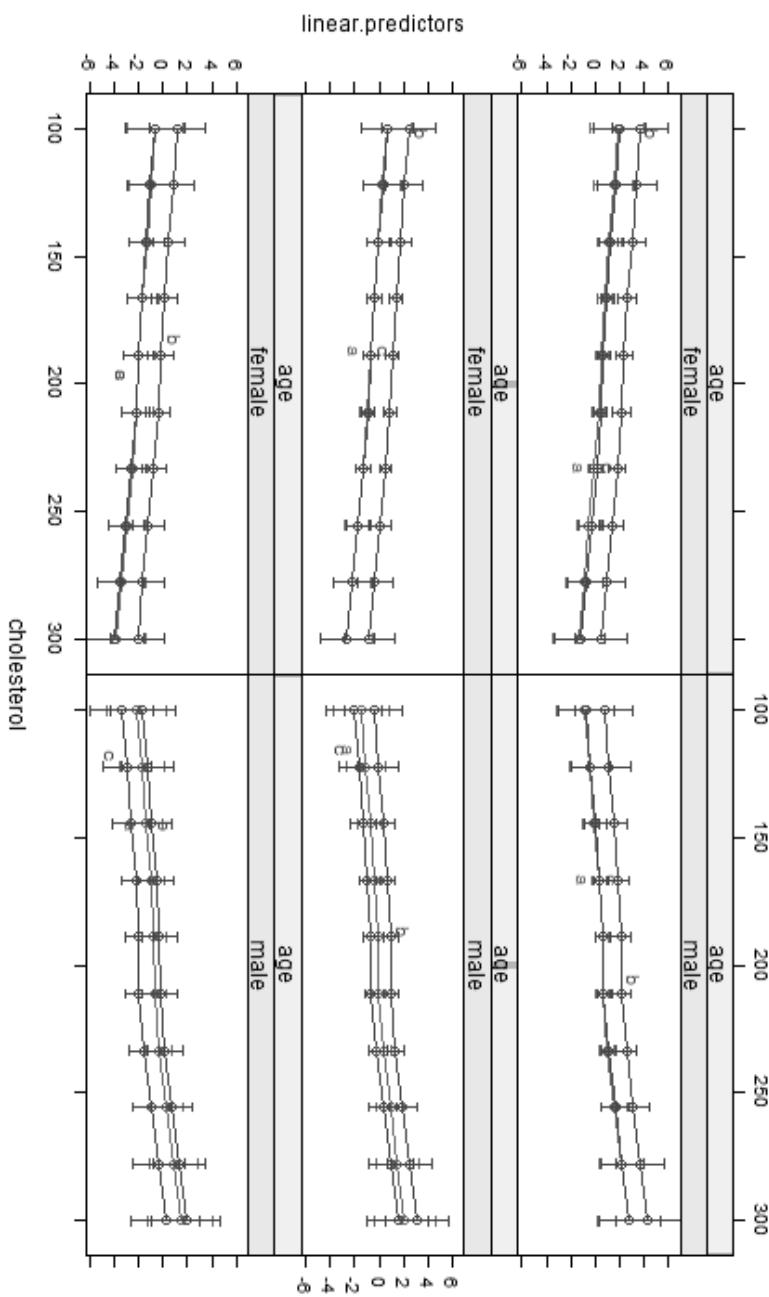
 d <- gendata(fit, treat=levels(treat), sex=levels(sex),
 age=seq(30,80,by=10))
 x <- predict(fit, d, type="x")
 betas <- fit$coef
 cov <- vcov(fit, intercepts='none')

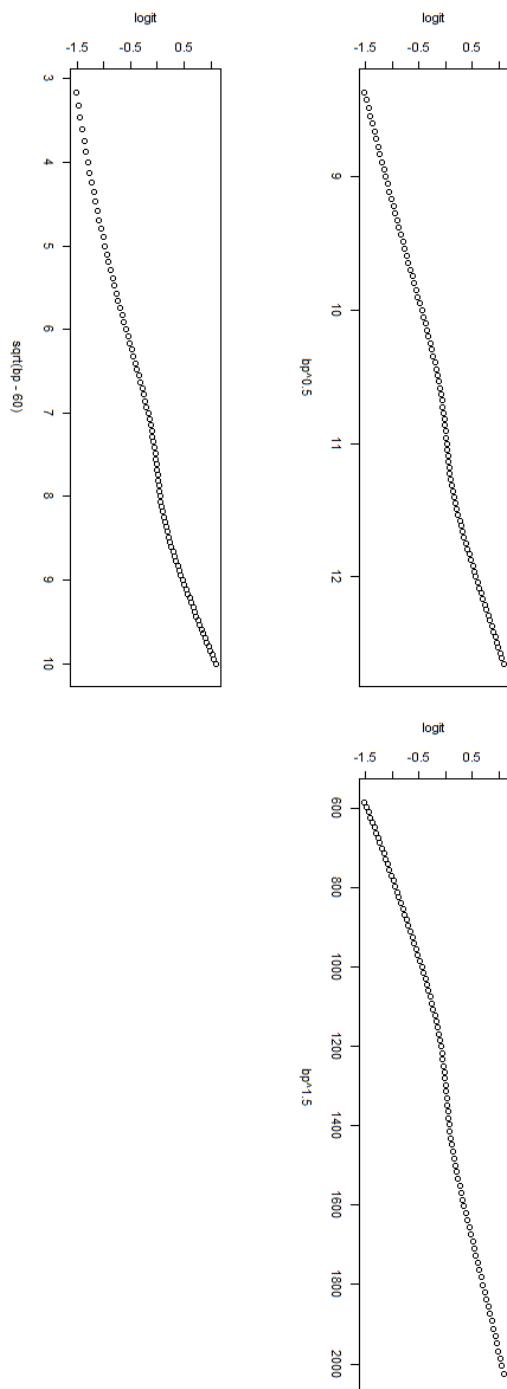
 i <- d$treat=="a"; xa <- x[i,]; Sex <- d$sex[i]; Age <- d$age[i]
 i <- d$treat=="b"; xb <- x[i,]
 i <- d$treat=="c"; xc <- x[i,]

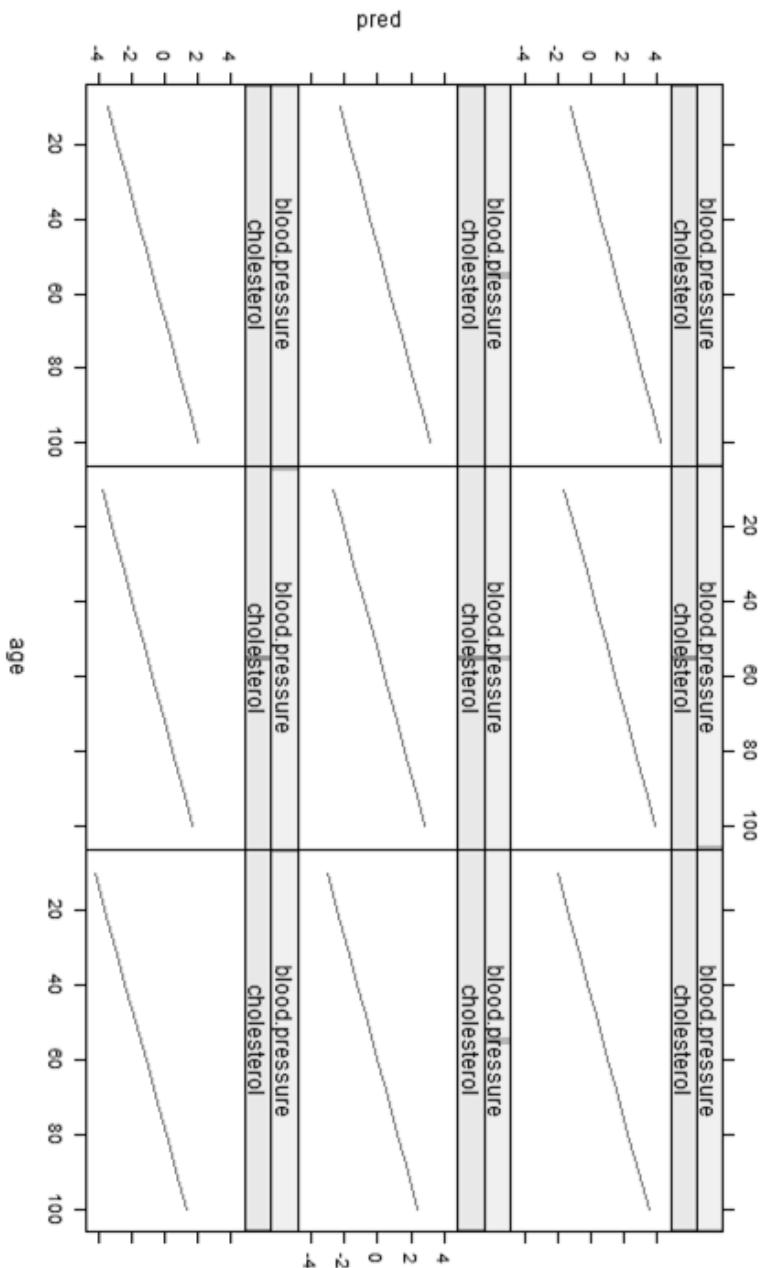
 doit <- function(xd, lab) {
   xb <- matxv(xd, betas)
   se <- apply((xd %*% cov) * xd, 1, sum)^.5
   q <- qnorm(1-.01/2) # 0.99 confidence limits
   lower <- xb - q * se; upper <- xb + q * se
   #Get odds ratios instead of linear effects
   xb <- exp(xb); lower <- exp(lower); upper <- exp(upper)
   #First elements of these agree with
   #summary(fit, age=30, sex='female',conf.int=.99)
   for(sx in levels(Sex)) {
     j <- Sex==sx
     errbar(Age[j], xb[j], upper[j], lower[j], xlab="Age",
            ylab=paste(lab, "Odds Ratio"), ylim=c(.1, 20), log='y')
     title(paste("Sex:", sx))
     abline(h=1, lty=2)
   }
 }

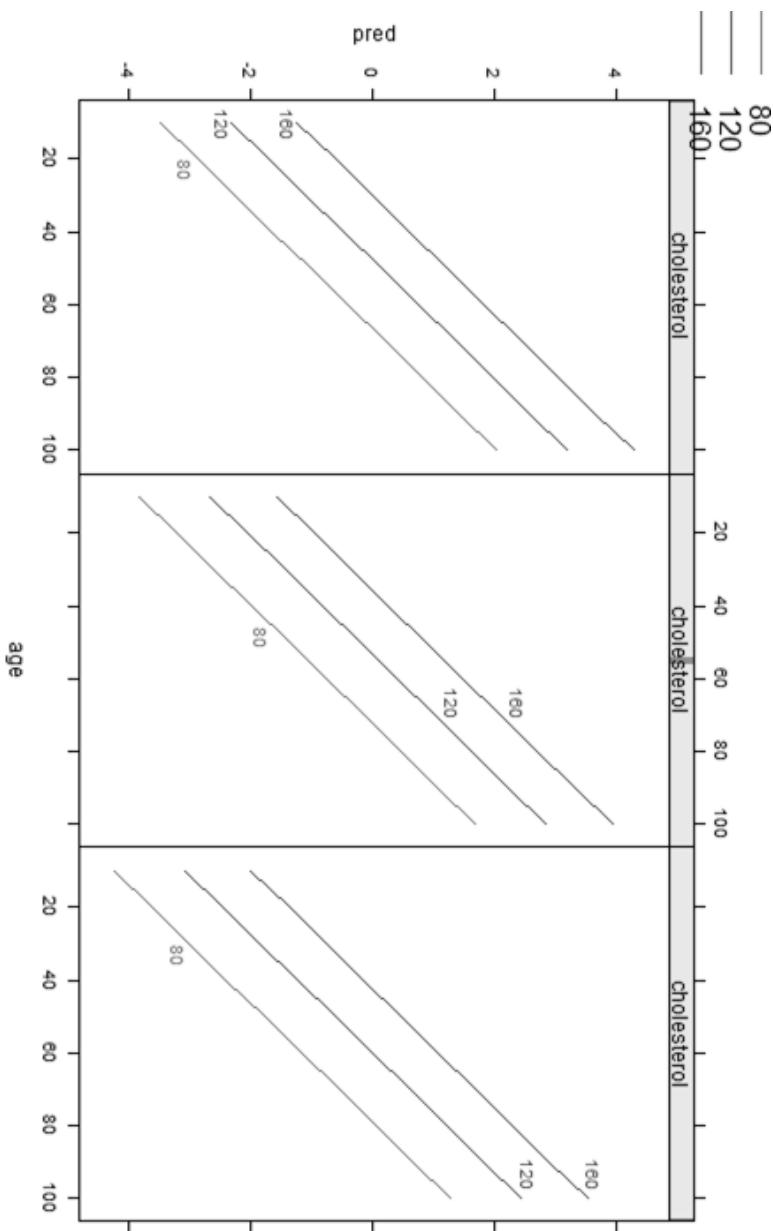
 par(mfrow=c(3,2), oma=c(3,0,3,0))
 doit(xb - xa, "b:a")
 doit(xc - xa, "c:a")
 doit(xb - xa, "c:b")

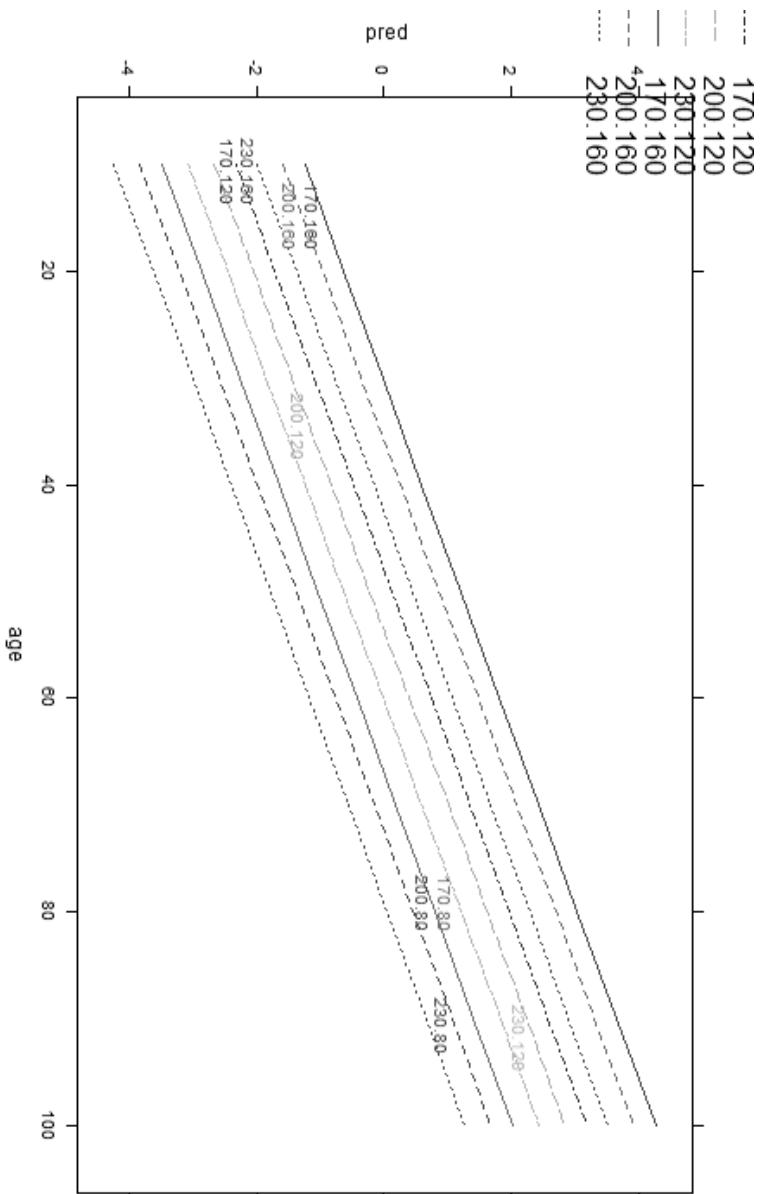
```

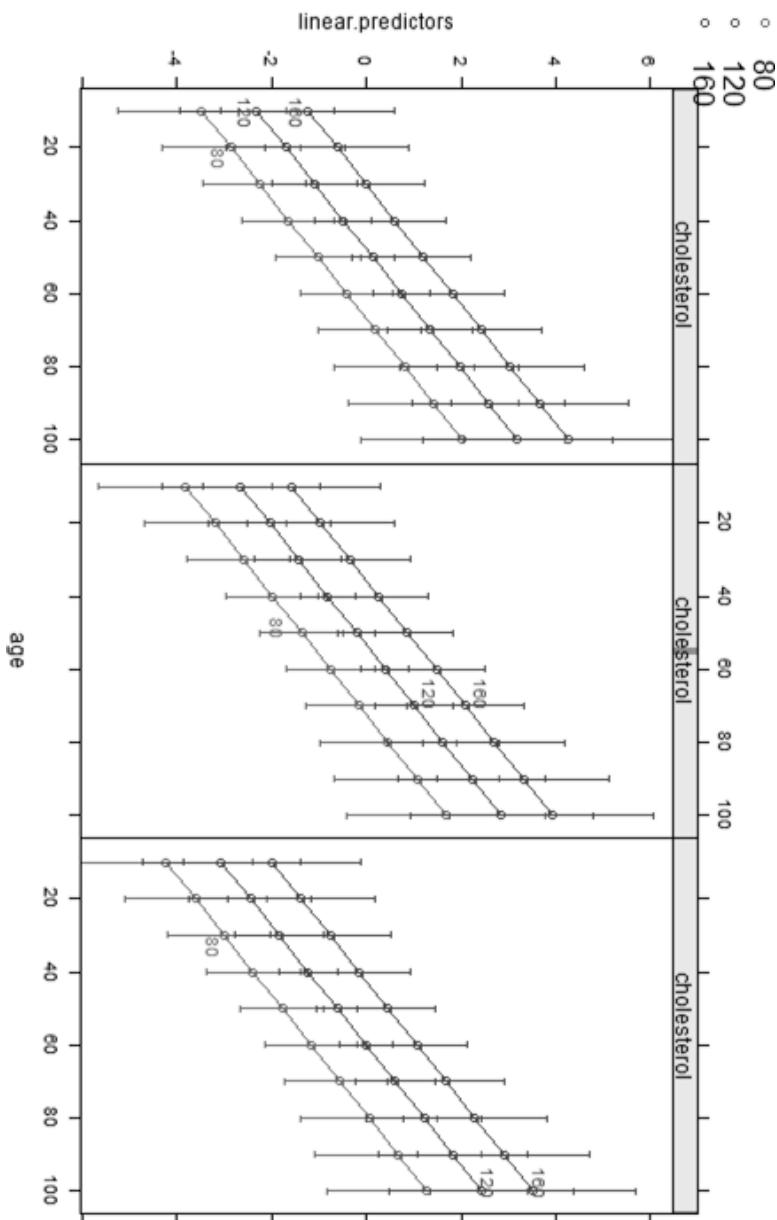


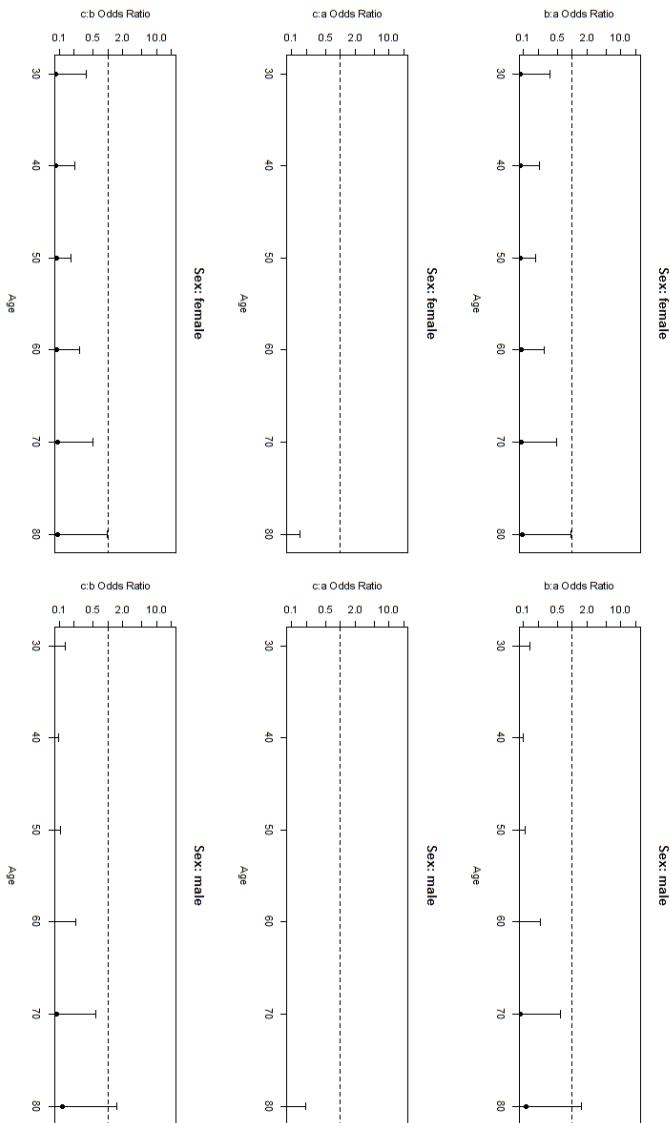












```
> summary(fit)
```

|                   | Effects |         |        | Response : y |      |       |      |       |      |  |
|-------------------|---------|---------|--------|--------------|------|-------|------|-------|------|--|
| Factor            | Low     | High    | Diff.  | Effect       | S.E. | Lower | 0.95 | Upper | 0.95 |  |
| blood.pressure    | 109.580 | 130.580 | 20.991 | 0.33         | 0.20 | -0.05 | 0.72 |       |      |  |
| Odds Ratio        | 109.580 | 130.580 | 20.991 | 1.40         | NA   | 0.95  | 2.05 |       |      |  |
| age               | 43.531  | 56.677  | 13.146 | 0.80         | 0.26 | 0.29  | 1.32 |       |      |  |
| Odds Ratio        | 43.531  | 56.677  | 13.146 | 2.23         | NA   | 1.34  | 3.73 |       |      |  |
| cholesterol       | 183.730 | 216.530 | 32.796 | -0.34        | 0.26 | -0.85 | 0.18 |       |      |  |
| Odds Ratio        | 183.730 | 216.530 | 32.796 | 0.71         | NA   | 0.43  | 1.20 |       |      |  |
| sex - male:female | 1.000   | 2.000   | NA     | 0.44         | 0.26 | -0.07 | 0.95 |       |      |  |
| Odds Ratio        | 1.000   | 2.000   | NA     | 1.56         | NA   | 0.94  | 2.59 |       |      |  |
| treat - b:a       | 1.000   | 2.000   | NA     | 1.84         | 0.27 | 1.30  | 2.37 |       |      |  |

```

Odds Ratio 1.000 2.000 NA 6.28 NA 3.68 10.70
treat - c:a 1.000 3.000 NA 0.11 0.24 -0.36 0.57
Odds Ratio 1.000 3.000 NA 1.11 NA 0.70 1.77
Adjusted to: sex=female age=50.25048 cholesterol=200.4846 treat=a

> summary(logit)
   Min. 1st Qu. Median Mean 3rd Qu. Max.
-1.50900 -0.71000 -0.05364 -0.15460 0.36370 1.08600
> cov
              blood.pressureblood.pressure'blood.pressure'
              sex=malesex=maleage
blood.pressure 2.99E-04 -7.33E-04 2.70E-03 -7.88E-04 -1.89E-06
blood.pressure' -7.33E-04 2.26E-03 -9.26E-03 -2.51E-03 -1.50900
blood.pressure'' 2.70E-03 -9.26E-03 4.02E-02 1.93E-02 0.71000
sex=male -7.88E-04 -2.51E-03 1.93E-02 1.45E+01 -0.05364
age -1.89E-06 -9.92E-06 6.97E-05 1.26E-02 3.96E-04
cholesterol -8.31E-06 7.31E-06 3.33E-06 3.52E-02 -8.31E-06
cholesterol' 1.62E-05 -1.69E-06 -7.42E-05 -8.72E-02 1.62E-05
cholesterol'' -4.48E-05 -5.62E-05 5.07E-04 3.13E-01 -4.48E-05
treat=b -8.09E-04 5.78E-04 1.02E-03 7.65E-01 -8.09E-04
treat=c -6.50E-04 1.45E-03 -5.95E-03 6.74E-01 -6.50E-04
sex=male*age 3.69E-06 -4.07E-06 2.62E-05 -2.42E-02 3.69E-06
sex=male*cholesterol 2.11E-06 1.81E-05 -1.28E-04 -7.67E-02 2.11E-06
sex=male*cholesterol' 2.39E-05 -1.27E-04 6.77E-04 1.90E-01 2.39E-05
sex=male*cholesterol'' -1.09E-04 5.82E-04 -3.08E-03 -6.90E-01 -1.09E-04
sex=male*treat=b 1.02E-03 -1.15E-03 3.19E-03 -1.32E+00 1.02E-03
sex=male*treat=c 1.11E-03 -3.38E-03 1.45E-02 -1.11E+00 1.11E-03
age*treat=b 2.03E-05 -1.77E-05 4.37E-07 -1.56E-02 2.03E-05
age*treat=c 1.23E-05 -3.09E-05 1.29E-04 -1.35E-02 1.23E-05
sex=male*age*treat=b -2.32E-05 3.30E-05 -1.07E-04 2.58E-02 3.92E-04
sex=male*age*treat=c -2.11E-05 7.00E-05 -3.12E-04 2.08E-02 3.96E-04
                                cholesterol cholesterol' cholesterol'' treat=b treat=c
blood.pressure -8.31E-06 1.62E-05 -4.48E-05 -8.09E-04 -6.50E-04
blood.pressure' 7.31E-06 -1.69E-06 -5.62E-05 5.78E-04 1.45E-03
blood.pressure'' 3.33E-06 -7.42E-05 5.07E-04 1.02E-03 -5.95E-03
sex=male 3.52E-02 -8.72E-02 3.13E-01 7.65E-01 6.74E-01
age -4.27E-05 7.75E-05 -2.43E-04 2.00E-02 2.02E-02
cholesterol 2.18E-04 -5.47E-04 1.97E-03 -1.59E-03 2.08E-03
cholesterol' -5.47E-04 1.70E-03 -6.77E-03 3.13E-03 2.59E-03
cholesterol'' 1.97E-03 -6.77E-03 2.84E-02 -1.02E-02 -5.45E-03
treat=b -1.59E-03 3.13E-03 -1.02E-02 1.76E+00 1.05E+00
treat=c -2.08E-03 2.59E-03 -5.45E-03 1.05E+00 1.94E+00
sex=male*age 4.31E-05 -7.83E-05 2.45E-04 -2.00E-02 -2.02E-02
sex=male*cholesterol -2.18E-04 5.47E-04 -1.98E-03 1.58E-03 2.10E-03
sex=male*cholesterol' 5.47E-04 -1.70E-03 6.77E-03 -3.12E-03 -2.73E-03
sex=male*cholesterol'' -1.97E-03 6.77E-03 -2.84E-02 9.99E-03 6.07E-03
sex=male*treat=b 1.60E-03 -3.14E-03 1.01E-02 -1.76E+00 -1.05E+00
sex=male*treat=c 2.09E-03 -2.63E-03 5.67E-03 -1.04E+00 -1.94E+00
age*treat=b 2.47E-05 -3.87E-05 9.39E-05 -3.51E-02 2.00E-02
age*treat=c 3.61E-05 -3.58E-05 4.68E-05 -2.00E-02 -3.69E-02
sex=male*age*treat=b -2.51E-05 3.95E-05 -9.55E-05 3.51E-02 2.00E-02
sex=male*age*treat=c -3.63E-05 3.68E-05 -5.16E-05 1.99E-02 3.69E-02
                                sex=malesex=male*cholesterol sex=malesex=male*cholesterol'
blood.pressure 3.69E-06 2.11E-06 2.39E-05
blood.pressure' -4.07E-06 1.81E-05 -1.27E-04
blood.pressure'' 2.62E-05 -1.28E-04 6.77E-04
sex=male -2.42E-02 -7.67E-02 1.90E-01
age -3.95E-04 4.24E-05 -7.64E-05
cholesterol 4.31E-05 -2.18E-04 5.47E-04
cholesterol' -7.83E-05 5.47E-04 -1.70E-03
cholesterol'' 2.45E-04 -1.98E-03 6.77E-03
treat=b -2.00E-02 1.58E-03 -3.12E-03
treat=c -2.02E-02 2.10E-03 -2.73E-03
sex=male*age 6.57E-04 -5.01E-05 9.82E-05
sex=male*cholesterol -5.01E-05 4.60E-04 -1.17E-03
sex=male*cholesterol' 9.82E-05 -1.17E-03 3.68E-03
sex=male*cholesterol'' -3.25E-04 4.26E-03 -1.48E-02
sex=male*treat=b 3.29E-02 -2.20E-03 5.26E-03
sex=male*treat=c 3.31E-02 -3.38E-03 5.18E-03
age*treat=b 3.91E-04 -2.44E-05 3.92E-05

```

|   |                       |                       |             |           |
|---|-----------------------|-----------------------|-------------|-----------|
| age*treat=c   | 3.97E-04              | -3.65E-05             | 3.86E-05    |           |
| sex=males*age*treat=b                                 | -6.53E-04             | 4.04E-05              | -8.97E-05   |           |
| sex=males*age*treat=c                                 | -6.58E-04             | 6.80E-05              | -1.01E-04   |           |
| sex=males*cholesterol'                                | sex=males*age*treat=b | sex=males*age*treat=c | age*treat=b |           |
| blood.pressure  | -1.09E-04             | 1.02E-03              | 1.11E-03    | 2.03E-05  |
| blood.pressure'                                       | 5.82E-04              | -1.15E-03             | -3.38E-03   | -1.77E-05 |
| blood.pressure''                                      | -3.08E-03             | 3.19E-03              | 1.45E-02    | 4.37E-07  |
| sex=males   | -6.90E-01             | -1.32E+00             | -1.11E+00   | -1.56E-02 |
| age   | 2.37E-04              | -2.00E-02             | -2.02E-02   | -3.92E-04 |
| cholesterol   | -1.97E-03             | 1.60E-03              | 2.09E-03    | 2.47E-05  |
| cholesterol'  | 6.77E-03              | -3.14E-03             | -2.63E-03   | -3.87E-05 |
| cholesterol''   | -2.84E-02             | 1.01E-02              | 5.67E-03    | 9.39E-05  |
| treat=b   | 9.99E-03              | -1.76E+00             | -1.04E+00   | -3.51E-02 |
| treat=c   | 6.07E-03              | -1.05E+00             | -1.94E+00   | -2.00E-02 |
| sex=males*age   | -3.25E-04             | 3.29E-02              | 3.31E-02    | 3.91E-04  |
| sex=males*cholesterol                                 | 4.26E-03              | -2.20E-03             | -3.38E-03   | -2.44E-05 |
| sex=males*cholesterol'                                | -1.48E-02             | 5.26E-03              | 5.18E-03    | 3.92E-05  |
| sex=males*cholesterol''                               | 6.32E-02              | -2.05E-02             | -1.28E-02   | -9.32E-05 |
| sex=males*treat=b                                     | -2.05E-02             | 3.84E+00              | 1.71E+00    | 3.51E-02  |
| sex=males*treat=c                                     | -1.28E-02             | 1.71E+00              | 3.44E+00    | 2.00E-02  |
| age*treat=b   | -9.32E-05             | 3.51E-02              | 2.00E-02    | 7.30E-04  |
| age*treat=c   | -5.85E-05             | 2.00E-02              | 3.69E-02    | 3.93E-04  |
| sex=males*age*treat=b                                 | 3.38E-04              | -7.73E-02             | -3.29E-02   | -7.29E-04 |
| sex=males*age*treat=c                                 | 2.47E-04              | -3.29E-02             | -6.58E-02   | -3.92E-04 |
| age*treat=csex=males*age*treat=bsex=males*age*treat=c |                       |                       |             |           |
| blood.pressure  | 1.23E-05              | -2.32E-05             | -2.11E-05   |           |
| blood.pressure'                                       | -3.09E-05             | 3.30E-05              | 7.00E-05    |           |
| blood.pressure''                                      | 1.29E-04              | -1.07E-04             | -3.12E-04   |           |
| sex=males   | -1.35E-02             | 2.58E-02              | 2.08E-02    |           |
| age   | -3.96E-04             | 3.92E-04              | 3.96E-04    |           |
| cholesterol   | 3.61E-05              | -2.51E-05             | -3.63E-05   |           |
| cholesterol'  | -3.58E-05             | 3.95E-05              | 3.68E-05    |           |
| cholesterol''   | 4.68E-05              | -9.55E-05             | -5.16E-05   |           |
| treat=b   | -2.00E-02             | 3.51E-02              | 1.99E-02    |           |
| treat=c   | -3.69E-02             | 2.00E-02              | 3.69E-02    |           |
| sex=males*age   | 3.97E-04              | -6.53E-04             | -6.58E-04   |           |
| sex=males*cholesterol                                 | -3.65E-05             | 4.04E-05              | 6.80E-05    |           |
| sex=males*cholesterol'                                | 3.86E-05              | -8.97E-05             | -1.01E-04   |           |
| sex=males*cholesterol''                               | -5.85E-05             | 3.38E-04              | 2.47E-04    |           |
| sex=males*treat=b                                     | 2.00E-02              | -7.73E-02             | -3.29E-02   |           |
| sex=males*treat=c                                     | 3.69E-02              | -3.29E-02             | -6.58E-02   |           |
| age*treat=b   | 3.93E-04              | -7.29E-04             | -3.92E-04   |           |
| age*treat=c   | 7.22E-04              | -3.93E-04             | -7.23E-04   |           |
| sex=males*age*treat=b                                 | -3.93E-04             | 1.62E-03              | 6.55E-04    |           |
| sex=males*age*treat=c                                 | -7.23E-04             | 6.55E-04              | 1.30E-03    |           |

# 15. What is a Simulation Model?

What is a simulation model? The question is actually misleading. A simulation can be comprised of one model or many models. For instance we might take a statistical model and simulate it using random variates. In this case we are simulating one model. But suppose we want to simulate the effectiveness of a ballistic missile defense system. This system is comprised of models for sensors, interceptors, guidance systems, command and control, and so on. So in this instance we are simulating a system of models.

Perhaps I just answered the question, “What is simulation?” Simulation is the process of executing a model over time (or another quantity) to see how a system performs. Yet this very answer is misleading as well. For a single run of a model does not generally yield reliable results. This is due to a number of inherent challenges of simulation, which I will not address here, but suffice it to say that we generally want to run a model over and over and see how it behaves on average.

There are several types of simulation models, some of which you may have heard.

## 15.1. Monte Carlo Simulation Model

Monte Carlo simulation is used to estimate stochastic processes where either the underlying probability distributions are unknown or difficult to calculate by exact computations. Monte Carlo simulations sample probability distributions for each variable to produce hundreds or thousands of possible outcomes. The results are analyzed to get probabilities of different outcomes occurring. For example, a comparison of a spreadsheet cost construction model run using traditional “what if” scenarios, and then run again with Monte Carlo simulation and Triangular probability distributions may show that the Monte Carlo analysis has a narrower range than the “what if” analysis. This is because the “what if” analysis gives equal weight to all scenarios, while Monte Carlo method hardly samples in the very low probability

regions. The samples in such regions are called "rare events". Monte Carlo simulations may be used to:

- ☛ To make inventory-ordering decisions in the face of uncertain demands
- ☛ To price bids for contracts when competitors' bids are uncertain
- ☛ To insure the consistent quality of manufactured goods when the sources of defects are uncertain
- ☛ To schedule and manage large projects when the durations of individual tasks are uncertain.
- ☛ To make investment decisions in sometimes volatile markets

## 15.2. Dynamic Simulation Models

Dynamic Simulation models include continuous system simulation and discrete system simulation. If time plays a substantive role, these are called continuous time models. Continuous systems simulations are typically solved using systems of differential equations. Discrete systems simulations are usually in the form of discrete time model or discrete event models.

## 15.3. Discrete Time Simulation Models

In Dynamic Time simulation, the models are iterated of fixed time increments. For example, we may want to experiment to the operation of a car wash over time and increment the model every ten minutes to observe the frequency that it is used through the working day. We might make decisions about its operation based on the frequency of use in peak hours. Or we might want to see how an interceptor missile performs over time as it attempts to destroy a theater ballistic missile, making observations every second. We can use these simulations to:

- ☛ To make inventory-ordering decisions in the face of uncertain demands
- ☛ To study the reliability, maintainability, and availability of machinery and systems
- ☛ To study the waiting times of customers in a certain service process

## 15.4. Discrete Event Simulation Models

With a Discrete Event simulation time plays a role, but the model is not iterated base on time. Rather, the model is iterated when an event occurs. Events are generally classified as arrivals, services, and departures (from queuing theory). The time between events is not constant, but are drawn from probability distributions. For example, we may know that the interarrival times of customers at a bank are exponentially distributed. Then, we would use random numbers to generate random arrival times. Likewise, we would have random service times. If the service times are significantly greater than the interarrival times, we might need to hire more bank tellers, for example. In the case of the ballistic missile intercept, we might want to observe its behavior every time its guidance system causes a change in course, rather than every second. We can use these simulations to study:

- ☛ Manufacturing plants
- ☛ Communications networks
- ☛ Transportation systems
- ☛ Combat operations
- ☛ Other systems that involve service, queuing, and processing

## 15.5. Simulation Architectures

Simulation architectures exist for running mixed time simulations, such as discrete time and discrete events simulations. One of the earliest architectures is known as Distributed Interactive Simulation or DIS. The High Level Architecture (HLA) is another example.

## 15.6. Conclusion

We use simulations to experiment with systems and perform analyses without having to use the actual systems. For example, it would be very disruptive to a call center if we performed experiments with it during its normal operating hours, which in many cases is 24-7. However, if we can construct a reasonable model of the call center, then we can use it to simulate call center operations and perform our experiments using the simulation.

## 15.7. Refresher on Randomness

### 15.7.1. Uniform Density Function

**Application:** Gives probability that observation will occur within a particular interval when probability of occurrence within that interval is directly proportional to interval length.

#### EXAMPLE 1. Uniform(0, 1) Random Numbers

The Uniform Density Function is used to generate random numbers in sampling and Monte Carlo simulation.

**Comments:** Special case of beta distribution.

The uniform probability density function is given by is:

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a \text{ or } x > b \end{cases}$$

(15.1)

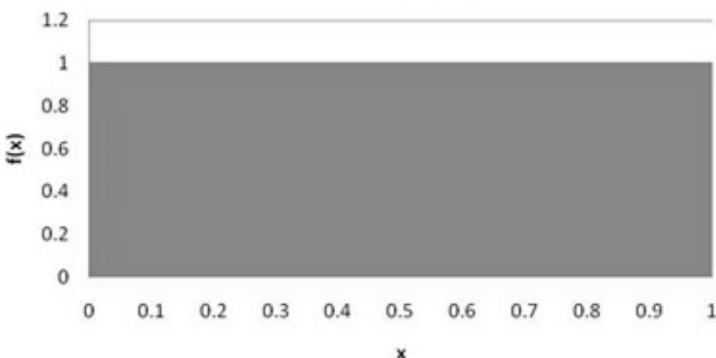
with mean =  $\frac{1}{2}(a + b)$  and variance =  $\frac{1}{12}(b - a)^2$

### 15.7.2. Random Number Generators

Random numbers are critical building block for introducing variation in simulation experiments. They are used for providing variation between simulation runs, and within each run of a simulation, by providing the basis for producing random variates (samples values or outcomes from particular probability distributions).

Random numbers in simulation provide real, uniformly generated numbers between 0 and 1, sometimes written as  $\text{Uniform} \sim (0, 1)$ . A real  $\text{Uniform} \sim (0, 1)$  probability distribution would appear as a solid rectangle from 0 to 1 with a height of 1, graphically, as depicted Figure 3. The distribution would have a mean of 0.5 and a variance of approximately 0.0833.

## Uniform $\sim (0, 1)$



**FIGURE 3.** Uniform  $(0, 1)$  Distribution

Classical uniform random number generators have some major defects, such as, short period length and lack of higher dimension uniformity. However, nowadays there are a class of rather complex generators which is as efficient as the classical generators while enjoy the property of a much longer period and of a higher dimension uniformity.

Computer programs that generate "random" numbers use an algorithm. That means if you know the algorithm and the seed-values you can predict what numbers will result. Because you can predict the numbers they are not truly random - they are pseudo-random. For statistical purposes "good" pseudo-random numbers generators are good enough.

Below, I have evaluated four Real Uniform  $(0, 1)$  random number generators using the correlation test and frequency test, as well as calculating estimates for their means and variances. The four random number generators are the Additive Congruence Generator, the Linear Congruential Generator, the Excel 2007 (based on Algorithm AS 183, Appl. Statist (1982) vol. 31, no. 2), and the ExtendSim 8 random number generator. The results of the tests for all four generators are generally, "good enough" for most discrete event simulation needs. I will caveat this with the fact that I did not examine cycles or conduct a runs test. Furthermore, I did not perform more rigorous goodness-of-fit test, such as the Anderson-Darling or Kolmogorov-Smirnov tests.

### 15.7.3. Additive Congruence Generator

The Additive Congruence Generator is given by the recursion:

$$x_i = (x_{i-1} + x_{i-k}) \bmod(m), k = 1 \dots i - 1 \quad (15.2)$$

For  $m := 5$  and  $k := 99991$

$$x_i = \text{mod}(x_{i-1} + x_{i-m}, k), w_i = \frac{x_i}{k} \quad (15.3)$$

### **Statistics**

The mean and variance for 1000 random numbers generated by this formula are very close to the "actual" mean and variance of the Uniform distribution. Again, we would have to perform more rigorous tests to determine if these estimates are "good enough," from the perspective of statistical significance.

Mean = 0.483182

Variance = 0.087944

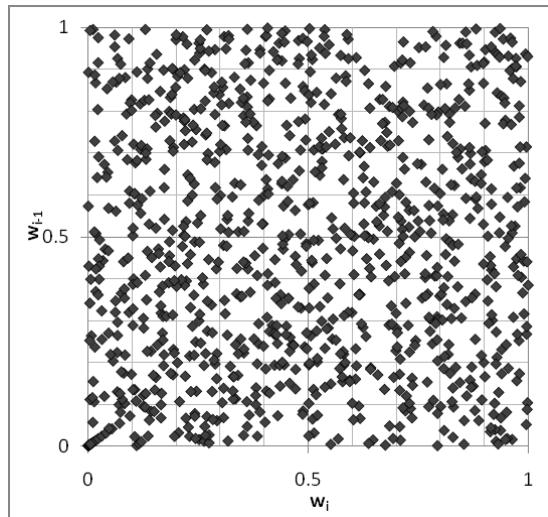
### **Correlation**

Correlation indicates the degree to which the data are linearly related. A correlation of -1 indicates there is a negative linear relationship, or that the data points approximately fit a line that slopes downward from left to right. A correlation of 1 indicates that the data approximately fits a line sloping upward from left to right. A correlation near zero indicates that the data is not linearly related. Graphically, that would look like Figure 4. The table that follows Figure 4 confirms this conclusion.

### **Frequency**

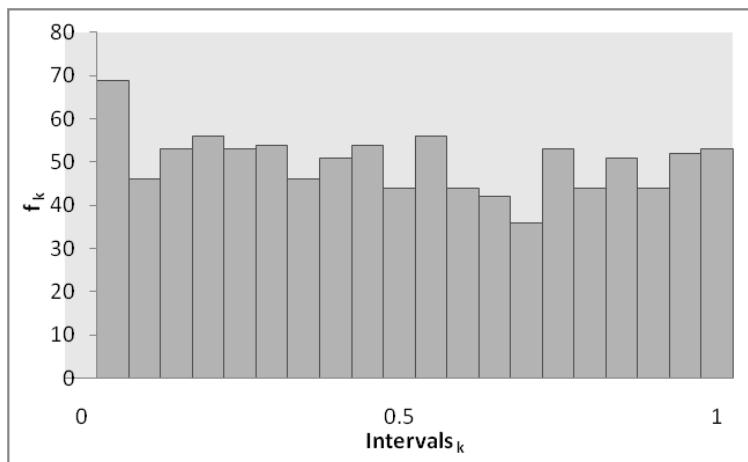
The frequency distribution shows how many data points fall within specified ranges called bin. I have selected bins from 0 to 1 with an interval of 0.1 for each bin. The data consisted of 1000 pseudo-random numbers from a uniform distribution. If the data is uniformly distributed between 0 and 1 the histogram (bar-chart) would look like a rectangle. The graph in Figure 5 is approximately the shape of a rectangle and is a good indication that the data is uniformly distributed. A more rigorous statistical test (a hypothesis test for a goodness-of-fit) would be required

to determine that the data is not uniformly distributed between 0 and 1.



**FIGURE 4.** Correlation Plot for the Additive Congruence Generator

|           | $w_i$    | $w_{i+1}$ |
|-----------|----------|-----------|
| $w_i$     | 1        |           |
| $w_{i+1}$ | 0.083997 | 1         |



**FIGURE 5.** Frequency Plot for the Additive Congruence Generator

#### 15.7.4. Linear Congruential Generator (LCG)

$$x_i = (ax_{i-1} + c) \bmod(m) \quad (15.4)$$

Where a, c, and m determine the statistical quality of the generator

Good parameter choices:

$a = 16807$  (IBM), or  $630360016$  (Simscrip)

$c = 0$

$m = 2^{31}-1=2147483647$

$x_0 = 123457$

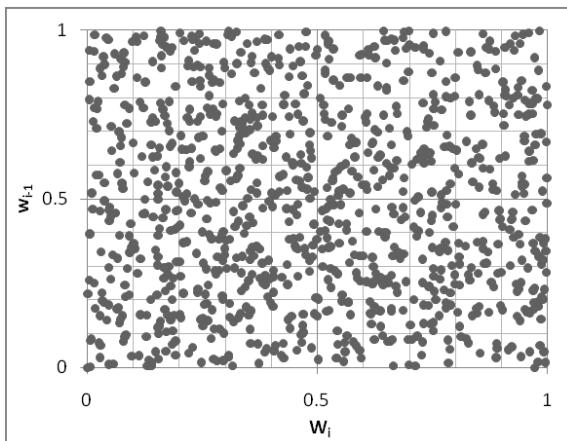
#### Statistics

Mean = 0.490559

Variance = 0.081979

#### Correlation

The correlation for LCG indicates a value closer to zero than to one, and implies data that is not highly correlated. This is shown in the graph in Figure 6 and the correlation table that follows

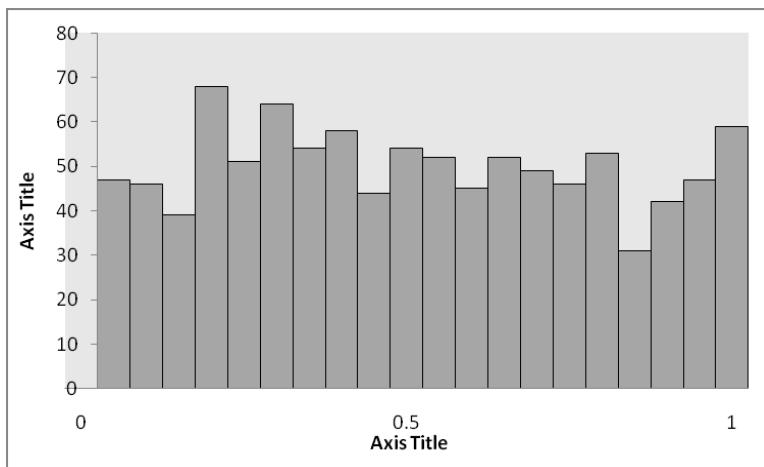


**FIGURE 6.** Correlation Plot for the Linear Congruential Generator

|           | $w_i$    | $w_{i+1}$ |
|-----------|----------|-----------|
| $w_i$     | 1        |           |
| $w_{i+1}$ | 0.131844 | 1         |

### **Frequency**

The frequency distribution shown in Figure 7 shows a distribution that is approximately uniform.



**FIGURE7.** Histogram for the Linear Congruential Generator

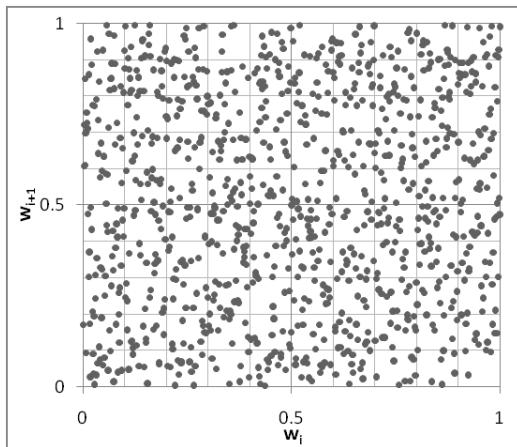
#### **15.7.5. Random Function in Excel**

The following are the results of one random sample of the Uniform (0, 1) Random number generator in *Excel*, for  $N = 1000$ .

### **Correlation**

Figure 8 shows data that seems not to be highly correlated, as does the value of  $w_i/w_{i+1} = -0.01739$ .

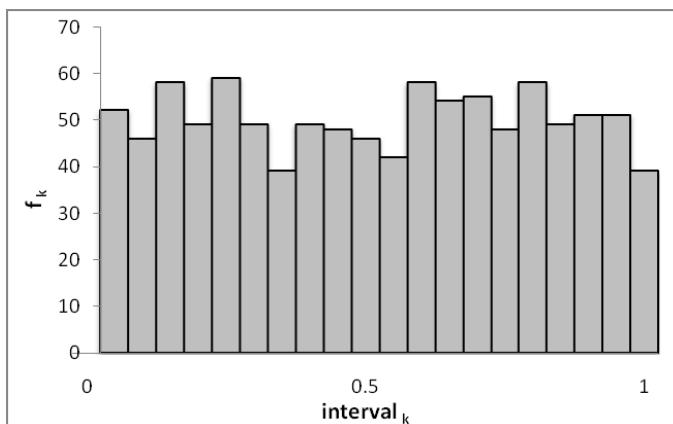
|           | $w_i$    | $w_{i+1}$ |
|-----------|----------|-----------|
| $w_i$     | 1        |           |
| $w_{i+1}$ | -0.01739 | 1         |



**FIGURE 8.** Correlation Plot for the *Excel* Random Number Generator

### Frequency

Figure 9 shows a distribution that is approximately uniform.



**FIGURE 9.** Histogram for the *Excel* Random Number Generator

### Statistics

Average = 0.501519

Variance = 0.080906

The algorithm used in *Excel* is shown below. For Details on the *Excel* Rand() function, see <http://support.microsoft.com/kb/828795>

```
real function random()
```

```

c
c      Algorithm AS 183 Appl. Statist. (1982) vol.31,
c      no.2, FORTRAN Code
c
c      Returns a pseudo-random numbers with rectangular c
c      distribution.
c      between 0 and 1.  The cycle length is 6.95E+12
c      (See page 123
c      of Applied Statistics (1984) vol.33), not as
c      claimed in the
c      original article.
c
c      IX, IY and IZ should be set to integer values
c      between 1 and
c      30000 before the first entry.
c
c      Integer arithmetic up to 30323 is required.
c
integer ix, iy, iz
common /randc/ ix, iy, iz
c
ix = 171 * mod(ix, 177) - 2 * (ix / 177)
iy = 172 * mod(iy, 176) - 35 * (iy / 176)
iz = 170 * mod(iz, 178) - 63 * (iz / 178)
c
if (ix .lt. 0) ix = ix + 30269
if (iy .lt. 0) iy = iy + 30307
if (iz .lt. 0) iz = iz + 30323
c
c      If integer arithmetic up to 5212632 is available,
c      the preceding
c      6 statements may be replaced by:
c
c      ix = mod(171 * ix, 30269)
c      iy = mod(172 * iy, 30307)
c      iz = mod(170 * iz, 30323)
c
random = mod(float(ix) / 30269. + float(iy) / 30307. +
float(iz) / 30323., 1.0)

```

```
return  
end  
c
```

### 15.7.6. Testing for Randomness

We need to test for both randomness as well as uniformity. The tests can be classified in 2 categories: Empirical or statistical tests, and theoretical tests. Theoretical tests deal with the properties of the generator used to create the realization with desired distribution, and do not look at the number generated at all. For example, we would not use a generator with poor qualities to generate random numbers. Statistical tests are based solely on the random observations produced.

Test for Randomness:

**A. Test for independence:**

Plot the  $w_i$  realization vs.  $w_{i+1}$ . If there is independence, the graph will not show any distinctive patterns at all, but will be perfectly scattered.

**B. Runs tests.(run-ups, run-downs):**

This is a direct test of the independence assumption. There are two test statistics to consider: one based on a normal approximation and another using numerical approximations.

**Test based on Normal approximation:**

Suppose you have  $N$  random realizations. Let  $a$  be the total number of runs in a sequence. If the number of positive and negative runs are greater than say 20, the distribution of  $a$  is reasonably approximated by a Normal distribution with mean  $(2N - 1)/3$  and  $(16N - 29)/90$ . Reject the hypothesis of independence or existence of runs if  $|Z_0| > Z_{(1-\alpha/2)}$  where  $Z_0$  is the Z score.

**C. Correlation tests:**

Do the random numbers exhibit discernible correlation? Compute the sample Autocorrelation Function.

**Frequency or Uniform Distribution Test:**

Use Kolmogorov-Smirnov test to determine if the realizations follow a  $U(0,1)$

### 15.7.7. Statistics for Correlated Output Data

Output analysis is the examination of data generated by a simulation. Its purpose is either to predict the performance of a system or to compare the performance of two or more alternative system designs. Estimating absolute performance means estimating the value of one or more system performance measures.

The need for statistical output analysis is based on the observation that the output data from a simulation exhibits random variability when random-number generators are used to produce the values of the input variable—that is, two different streams or sequences of random numbers will produce two sets of outputs, which (probably) will differ. If the performance of the system is measured by a parameter  $\theta$ , the results of a set of simulation experiment will be an estimator  $\hat{\theta}$  of  $\theta$ . The precision of the estimator  $\hat{\theta}$  can be measured by the standard error of  $\hat{\theta}$  or by the width of a confidence interval for  $\theta$ . The purpose of statistical analysis is either to estimate this standard error or confidence interval or to figure out the number of observations required to achieve a standard error or confidence interval of a given size—or both.

Consider a typical output variable  $Y$ , the total cost per week of an inventory system;  $Y$  should be treated as a random variable with an unknown distribution. A simulation run of length 1 week provides a single sample observation from the population of all possible observations on  $Y$ . By increasing the run length, the sample size can be increased to  $n$  observations,  $Y_1, Y_2, \dots, Y_n$ , based on a run of  $n$  weeks. However, these observations do not constitute a random sample, in the classic sense, because they are not statistically independent. In this case, the inventory on hand at the end of one week is the beginning inventory on hand the next week, and so the value of  $Y_i$  has some influence on the value of  $Y_{i+1}$ . Thus, the sequence of random variables  $Y_1, Y_2, \dots, Y_n$  could be autocorrelated (i.e., correlated with itself). This autocorrelation, which is a measure of a lack of statistical independence, means that classic methods of statistics, which assume independence, are not directly applicable to the analysis of these output data. The methods must be properly modified and the simulation experiments properly designed for valid inferences to be made.

In discrete event simulation we are frequently interested in how a set of performance measure data is autocorrelated. For example, in a steady-state analysis, the waits experienced by consecutive jobs entering a service node may be autocorrelated. If the utilization of the service node is high, there will be a high positive correlation between the wait  $w_i$  experienced by the  $i$  th job and the wait  $w_{i+1}$  experienced by the next job. There will be a statistically significant positive correlation between  $w_i$  and  $w_{i+j}$  for some range of small, positive  $j$  values. For this analysis the usual statistical measure may not be used. In these cases, we concern ourselves with  $n$  realizations that are related to time, that is having  $n$  correlated observations; the **estimate of the mean** is given by

$$\text{mean} = \sum_{i=1}^n \frac{X_i}{n} \quad (15.5)$$

Let

$$A = \sum_{j=1}^m \left(1 - \frac{j}{m+1}\right) \rho_{j,x} \quad (15.6)$$

then the **estimated variance** is:

$$\frac{(1 + 2A)S^2}{m} \quad (15.7)$$

Where

$S^2$  = the usual variance estimate

$\rho_{j,x}$  = the  $j$ th coefficient of autocorrelation

$m$  = the maximum time lag for which autocorrelations are computed, such that  $j = 1, 2, 3, \dots, m$

As a good rule of thumb, the maximum lag for which autocorrelations are computed should be approximately 2% of the number of  $n$  realizations, although each  $\rho_{j,x}$  could be tested to determine if it is significantly different from zero.

**Sample Size Determination:** We can calculate the minimum sample size required by

$$n = \frac{(1 + 2A)S^2 t^2}{\delta^2 \text{mean}^2} \quad (15.8)$$

**Application:** A pilot run was made of a model, observations numbered 150, the mean was 205.74 minutes and the variance  $s_2 = 101,921.54$ , estimate of the lag coefficients were computed as:  $\rho_{1,x} = 0.3301$   $\rho_{2,x} = 0.2993$ , and  $\rho_{3,x} = 0.1987$ . Calculate the minimum sample size to assure the estimate lies within  $\pm \delta = 10\%$  of the true mean with  $\alpha = 0.05$ .

$$\begin{aligned} n = & [(1.96)^2 (101,921.54) \{1 + 2[(1 - 1/4) 0.3301 + (1 \\ & - 2/4) 0.2993 + (1 \\ & - 3/4) 0.1987]\}] / (0.1)^2 (205.74)^2 \approx 1757 \end{aligned}$$

## 15.8. Review Problems

- Given the probability density function of a discrete random variable

|             |     |     |     |     |     |     |     |     |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| X Values    | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| Probability | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 |

- Calculate the mean value of the distribution
- Calculate the variance of the distribution

Given the variance  $\sigma^2 = 2$  for a population, calculate the variance of the sampling distribution of the population.

- Given the sample mean  $\bar{x} = 1.75$  from a population with mean  $\mu =$

2, and a sample size of  $n = 20$ , calculate the sample variance.

# 16.What is Queuing Theory?

Each of us has spent a great deal of time waiting in lines. It is estimated that Americans spend a total of 37 billion hours a year waiting in lines.

Places we wait in line...

- stores
- restaurants
- hotels
- airports
- post offices
- ATMs
- banks
- theme parks
- traffic lights
- on the phone

Waiting lines do not always contain people...

- returned videos
- subassemblies in a manufacturing plant
- electronic message on the Internet

Queuing theory deals with the analysis and management of waiting lines.

In this chapter, we develop mathematical models for waiting lines, or queues. In Section 16.1, we begin by discussing some terminology that is often used to describe queues. In Section 16.2, we look at some distributions (the exponential and the Erlang distributions) that are needed to describe queuing models. In Section 16.3, we introduce the idea of a birth–death process, which is basic to many queuing models involving the exponential distribution. The remainder of the chapter examines several models of queuing systems that can be used to answer questions like the following:

1. What fraction of the time is each server idle?
2. What is the expected number of customers present in the queue?
3. What is the expected time that a customer spends in the queue?
4. What is the probability distribution of the number of customers present in the queue?
5. What is the probability distribution of a customer's waiting time?

6. If a bank manager wants to ensure that only 1% of all customers will have to wait more than 5 minutes for a teller, how many tellers should be employed?

## 16.1. History of Queuing Theory

As you can probably tell, queue is the British term for any type of line for waiting.

The Danish engineer A. K. Erlang is credited with founding queuing theory by studying telephone switchboards in Copenhagen for the Danish Telephone Company. He developed many of the queuing results used today.

One of the greatest uses of queuing theory in the United States is for analyzing automobile traffic flow—studying how many lanes to have, how to regulate the traffic lights, and so forth—in order to maximize the flow of traffic.

## 16.2. The Purpose of Queuing Models

Queuing models are used to:

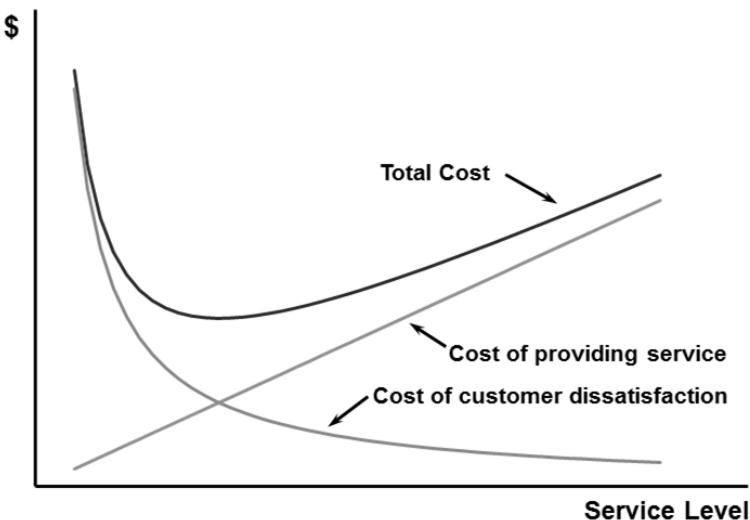
- describe the behavior of queuing systems
- determine the level of service to provide
- evaluate alternate configurations for providing service

## 16.3. Queuing System Components

A queuing system consists of a stream of arriving customers, jobs, or other entities, as well as a mechanism to provide some sort of service or processing to the arriving entities. It also has one or more facilities for entities to wait until services can be provided when demand for service exceeds the capacity to provide service.

The input process is usually called the arrival process. Arrivals are called customers. In all models that we will discuss, we assume that no more than one arrival can occur at a given instant. For a case like a restaurant, this is a very unrealistic assumption. If more than one arrival can occur at a given instant, we say that bulk arrivals are allowed.

## Queuing Costs



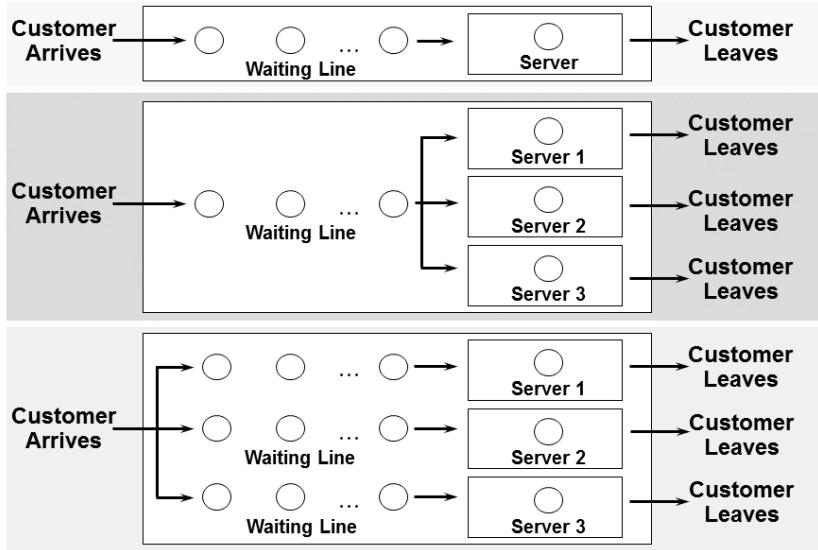
**Figure 1.** Queuing theory trade-space (cost versus service)

### 16.4. The Arrival Process

Usually, we assume that the arrival process is unaffected by the number of customers present in the system. In the context of a bank, this would imply that whether there are 500 or 5 people at the bank, the process governing arrivals remains unchanged.

There are two common situations in which the arrival process may depend on the number of customers present. The first occurs when arrivals are drawn from a small population. Suppose that there are only four ships in a naval shipyard. If all four ships are being repaired, then no ship can break down in the near future. On the other hand, if all four ships are at sea, a breakdown has a relatively high probability of occurring in the near future. Models in which arrivals are drawn from a small population are called **finite source models**. Another situation in which the arrival process depends on the number of customers present occurs when the rate at which customers arrive at the facility decreases when the facility becomes too crowded. For example, if you see that the bank parking lot is full, you might pass by and come another day. If a

customer arrives but fails to enter the system, we say that the customer has **balked**. The phenomenon of balking was described by Yogi Berra when he said, “Nobody goes to that restaurant anymore; it’s too crowded.”



**Figure 2.** Common Queuing System Configurations

If the arrival process is unaffected by the number of customers present, we usually describe it by specifying a probability distribution that governs the time between successive arrivals.

#### 16.4.1. Arrival Characteristics

Interarrival times are the times between successive customers—scheduled or random.

##### Types of arrivals

- One at a time or in batches
- Alike or separated into priority classes
- Balking—after arrival, a customer decides not to enter the system (or the choice is made by the system).
- Reneging—a customer is already in line becomes impatient and leaves the system before starting service.

Arrival rate — the manner in which customers arrive at the system for service.

Arrivals are often described by a Poisson random variable:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{ for } x = 0, 1, 2, \dots$$

$\lambda$  is the arrival rate (e.g., calls arrive at a rate of  $\lambda = 5$  per hour)

See file Poisson.xls

#### 16.4.2. Review of the Exponential Distribution

The most common choice of  $X$  is the exponential distribution, with parameter  $\lambda$ .

It has a density function  $x(t) = \lambda e^{-\lambda t}$ .

The mean interarrival time is  $E(x(t)) = \frac{1}{\lambda}$ .

The variance is  $\text{var } x(t) = \frac{1}{\lambda^2}$ .

It possesses the “no-memory” property:

$$P(X > t + h | X \geq t) = e^{-\lambda t} = P(X > h) \quad (16.1)$$

for nonnegative values  $t$  and  $h$ . This implies that this probability does not depend on the value of  $t$ .

It can be shown that no other density function can satisfy (5) (see Feller (1957)). For reasons that become apparent, a density that satisfies (5) is said to have the no-memory property. The no-memory property of the exponential distribution is important, because it implies that if we want to know the probability distribution of the time until the next arrival, then it does not matter how long it has been since the last arrival. To put it in concrete terms, suppose interarrival times are exponentially distributed with  $\lambda = 6$ . Then the no-memory property implies that no matter how long it has been since the last arrival, the probability distribution governing the time until the next arrival has the density function  $6e^{-6t}$ . This means that to predict future arrival patterns, we

need not keep track of how long it has been since the last arrival. This observation can appreciably simplify analysis of a queuing system.

### 16.4.3. Relationship between the Poisson and Exponential Distributions

If the interarrival times are exponential, the probability distribution of the number of arrivals occurring in any time interval of length  $t$  is given by the following theorem:

**Theorem 1.** Interarrival times are exponential with parameter  $\lambda$  if and only if the number of arrivals to occur in an interval of length  $t$  follows a Poisson distribution with parameter  $\lambda t$ .

If the arrival rate is stationary, if bulk arrivals cannot occur, and past arrivals do not affect future arrivals, then the following theorem holds.

**Theorem 2.** If arrivals follow a Poisson distribution with mean  $\lambda$ , Then interarrival times follow an Exponential distribution with mean  $1/\lambda$ .

That is, interarrival times are  $\text{Expon}(\lambda)$ , and the number of arrivals in any interval of length  $t$  is  $\text{Poisson}(\lambda t)$ .

#### EXAMPLE 2. Call Center

Assume calls arrive according to a Poisson distribution with mean  $\lambda = 5$  per hour. Interarrivals follow an exponential distribution with mean  $1/5 = 0.2$  per hour. On average, calls arrive every 0.2 hours or every 12 minutes. The exponential distribution exhibits the Markovian (memoryless) property.

- Stationary arrival process
- Arrivals occur one at a time
- Arrivals are independent of one another
- The average rate of arrival is constant over time
- Nonstationary arrival process
- Parameters defining the queuing system vary over time

## 16.5. Service Process

To describe the output process (often called the service process) of a queuing system, we usually specify a probability distribution—the service time distribution—which governs a customer's service time. In most cases, we assume that the service time distribution is independent of the number of customers present. This implies, for example, that the server does not work faster when more customers are present.

In this chapter, we study two arrangements of servers: servers in parallel and servers in series. Servers are in parallel if all servers provide the same type of service and a customer need only pass through one server to complete service. For example, the tellers in a bank are usually arranged in parallel; any customer need only be serviced by one teller, and any teller can perform the desired service. Servers are in series if a customer must pass through several servers before completing service. An assembly line is an example of a series queuing system.

To describe a queuing system completely, we must also describe the queue discipline and the manner in which customers join lines.

**TABLE 1.** Examples of Queuing Systems

| Situation           | Input Process  | Output Process                                 |
|---------------------|--|--|
| Bank                | Customers arrive at bank                                     | Tellers serve the customers                    |
| Pizza parlor        | Requests for pizza delivery are received                     | Pizza parlor sends out truck to deliver pizzas |
| Hospital blood bank | Pints of blood arrive  | Patients use up pints of blood                 |
| Naval shipyard      | Ships at sea break down and are sent to shipyard for repairs | Ships are repaired return to sea               |

## 16.6. Queue Discipline

The queue discipline describes the method used to determine the order in which customers are served. The most common queue discipline is the FCFS discipline (first come, first served), in which customers are

served in the order of their arrival. Under the LCFS discipline (last come, first served), the most recent arrivals are the first to enter service. If we consider exiting from an elevator to be service, then a crowded elevator illustrates an LCFS discipline. Sometimes the order in which customers arrive has no effect on the order in which they are served. This would be the case if the next customer to enter service is randomly chosen from those customers waiting for service. Such a situation is referred to as the SIRO discipline (service in random order). When callers to an airline are put on hold, the luck of the draw often determines the next caller serviced by an operator.

Finally, we consider priority queuing disciplines. A priority discipline classifies each arrival into one of several categories. Each category is then given a priority level, and within each priority level, customers enter service on an FCFS basis. Priority disciplines are often used in emergency rooms to determine the order in which customers receive treatment, and in copying and computer time-sharing facilities, where priority is usually given to jobs with shorter processing times.

The service discipline is the rule that states which customer, from all who are waiting, goes into the next service.

- First-come-first-serve (FCFS)—customers are served in the order they arrive.
- Service-in-random-order (SIRO)
- Last-come-first-served (LCFS)
- Shortest-processing-time (SPT)
- Number of servers.

Queue time is the amount of time spent awaiting service after arrival. Service time is the amount of time a customer spends receiving service (not including time in the queue). Service times are often described by an Exponential random variable:

$$P(t_1 \leq T \leq t_2) = \int_{t_1}^{t_2} \mu e^{-\mu x} dx = e^{-\mu t_1} - e^{-\mu t_2}, \text{ for } t_1 \leq t_2 \quad (16.2)$$

$\mu$  is the service rate (e.g., calls can be serviced at a rate of  $\mu = 7$  per hour), The average service time is  $1/\mu$ . (See file Exponential.xls)

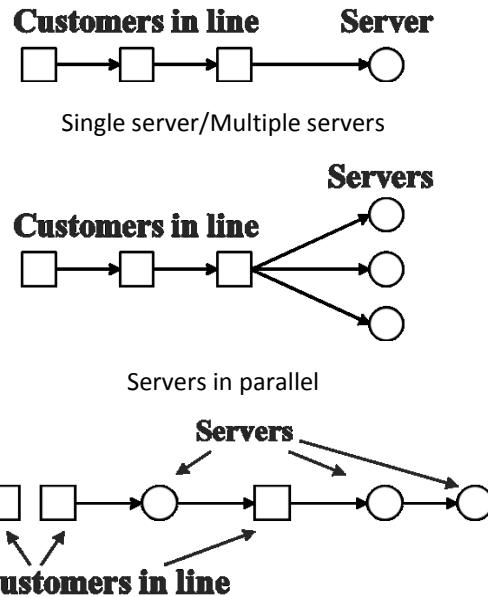


Figure 3. Servers in series—queuing network

## 16.7. Modeling the Arrival Process

As previously mentioned, we assume that at most one arrival can occur at a given instant

of time. We define  $t_i$  be the time at which the  $i$ th customer arrives. For  $i \geq 1$ , define  $T_i = t_{i+1} - t_i$  to be the  $i$ th interarrival time. In modeling the arrival process, we assume that the  $T_i$ 's are independent, continuous random variables described by the random variable  $X$ . The independence assumption means, for example, that the value of  $T_2$  has no effect on the value of  $T_3$ ,  $T_4$ , or any later  $T_i$ . The assumption that each  $T_i$  is continuous The assumption that each interarrival time is governed by the same random variable implies that the distribution of arrivals is independent of the time of day or the day of the week. This is the assumption of stationary interarrival times. Because of phenomena such as rush hours, the assumption of stationary interarrival times is often unrealistic, but we may often approximate reality by breaking the time

of day into segments. For example, if we were modeling traffic flow, we might break the day up into three segments: a morning rush hour segment, a midday segment, and an afternoon rush hour segment. During each of these segments, interarrival times may be stationary. is usually a good approximation of reality.

As we mentioned, the distribution of arrivals is independent of the time of day—stationary interarrival times. Assume  $X$  has a density function  $x(t)$ , then

$$P(X \leq c) = \int_0^c x(t) dt \text{ and } P(X > c) = \int_c^\infty x(t) dt$$

We define  $1/\lambda$  to be the mean or average interarrival time (time per arrivals), then

$$\frac{1}{\lambda} = \int_0^\infty t x(t) dt \tag{16.3}$$

We define  $\lambda$  to be the arrival rate (arrivals per time)

## 16.8. Queuing Systems Taxonomy

If the distribution of inter-arrival times and service times are exponential (with different means), then the queuing system is known as the “M/M/1” Queue. M is used for the exponential distribution due to the Markov property. The notation is “interarrival time distribution/service time distribution/ number of servers.” “M/G/1” refers to a system with any general service time distribution. “GI/M/1” refers to a system with any interarrival time distribution. “GI/G/1” (or “G/G/1”) refers to a system with any interarrival time distribution and any service time distribution. The so-called Lindley’s formula satisfies all these systems.

## 16.9. Queuing System Theory

If  $\lambda < \mu$ , that is, if the arrival rate is less than the service rate, then the system will be stable. Expressions for the waiting time for an arbitrary customer have been found for the M/M/1, M/G/1, GI/M/1, and queues

having Erlang interarrival or service times (or both). If either interarrival or service times do not have the Markov property, then such derivation is not possible. For these systems, we can use Lindley's formula and a spreadsheet to do these calculations.

### 16.9.1. Erlang Distribution

If  $X_1, X_2, \dots, X_k$  are IID exponential random variables, the sum of these  $k$  samples has an Erlang- $k$  distribution. The mean ( $\beta$ ) of each of the component exponential distributions and the number of exponential random variables ( $k$ ) are the parameter of the Erlang distribution: Erlang(ExpMean,  $k$ ). Moreover,

The range is  $[0, +\infty)$

The mean is  $k\beta$

The variance is  $k\beta^2$

### 16.9.2. Lindley's Formula

Assume a FCFS queuing system. Both service times and successive inter-arrival times are IID random variables, and they may be different. Let  $X_1, X_2, \dots$  represent successive service times and  $Y_1, Y_2, \dots$  represent successive inter-arrival times. If  $1/\lambda$  is the mean of each  $Y_i$ , and  $1/\mu$  is the mean of each  $X_i$ , then  $\lambda$  is called the arrival rate, and  $\mu$  is called the service rate. Let  $W_n$  represent the waiting time in the queue. A simple model (due to Lindley [1952]) for successive waiting times is given by:

$$W_{n+1} = \max(0, W_n + X_n - Y_n) \quad (16.4)$$

### 16.9.3. A Spreadsheet Simulation of M/M/1 Queue Waiting Times

Suppose we have a queue with exponential interarrival and service times, with mean interarrival time equal 0.7 units and mean service time equal 1.0 units. Use simulation to determine parameter estimates of the average waiting time. Also calculate the known mean waiting time, given by:

$$W_q = \frac{\lambda}{\mu(\mu-\lambda)} \quad (16.5)$$

### **EXAMPLE 3. M/M/1 Queue in Excel**

See Queue1.xls at [www.simulation-educators.com/downloads.html](http://www.simulation-educators.com/downloads.html)

1. Enter 0 as the initial waiting time in cell C10.
2. Enter  $=-\text{MeanServTime} * \text{LN}(\text{RAND}())$  in cell D10, and copy it down to cell D20 (this is the random deviate an exponential RV).
3. Enter  $=-\text{MeanIntArrTime} * \text{LN}(\text{RAND}())$  in cell E10, and copy it down to cell E20.
4. Enter  $C8+D8-E8$  in cell F10, and copy it down to F20.
5. Except for the first entry in column C, which is 0, enter the formula  $\text{IF}(F_{n-1}<0, 0, F_{n-1})$ . That is, enter  $=\text{MAX}(0, F10)$  in cell C11, and copy it down to cell C20.
6. Calculate the averages for columns C, D, and E in row 21.

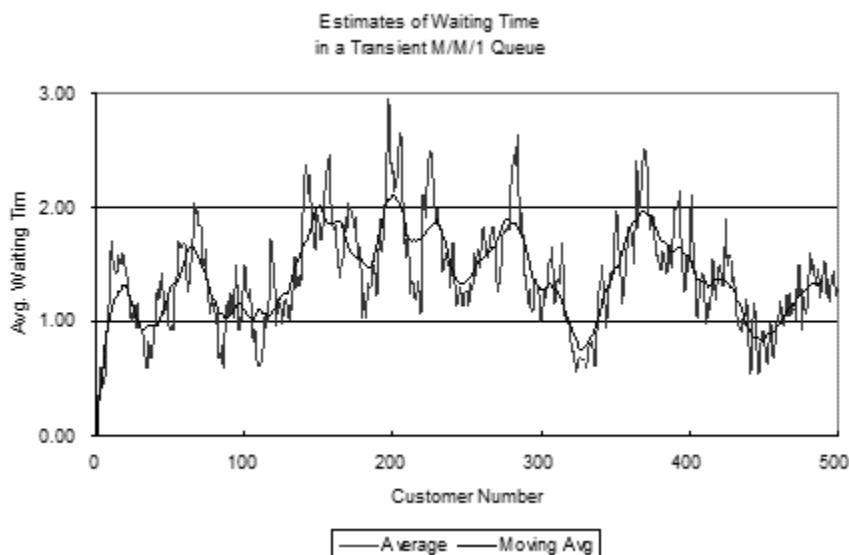
#### **16.10. Characteristics of Data from Dynamic Systems**

This system is stable because the arrival rate is less than the service rate, and the server is busy approximately 70 percent of the time over a long period. Twenty independent replications of the model, consisting of 500 observations each, were run. In each replication, the initial waiting time was set to zero, indicating that the system started in the empty-and-idle state. The graph in Figure 5 shows the average over 20 replications of each observation in the series.

See Queue2.xls at [www.simulation-educators.com/downloads.html](http://www.simulation-educators.com/downloads.html)

|    | A                                   | B               | C                 | D                 | E                       | F   | G         |
|----|-------------------------------------|-----------------|-------------------|-------------------|-------------------------|---|-----------|
| 1  | <b>Simulation of an M/M/1 Queue</b> |                 |                   |                   |                         |   |           |
| 2  |                                     |                 |                   |                   | Known mean waiting time |   |           |
| 3  | Mean service time                   |                 | 0.7               |                   |                         | 1.6333  |           |
| 4  |                                     | $\mu$           | 1.4286            |                   |                         |   |           |
| 5  | Mean interarrival time              |                 | 1.0               |                   |                         |   |           |
| 6  |                                     | $\lambda$       | 1.0               |                   |                         |   |           |
| 7  |                                     |                 |                   |                   |                         |   |           |
| 8  |                                     | Customer number | Waiting Time      | Service Time      | Interarrival Time       |   | Busy/Idle |
| 9  |                                     | (n)             | (W <sub>n</sub> ) | (X <sub>n</sub> ) | (Y <sub>n</sub> )       | (W <sub>n</sub> +X <sub>n</sub> -Y <sub>n</sub> ) |           |
| 10 |                                     | 0               | 0.0000            | 0.9491            | 1.0335                  | -0.0844   | Idle      |
| 11 |                                     | 1               | 0.0000            | 1.2326            | 1.1062                  | 0.1264  | Idle      |
| 12 |                                     | 2               | 0.1264            | 0.0009            | 0.4013                  | -0.2740   | Busy      |
| 13 |                                     | 3               | 0.0000            | 0.3232            | 1.7507                  | -1.4274   | Idle      |
| 14 |                                     | 4               | 0.0000            | 0.2573            | 0.9400                  | -0.6827   | Idle      |
| 15 |                                     | 5               | 0.0000            | 0.6396            | 0.2878                  | 0.3518  | Idle      |
| 16 |                                     | 6               | 0.3518            | 1.1383            | 0.0985                  | 1.3916  | Busy      |
| 17 |                                     | 7               | 1.3916            | 0.6984            | 1.4017                  | 0.6884  | Busy      |
| 18 |                                     | 8               | 0.6884            | 2.3139            | 0.8587                  | 2.1437  | Busy      |
| 19 |                                     | 9               | 2.1437            | 0.3934            | 0.2827                  | 2.2544  | Busy      |
| 20 |                                     | 10              | 2.2544            | 0.0083            | 4.0976                  | -1.8350   | Busy      |
| 21 | Average                             | 0.6324          | 0.7232            | 1.1144            |                         |   |           |
| 22 | Minimum                             | 0.0000          | 0.0009            | 0.0985            |                         |   |           |
| 23 | Maximum                             | 2.2544          | 2.3139            | 4.0976            |                         |   |           |

**Figure 4.** Excel simulation of the M/M/1 queue



**Figure 5.** Waiting times for the M/M/1 queue

### 16.10.1. Influences on Mean Estimation

We will discuss the Initial Transient Period next, but for now we place our attention on Autocorrelated Observations.

Waiting times are not independent—they are autocorrelated. If a customer's waiting time is small, then the next customer's waiting time is likely to be small.

The previous graph (Figure 5) shows this effect: instead of “bouncing around” above and below the stationary mean, the waiting times will “make excursions” staying above the mean, then dipping below the mean for periods of time. The practical effect of this behavior is that ordinary statistical methods cannot be applied naively to compute a reliable confidence interval for the mean. When appropriate techniques are used, a great deal of data are frequently needed.

### 16.10.2. The Initial Transient Period

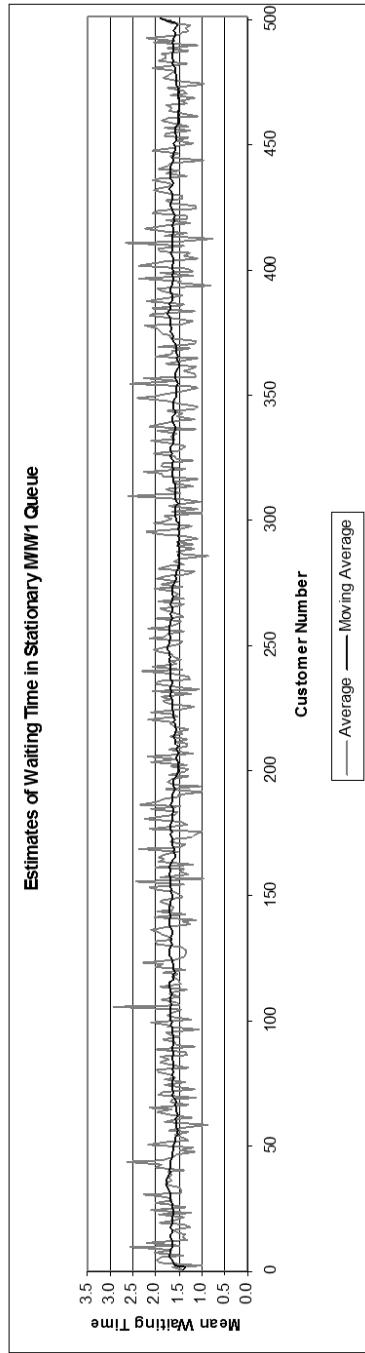
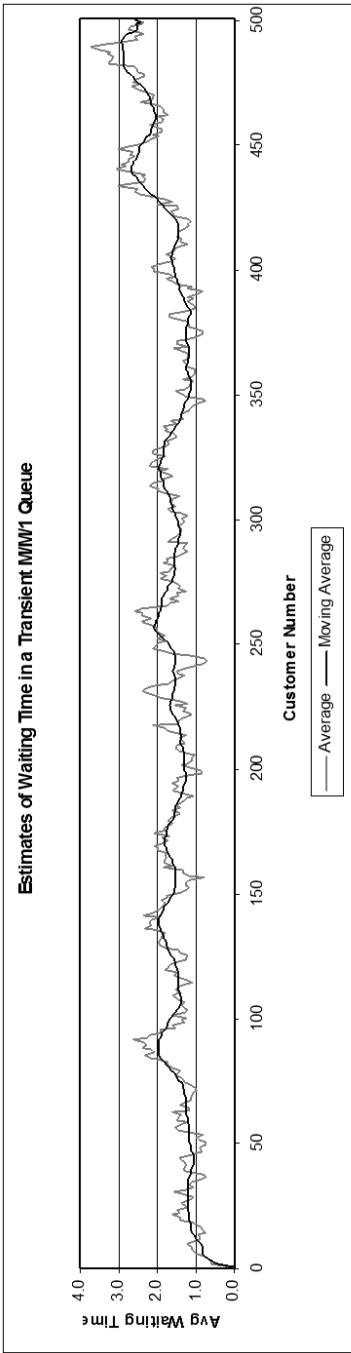
There is a “warm-up period” during which the mean waiting time appears to increase to a value close to 2.0 and then settle back to a value close to 1.6.

Using  $\lambda = 0.7$  and  $\mu = 1.0$ , we can compute that the true stationary mean waiting time is 1.6333. Thus, we see that in the first 100 observations, there is much evidence that the waiting times have settled into their stationary behavior (see Figure 6).

The actual stationary distribution of waiting time is known for the M/M/1 queue:

$$\Pr(W \leq w) = \begin{cases} 1 - \rho & \text{if } w = 0 \\ 1 - \rho e^{-(\mu-\lambda)w} & \text{if } w >= 0 \end{cases} \quad (16.6)$$

where  $\rho = \lambda/\mu$ , with mean  $1/(\mu - \lambda)$ .



**Figure 6.** Transient vs. Stationary M/M/1 Queue

If the first waiting time is chosen from this distribution, then all subsequent waiting times will be from this distribution. By generating the first waiting time, we can produce a sequence of waiting times, all of which are from the stationary distribution. The initial transient behavior is an important characteristic of observations from a dynamic system.

Every simulation of a dynamic system must begin in a fixed state. The initial state cannot be selected from the stationary distribution because the stationary distribution is not known for those systems that we wish to simulate. Moreover, every simulation of a dynamic system must go through an initial transient period before exhibiting the behavior of a stationary distribution.

#### 16.10.3. Determination of the Warm-up Period

To estimate the long-term performance measure of the system, there are several methods such as Batch Means, Independent Replications and Regenerative Method

#### 16.10.4. Batch Means Computation

Batch Means is a method of estimating the steady-state characteristic from a single-run simulation. The single run is partitioned into equal size batches large enough for estimates obtained from different batches to be approximately independent. In the method of Batch Means, it is important to ensure that the bias due to initial conditions is removed to achieve at least a covariance stationary waiting time process. An obvious remedy is to run the simulation for a period large enough to remove the effect of the initial bias. During this warm-up period, no attempt is made to record the output of the simulation. The results are thrown away. At the end of this warm-up period, the waiting time of customers are collected for analysis. The practical question is “How long should the warm-up period be?”. Abate and Whitt provided a relatively simple and nice expression for the time required ( $t_p$ ) for an M/M/1/ queue system (with traffic intensity  $\rho$ ) starting at the origin (empty) to reach and remain within 100p% of the steady- state limit as follows:

$$t_p(\rho) = 2C(\rho) \ln \frac{\left\{ \frac{1}{[(1-p)(1+2C(\rho))]} \right\}}{(1-\rho)^2}$$

where

(16.7)

$$C(\rho) = \frac{[2 + \rho + (\rho^2 + 4\rho)^{\frac{1}{2}}]}{4}.$$

Some notions of  $t_p(\rho)$  as a function of  $\rho$  and  $p$ , are given in Table 3.

**Table 2.** Time ( $tp$ ) required for an M/M/1 queue to reach and remain with  $100p\%$  limits of the steady-state value.

| Traffic Intensity | 100p   |        |         |         |
|-------------------|--------|--------|---------|---------|
| $\rho$            | 95.0   | 99.0   | 99.9    | 99.99   |
| 0.10              | 3.61   | 6.33   | 10.23   | 14.12   |
| 0.20              | 5.01   | 8.93   | 14.53   | 20.14   |
| 0.30              | 7.00   | 12.64  | 20.71   | 28.79   |
| 0.40              | 10.06  | 18.39  | 30.31   | 42.23   |
| 0.50              | 15.18  | 28.05  | 46.47   | 64.89   |
| 0.60              | 24.70  | 46.13  | 76.79   | 107.45  |
| 0.70              | 45.51  | 85.87  | 143.61  | 201.36  |
| 0.80              | 105.78 | 201.53 | 338.52  | 475.51  |
| 0.90              | 435.74 | 838.10 | 1413.70 | 1989.40 |

Although this result is developed for M/M/1 queues, it has already been established that it can serve as an approximation for more general; i.e., GI/G/1 queues. We assume that all remaining observations (non-transient) are sampled from the system operating in steady state.

The next problem is to choose a method to compute a confidence interval for the mean. The simplest method is called the batched means method (Fishman, 1979; Law & Kelton, 1984).

**EXAMPLE 4.** Batched Means Method:

- group observations into several batches,
- compute the sample mean of the observations in each batch, and
- computes the confidence interval from these batched means using traditional statistical techniques.

Suppose we have the following 12 observations, after removing initial transient observations:

$$1.2, 3.3, 2.6, 5.1, 4.4, 0.6, 0.0, 1.3, 1.5, 3.7, 3.5, 2.4$$

Suppose we group these into three batches:

|         |                    |      |                    |
|---------|--------------------|------|--------------------|
| Batch 1 | 1.2, 3.3, 2.6, 5.1 | Mean | $\bar{X}_1 = 3.05$ |
| Batch 2 | 4.4, 0.6, 0.0, 1.3 | Mean | $\bar{X}_2 = 1.58$ |
| Batch 3 | 1.5, 3.7, 3.5, 2.4 | Mean | $\bar{X}_3 = 2.78$ |

Now we treat the three batched means as if they are a sequence of independent observations from the same population whose mean we want to estimate.

$$\bar{\bar{X}} = 2.47 \text{ and } S_{\bar{x}} = 0.78$$

A 95 percent confidence interval for the mean is

$$\begin{aligned} \bar{\bar{X}} \pm t_{0.025,2} \frac{S_{\bar{x}}}{\sqrt{3}} &= 2.47 \pm 4.303 \frac{0.78}{\sqrt{3}} \\ &= 2.47 \pm 1.95 \\ &= (0.52, 4.42) \end{aligned}$$

Note that  $t_{\alpha,k}$  is the  $100(1 - \alpha)$  percentage point on a Student's t-distribution with  $k$  degrees of freedom.

### ***Guidelines***

If batch size is sufficiently large, the batched means from separate batches will be approximately uncorrelated. The batch means will be approximately normally distributed when the batch size is sufficiently large. The batch means will also be approximately independent when the batch size is sufficiently large.

Our objective is to select the batch size large enough that the batch means are approximately uncorrelated. At the same time, we must ensure a maximum number of batches is formed. We usually want the

number of batches to be at least 10, and the batch size to be at least 100.

## Alternative

An alternative to the batched means method, is to run several independent replications.

First, we compute the sample mean for each replication. We then use these sample means to compute a confidence interval for the mean. Because each replication starts with the same initial condition, we are guaranteed that the sample means are mutually independent, not just approximately uncorrelated. Each replication must run through the initial transient period, and each must reach steady state. Figures 7 and 8 show the setup and results for 20 runs using the Excel add in @Risk..

We then use a graphical method, such as a time series plot, to judge the adequacy of the initial transient period. Figures 9-12 show the results using Arena Simulation.

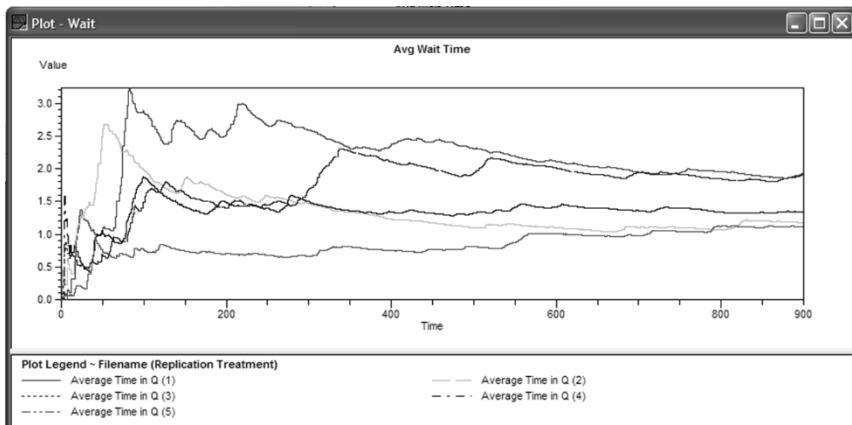
See Queue3.xls at [www.simulation-educators.com/downloads.html](http://www.simulation-educators.com/downloads.html)

|    | A   | B            | C  | D  | E                     | F     | G           | H                               |  |  |
|----|---|--------------|--|--|-----------------------|-------|-------------|---------------------------------|--|--|
| 1  | <b>@Risk Simulation of an M/M/1 Queue</b> |              |  |  |                       |       |             |                                 |  |  |
| 2  |   |              |  |  |                       |       |             |                                 |  |  |
| 3  | <b>Model inputs</b>                       |              | <b>Known mean waiting time, <math>W_q</math></b> |  |                       |       |             |                                 |  |  |
| 4  | Mean service time                         | 0.700        |  | 1.633333   |                       |       |             |                                 |  |  |
| 5  | $\mu$                                     | 1.429        |  | <b>Simulated mean waiting time, <math>W_n</math></b>             |                       |       |             |                                 |  |  |
| 6  | Mean interarrival time                    | 1.000        |  | 1.61118  |                       |       |             |                                 |  |  |
| 7  | $\lambda$                                 | 1.000        |  | <b>Simulated stdev of mean waiting time, <math>S_{Wn}</math></b> |                       |       |             |                                 |  |  |
| 8  |   |              |  | 0.071412   |                       |       |             |                                 |  |  |
| 9  |   |              |  | <b>95% confidence interval for mean waiting time</b>             |                       |       |             |                                 |  |  |
| 10 |   |              |  | Lower  | 1.577758              | Upper | 1.650037305 |                                 |  |  |
| 11 |   |              |  |  |                       |       |             |                                 |  |  |
| 12 | <b>Simulation model</b>                   |              |  |  |                       |       |             |                                 |  |  |
| 13 | Customer number                           | Waiting Time | Service Time                                     | Interarrival Time  |                       |       |             | <b>@Risk Mean Waiting Times</b> |  |  |
| 14 | (n)                                       | ( $W_n$ )    | ( $X_n$ )  | ( $Y_n$ )  | ( $W_n + X_n - Y_n$ ) |       |             |                                 |  |  |
| 15 | 0   | 0.0000       | 0.8120   | 2.8358   | -2.0239               |       |             | Sim 1 1.616574402               |  |  |
| 16 | 1   | 0.0000       | 4.8848   | 0.6112   | 4.2735                |       |             | Sim 2 1.683005505               |  |  |
| 17 | 2   | 4.2735       | 2.0866   | 0.2575   | 6.1026                |       |             | Sim 3 1.518170145               |  |  |
| 18 | 3   | 6.1026       | 0.6387   | 0.2947   | 6.4466                |       |             | Sim 4 1.709264186               |  |  |
| 19 | 4   | 6.4466       | 2.0607   | 0.5655   | 7.9417                |       |             | Sim 5 1.610518539               |  |  |
| 20 | 5   | 7.9417       | 0.5665   | 0.5741   | 7.9341                |       |             | Sim 6 1.502194958               |  |  |
| 21 | 6   | 7.9341       | 0.4788   | 1.5555   | 6.8575                |       |             | Sim 7 1.695268714               |  |  |
| 22 | 7   | 6.8575       | 0.8715   | 1.1199   | 6.6090                |       |             | Sim 8 1.604255537               |  |  |
| 23 | 8   | 6.6090       | 1.4512   | 0.3889   | 7.6713                |       |             | Sim 9 1.681859807               |  |  |
| 24 | 9   | 7.6713       | 0.4505   | 0.3222   | 7.7997                |       |             | Sim 10 1.619694122              |  |  |
| 25 | 10  | 7.7997       | 1.4619   | 1.0398   | 8.2218                |       |             | Sim 11 1.549878574              |  |  |

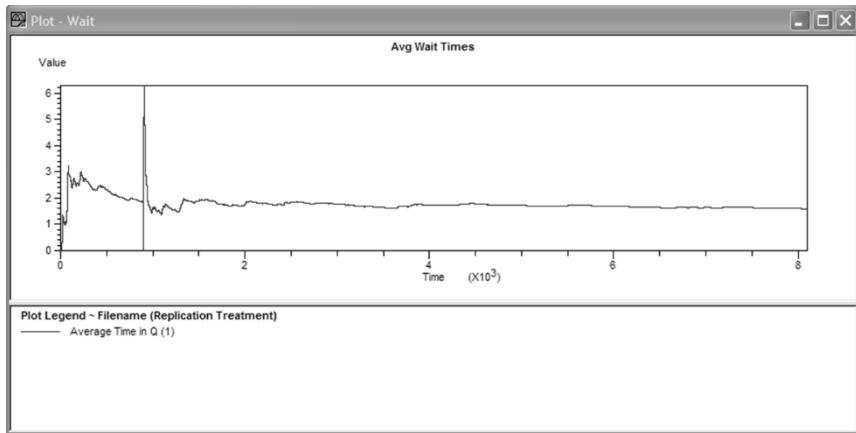
**Figure 7.** An @Risk Simulation of M/M/1 Queue Waiting Times with 20 Runs

|    | A  | B | C      | D |
|----|--|---|--------|---|
| 1  | <i>Results for one-sample analysis for Mean waiting times</i>  |   |        |   |
| 2  |  |   |        |   |
| 3  | <i>Summary measures</i>  |   |        |   |
| 4  | Sample size  |   | 20     |   |
| 5  | Sample mean  |   | 1.611  |   |
| 6  | Sample standard deviation                                      |   | 0.071  |   |
| 7  |  |   |        |   |
| 8  | <i>Test of mean&gt;=1.633333 versus one-tailed alternative</i> |   |        |   |
| 9  | Hypothesized mean  |   | 1.633  |   |
| 10 | Sample mean  |   | 1.611  |   |
| 11 | Std error of mean  |   | 0.016  |   |
| 12 | Degrees of freedom   |   | 19     |   |
| 13 | t-test statistic   |   | -1.387 |   |
| 14 | p-value  |   | 0.091  |   |

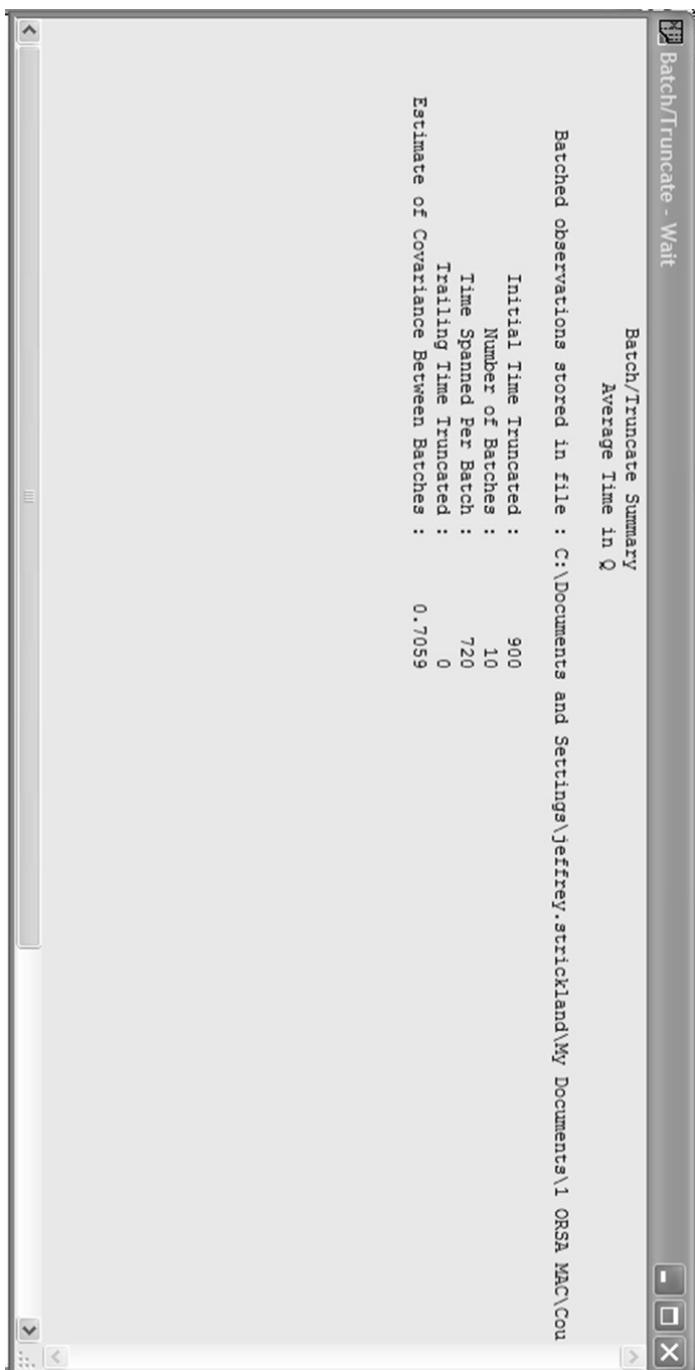
**Figure 8.** An @Risk Simulation of M/M/1 Hypothesis Test on Mean Time with 20 Runs



**Figure 9.** Multiple Runs in Arena's Output Analyzer (5 Sims)



**Figure 10.** Long Run in Arena's Output Analyzer (8100 minutes)



**Figure 11.** Batched Observations in Output Analyzer

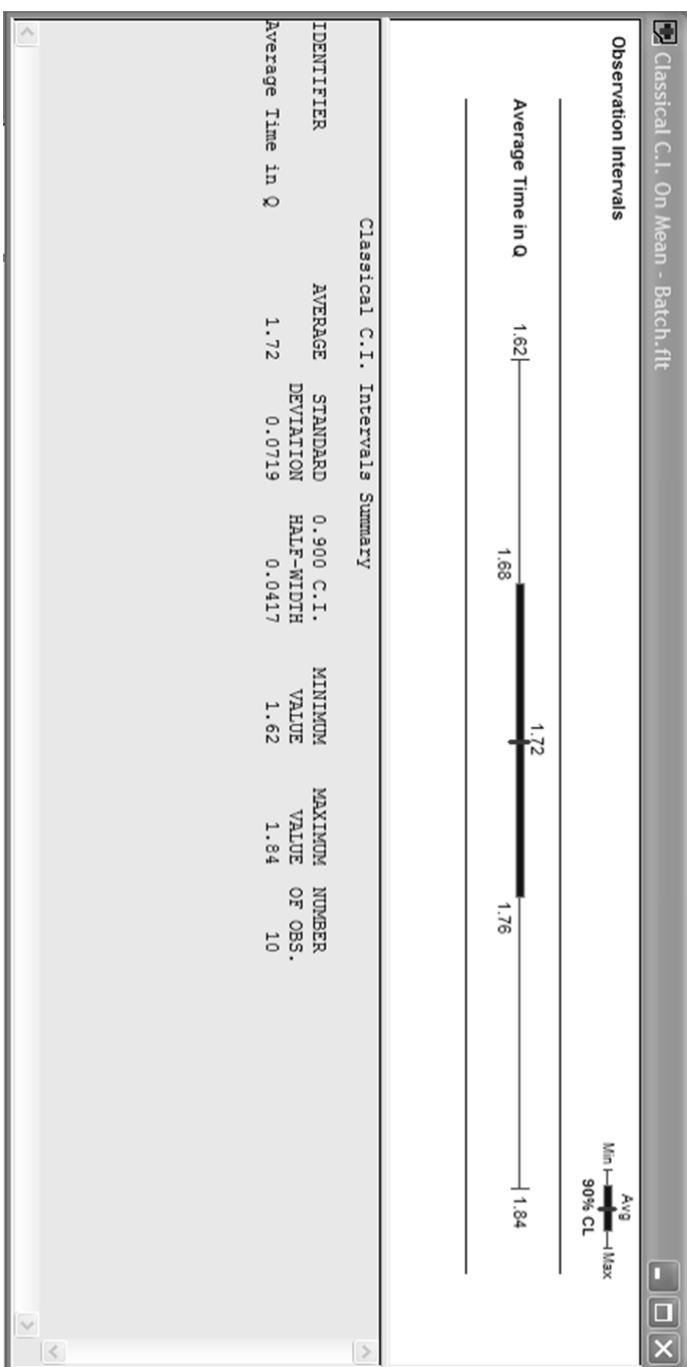


Figure 12. Batched Mean in Output Analyzer

### 16.10.5. Determination of the Desirable Number of Simulation Runs

The two widely used methods for experimentation on simulation models are method of batched means, and independent replications. Intuitively one may say the method of independent replication is superior in producing statistically a “good” estimate for the system's performance measure. In fact, not one method is superior in all cases and it all depends on the traffic intensity  $\rho$ .

After deciding what method is more suitable to apply, the main question is determination of number of runs. That is, at the planning stage of a simulation investigation of the question of number of simulation runs ( $n$ ) is critical.

The confidence level of simulation output drawn from a set of simulation runs depends on the size of data set. The larger the number of runs, the higher is the associated confidence. However, more simulation runs also require more effort and resources for large systems. Thus, the main goal must be in finding the smallest number of simulation runs that will provide the desirable confidence.

Pilot Studies: When the needed statistics for number of simulation runs calculation is not available from existing database, a pilot simulation is needed.

For large pilot simulation runs ( $n$ ), say over 30, the simplest number of runs determinate is:

$$\frac{\left[ \left( Z_{\frac{\alpha}{2}} \right)^2 S^2 \right]}{\delta^2} \quad (16.8)$$

where  $\delta$  is the desirable margin of error (i.e., the absolute error), which is the half-length of the confidence interval with  $100(1 - \alpha)\%$  confidence interval.  $S^2$  is the variance obtained from the pilot run.

One may use the following sample size determinate for a desirable relative error  $\Delta$  in %, which requires an estimate of the coefficient of variation (C.V. in %) from a pilot run with  $n$  over 30:

$$\frac{\left[ \left( \frac{Z_a}{2} \right)^2 (C.V.)^2 \right]}{\Delta^2} \quad (16.9)$$

These sample size determinates could also be used for simulation output estimation of unimodal output populations, with discrete or continuous random variables provided the pilot run size ( $n$ ) is larger than (say) 30.

The aim of applying any one of the above number of runs determinates is at improving your pilot estimates at feasible costs.

## 16.11. Problems

7. For M/M/1 queue, given the mean interarrival time is 15 minutes and the mean service time is 18 minutes:
  - a. What is the known mean waiting time?
  - b. If you are simulating this, what should you do in order to achieve steady state, i.e., get close-enough to the known mean waiting time?
8. Suppose there is an ATM inside your workplace. During the lunch-hour, the interarrival time of ATM users follows an exponential distribution with a mean of 3 minutes. The ATM mean service time is 4 minutes.
  - a. Describe what happens at the TAM during lunch-time.
  - b. Will people be incline to either balk or renege?
  - c. What can be done to fix the problem?



# 17. What is Simulation?

## 17.1. Simulation Software Selection

The vast amount of simulation software available can be overwhelming for the new users. The following are only a random sample of software in the market today:

ACSL, APROS, ARTIFEX, Arena, AutoMod, C++SIM, CSIM, Call\$im, ExtendSim, FluidFlow, GPSS, Gepasi, JavSim, MJX, MedModel, Mesquite, Multiverse, NETWORK, OPNET Modeler, POSES++, Simulat8, Powersim, QUEST, REAL, SHIFT, SIMPLE++, SIMSCRIPT, SLAM, SMPL, SimBank, SimPlusPlus, TIERRA, Witness, SIMNON, VISSIM, and javasim.

There are several things that make an ideal simulation package. Some are properties of the package, such as support, reactivity to bug notification, interface, etc. Some are properties of the user, such as their needs, their level of expertise, etc. For these reasons asking which package is best is a sudden failure of judgment. The first question to ask is for what purpose you need the software? Is it for education, teaching, student-projects or research?

The main question is: What are the important aspects to look for in a package? The answer depends on specific applications. However some general criteria are: Input facilities, processing that allows some programming, Optimization capability, Output facilities, Environment including training and support services, Input-output statistical data analysis capability, and certainly the Cost factor.

You must know which features are appropriate for your situation, although, this is not based on a "Yes" or "No" judgment.

### 17.1.1. SIMSCRIPT II.5

Without computer one cannot perform any realistic dynamic systems simulation.

SIMSCRIPT II.5 is a powerful, free-format, English-like simulation language designed to greatly simplify writing programs for simulation

modelling. Programs written in SIMSCRIPT II.5 are easily read and maintained. They are accurate, efficient, and generate results which are acceptable to users. Unlike other simulation programming languages, SIMSCRIPT II.5 requires no coding in other languages. SIMSCRIPT II.5 has been fully supported for over 33 years. Contributing to the wide acceptance and success of SIMSCRIPT II.5 modelling are:

**DESIGN.** A powerful worldview, consisting of Entities and Processes, provides a natural conceptual framework with which to relate real objects to the model.

**PROGRAMMING.** SIMSCRIPT II.5 is a modern, free-form language with structured programming constructs and all the built-in facilities needed for model development. Model components can be programmed so they clearly reflect the organization and logic of the modeled system. The amount of program needed to model a system is typically 75% less than its FORTRAN or C counterpart.

**DEBUGGER.** A well designed package of program debug facilities is provided. The required tools are available to detect errors in a complex computer program without resorting an error. Simulation status information is provided, and control is optionally transferred to a user program for additional analysis and output.

**EVOLUTION.** This structure allows the model to evolve easily and naturally from simple to detailed formulation as data becomes available. Many modifications, such as the choice of set disciplines and statistics are simply specified in the Preamble.

**DOCUMENTATION.** You get a powerful, English-like language supporting a modular implementation. Because each model component is readable and self-contained, the model documentation is the model listing; it is never obsolete or inaccurate.

#### **EXAMPLE 1.** SimScript II Solving an Analytic Equation

```
'' Solving an analytic equation arising from optimization
```

```
'' of a coherent reliability system with 3 homogeneous  
components
```

```
Preamble  
    Define V as a real 1-dimensional arrays  
    Define I, N as integer variables  
End  
  
Main  
    Open 3 for output, Name = "PROG.OUT"  
    Use 3 for ouTPUT  
    LET N=50  
    Reserve V(*) as N  
    LET V(1)=1.  
    For I = 1 to N-1  
        DO  
            LET U=V(I)  
            LET PPRAM=-9./((1+U)**2) + 9./((2.+U)**2) +  
1./U**2  
            LET V(I+1)=V(I)+PPRAM/I  
        LOOP  
        PRINT 1 LINE WITH V(N) AND PPRAM THUS  
        OPTIMAL RATE IS ****.***** , DERIVATIVE IS ***.*****  
    END
```

The output

```
OPTIMAL RATE IS      0.76350,   DERIVATIVE IS -  
0.00000
```

### 17.1.2. SIMAN

IMAN is a new general purpose SIMulation ANalysis program for modeling combined discrete-continuous systems. The modeling framework of SIMAN allows component models based on three distinct modeling orientations to be combined in a single system model. For discrete change systems either a process or event orientation can be set to describe the model. Continuous change systems are modeled with algebraic, difference, or differential equations. A combination of these orientations can be used to model combined discrete-continuous models.

**DESIGN.** SIMAN is designed around a logical modeling framework in which the simulation program is decomposed into a model frame and an experiment frame.

SIMAN incorporates a number of unique and powerful general purpose modeling constructs which represents a natural evolution and refinement of existing language designs.

SIMAN imbeds within this general purpose framework a set of special purpose constructs which are specifically designed to simplify and enhance the modeling of manufacturing systems. Existing general purpose languages such as GPSS [6] and SLAM [4] lack the special purpose manufacturing features provided by SIMAN. On the other hand, existing special purpose manufacturing languages such as GALS [1] and SPEED [3] are intended for a restricted class of manufacturing systems and are not applicable to systems in general.

**THE SYSTEM MODELING FRAMEWORK.** The SIMAN modeling framework is based on the system theoretic concepts developed by Zeigler [7]. Within this framework, a fundamental distinction is stressed between the system model and the experimental frame. The system model defines the static and dynamic characteristics of the system. The experimental frame defines the experimental conditions under which the model is run to generate specific output data. For a given model, there can be many experimental frames resulting in many sets of output data. By separating the model structure and the experimental frame into two distinct elements, different simulations experiments can be performed by changing only the experimental frame. The system model remains the same.

Given the system model and the experimental frames, the SIMAN simulation program generates output files which record the model state transitions as they occur in simulated time. The data in the output files can then be subjected to various data analyses such as data truncation and compression, and the formatting and display of histograms, plots, tables, etc. Within the SIMAN framework, the data analysis and display function follow the development and running of the simulation program and are completely distinct from it. One output file can be subjected to many different data treatments without re-executing the simulation program. Data treatments can also be applied to sets of output files—this is useful when performing an analysis based on multiple runs of a model or when comparing the system response of two or more models.

## **EXAMPLE 2. TV Inspection and Adjustment.**

The following is example code for a statement model for a simulation of TV inspection and adjustment

```
BEGIN;

10      CREATE:UNC(1,1):MRRK(1);          CREATE ARRIVING TVs
20 INSPECT  QUEUE,1;                  QUEUE FOR INSPECTOR IN FILE: 1
30      SEIZE:ADJUSTOR;                SEIZE AN INSPECTOR
40      DELAY:LN(3,1);                DELAY BY INSPECTOR TIME
50      RELEASE: INSPECTOR;          RELEASE THE INSPECTOR
      BRANCH, 1;
          WITH,.15,ADJUST;
60          WITH,.85,PACKING;        BRANCH TO ADJUST FOR PACKING
70 ADJUST   QUEUE 2;                 QUEUE FOR ADJUSTOR IN FILE 2
80      SEIZE: INSPECTOR;           SEIZE THE ADJUSTOR
90      DELAY:UNIF(3,1);            DELAY BY INSPECTOR TIME
10      RELEASE:ADJUSTER NEXT (INSPECT); RELEASE THE INSPECTOR
0  PACKING  TALLY:1,INIT(1):DISPOSE;    TALLY TIME IN SYSTEM
11
0 END;
```

## **17.2. Animation in Systems Simulation**

Animation in systems simulation is a useful tool. Most graphically based software packages have default animation. This is quite useful for model debugging, validation, and verification. This type of animation comes with little or no additional effort and gives the modeler additional insight into how the model. This type of animation comes with little or no additional effort and gives the modeler additional insight into how the model works. However, it augments the modeling tools available. The more realistic animation presents qualities which intend to be useful to the decision-maker in implementing the developed simulation model. There are also, good model management tools. Some tools have been developed which combined a database with simulation to store models, data, results, and animations. However, there is not one product that provides all of those capabilities.

## **17.3. System Dynamics and Discrete Event Simulation**

The modeling techniques used by system dynamics and discrete event simulations are often different at two levels: The modeler way of representing systems might be different, the underlying simulators' algorithms are also different. Each technique is well tuned to the purpose it is intended. However, one may use a discrete event approach to do system dynamics and vice versa.

Traditionally, the most important distinction is the purpose of the modeling. The discrete event approach is to find, e.g., how many resources the decision maker needs such as how many trucks, and how to arrange the resources to avoid bottlenecks, i.e., excessive of waiting lines, waiting times, or inventories. While the system dynamics approach is to prescribe for the decision making to, e.g., timely respond to any changes, and how to change the physical structure, e.g., physical shipping delay time, so that inventories, sales, production, etc.

System dynamics is the rigorous study of problems in system behavior using the principles of feedback, dynamics and simulation. In more words system dynamics is characterized by:

- Searching for useful solutions to real problems, especially in social systems (businesses, schools, governments,...) and the environment.
- Using computer simulation models to understand and improve such systems.
- Basing the simulation models on mental models, qualitative knowledge and numerical information.
- Using methods and insights from feedback control engineering and other scientific disciplines to assess and improve the quality of models.
- Seeking improved ways to translate scientific results into achieved implemented improvement.
- Systems dynamics approach looks at systems at a very high level so is more suited to strategic analysis. Discrete event approach may look at subsystems for a detailed analysis and is more suited, e.g., to process re-engineering problems.
- Systems dynamics is indicative, i.e., helps us understand the

direction and magnitude of effects (i.e., where in the system do we need to make the changes), whereas discrete event approach is predictive (i.e., how many resources do we need to achieve a certain goal of throughout).

- Systems dynamics analysis is continuous in time and it uses mostly deterministic analysis, whereas discrete event process deals with analysis in a specific time horizon and uses stochastic analysis.

Some interesting and useful areas of system dynamics modeling approach are:

- Short-term and long term forecasting of agricultural produce with special reference to field crops and perennial fruits such as grapes, which have significant processing sectors of different proportions of total output where both demand and supply side perspectives are being considered.
- Long term relationship between the financial statements of balance sheet, income statement and cash flow statement balanced against scenarios of the stock market's need to seek a stable/growing share price combined with a satisfactory dividend and related return on shareholder funds policy.
- Managerial applications include the development and evaluation of short-term and long-term strategic plans, budget analysis and assessment, business audits and benchmarking.

A modeler must consider both as complementary tools to each other. Systems dynamic to look at the high level problem and identify areas which need more detailed analysis. Then, use discrete event modeling tools to analyze (and predict) the specific areas of interest.

#### 17.4. Parallel and Distributed Simulation

The increasing size of the systems and designs requires more efficient simulation strategies to accelerate the simulation process. Parallel and distributed simulation approaches seem to be a promising approach in this direction. Current topics under extensive research are:

- Synchronization, scheduling, memory management, randomized

and reactive/adaptive algorithms, partitioning and load balancing.

- Synchronization in multi-user distributed simulation, virtual reality environments, HLA, and interoperability.
- System modeling for parallel simulation, specification, re-use of models/code, and parallelizing existing simulations.
- Language and implementation issues, models of parallel simulation, execution environments, and libraries.
- Theoretical and empirical studies, prediction and analysis, cost models, benchmarks, and comparative studies.
- Computer architectures, VLSI, telecommunication networks, manufacturing, dynamic systems, and biological/social systems.
- Web based distributed simulation such as multimedia and real time applications, fault tolerance, implementation issues, use of Java, and CORBA.

## 17.5. “What-if” Analysis Techniques

### 17.5.1. Introduction

The simulation models are often subject to errors caused by the estimated parameter(s) of underlying input distribution function. “What-if” analysis is needed to establish confidence with respect to small changes in the parameters of the input distributions. However the direct approach to “what-if” analysis requires a separate simulation run for each input value. Since this is often inhibited by cost, as an alternative, what people are basically doing in practice is to plot results and use a simple linear interpolation/extrapolation. This section presents some simulation-based techniques that utilize the current information for estimating performance function for several scenarios without any additional simulation runs.

Simulation continues to be the primary method by which system analysts obtain information about analysis of complex stochastic systems. In almost all simulation models, an expected value can express the system's performance. Consider a system with continuous parameter  $v \in V$ , where  $V$  is the feasible region. Let

$$J(v) = E_{(Y|v)}[Z(Y)] \quad (17.1)$$

be the steady state expected performance measure, where  $Y$  is a random vector with known probability density function (pdf),  $f(y; \nu)$  depends on  $\nu$ , and  $Z$  is the performance measure.

In discrete event systems, Monte Carlo simulation is usually needed to estimate  $J(\nu)$  for a given value  $\nu$ . By the law of large numbers

$$J(\nu) = \frac{1}{n} \sum_{i=1}^n Z(y_{-i}) \quad (17.2)$$

where  $y_i, i = 1, 2, \dots, n$  are independent, identically distributed, random vector realizations of  $Y$  from  $f(y; \nu)$ , and  $n$  is the number of independent replications. This is an unbiased estimator for  $J(\nu)$  and converges to  $J(\nu)$  by law of large numbers.

There are strong motivations for estimating the expected performance measure  $J(\nu)$  for a small change in  $\nu$  to  $\nu + \delta\nu$ , that is to solve the so-called “what if” problem.

The simulationist must meet managerial demands to consider model validation and cope with uncertainty in the estimation of  $\nu$ . Adaptation of a model to new environments also requires an adjustment in  $\nu$ .

An obvious solution to the “what if” problem is the Crude Monte Carlo (CMC) method, which estimates  $J(\nu + \delta\nu)$  for each  $\nu$  separately by rerunning the system for each  $\nu + \delta\nu$ . Therefore costs in CPU time can be prohibitive. The use of simulation as a tool to design complex computer stochastic systems is often inhibited by cost. Extensive simulation is needed to estimate performance measures for changes in the input parameters. As an alternative, what people are basically doing in practice is to plot results of a few simulation runs and use a simple linear interpolation/extrapolation.

In this section we consider the “What-if” analysis problem by extending the information obtained from a single run at the nominal value of parameter  $\nu$  to the closed neighborhood. We also present the use of results from runs at two or more points over the intervening interval. We refer to the former as extrapolation and the latter as interpolation

by simulation. The results are obtained by some *computational cost* as opposed to *simulation cost*. Therefore, the proposed techniques are for estimating a performance measure at multiple settings from a simulation at a nominal value.

### 17.5.2. Likelihood Ratio (LR) Method

A model based on Radon-Nikodym theorem to estimate  $J(\nu + \delta\nu)$  for stochastic systems in a single run is as follows:

$$J(\nu + \delta\nu) = E_{(Y|\nu + \delta\nu)}[Z(Y)] = E_{(Y|\nu)}[Z(Y). W]$$

where the likelihood ratio  $W$  is:

$$W = \frac{f(y; \nu + \delta\nu)}{f(y; \nu)} \quad (17.3)$$

adjusts the sample path, provided  $f(y; \nu)$  does not vanish. Notice that by this change of probability space, we are using the common realization as  $J(\nu)$ .

The generated random vector  $y$  is roughly representative of  $Y$ , with  $f(\nu)$ . Each of these random observations, could also hypothetically came from  $f(\nu + \delta\nu)$ .  $W$  weights the observations according to this phenomenon.

Therefore, from (17.2) and (17.3) the "What-if" estimate is:

$$J(\nu) = \frac{1}{n} \sum_{i=1}^n Z(y_i). W_i \quad (17.4)$$

which is based on only one sample path of the system with parameter  $\nu$  and the simulation for the system with  $\nu + \delta\nu$  is not required.

Unfortunately LR produces a larger variance compared with CMC. However, since  $E(W) = 1$ , the following variance reduction techniques (VRT) may improve the estimate.

$$J(v) = \frac{\sum_{i=1}^n Z(y_i) \cdot W_i}{\sum_{i=1}^n W_i} \quad (17.5)$$

### 17.5.3. Exponential Tangential in Expectation Method

In the statistical literature the efficient score function is defined to be the gradient

$$S(y) = \delta \ln \frac{f(y; v)}{\delta v} \quad (17.6)$$

We consider the exponential (approximation) model for  $J(v + \delta v)$  in a first derivative neighborhood of  $v$  by:

$$J(v + \delta v) = \frac{E[Z(y) \cdot \exp(\delta v S(y))]}{E[\exp(\delta S(y))]} \quad (17.7)$$

Now we are able to estimate  $J(v + \delta v)$  based on  $n$  independent replications as follows:

$$J(v + \delta v) = \frac{\sum A_i}{\sum B_i} \quad (17.8)$$

where,

$$A_i = Z(y_i) \cdot \exp[\delta v \cdot S(y_i)]$$

and

$$B_i \exp[\delta v \cdot S(y_i)]. \quad (17.7a)$$

Taylor Expansion of Response Function

The following linear Taylor model can be used as an auxiliary model.

$$J(v + \delta v) = J(v) + d v. J'(v) + \dots, \quad (17.9)$$

where the prime denotes derivative. This metamodel approximates  $J(v + \delta v)$  for small  $\delta v$ . For this estimate, we need to estimate the

nominal  $J(v)$  and its first derivative. Traditionally, this derivative is estimated by crude Monte Carlo; i.e., finite difference, which requires rerunning the simulation model. Methods which yield enhanced efficiency and accuracy in estimating, at little additional cost, are of great value.

There are few ways to obtain efficiently the derivatives of the output with respect to an input parameter as presented earlier on this site. The most straightforward method is the Score Function (SF). The SF approach is the major method for estimating the performance measure and its derivative, while observing only a single sample path from the underlying system. The basic idea of SF is that the derivative of the performance function,  $J'(v)$ , is expressed as expectation with respect to the same distribution as the performance measure itself.

Therefore, for example, using the estimated values of  $J(v)$  and its derivative  $J'(v)$ , the estimated  $J(v + \delta v)$  is:

$$J(v + \delta v) = J(v) + \delta v J'(v_0) \quad (17.10)$$

with variance

$$\begin{aligned} \text{Var}[J(v + \delta v)] &= \text{Var}[J(v)] + (\delta v)^2 \text{Var}[J'(v)] \\ &\quad + 2\delta v \text{Cov}[J(v), J'(v)]. \end{aligned} \quad (17.11)$$

This variation is needed for constructing a confidence interval for the perturbed estimate.

#### 17.5.4. Interpolation Techniques

Given two points,  $v1$  and  $v2$  (scalars only) "sufficiently" close, one may simulate at these two points then interpolates for any desired points in between. Assuming the given  $v1$  and  $v2$  are sufficiently close and looks for the "best" linear interpolation in the sense of minimum error on the interval. Clearly,

$$J(v) = E_{(Y|v)}[Z(Y)] = \varphi \cdot E_{(Y|v)}[Z(Y)] + (1 - \varphi) \cdot E(Y|v) [Z(Y)]$$

Similar to the Likelihood Ratio approach, this can be written as:

$$\begin{aligned}
 J(v) &= E_{(Y|v)} [Z(Y)] \\
 &= \varphi \cdot E_{(Y|v_1)} [Z(Y). W_1] + (1 - \varphi) \\
 &\quad \cdot E_{(Y|v_2)} [Z(Y). W_2]
 \end{aligned} \tag{17.12}$$

where the likelihood ratios  $W_1$  and  $W_2$  are  $W_1 = \frac{f(y; v)}{f(y; v_1)}$  and  $W_2 = \frac{f(y; v)}{f(y; v_2)}$ , respectively.

One obvious choice is  $f = \frac{f(y; v_1)}{[f(y; v_1) + f(y; v_2)]}$ . This method can easily be extended to  $k$ -point interpolation.

For 2-point interpolation, if we let  $f$  to be constant within the interval  $[0, 1]$ , then the linear interpolated "what-if" estimated value is:

$$J(v) = \varphi \cdot J_1 + (1 - \varphi) \cdot J_2 \tag{17.13}$$

where the two estimates on the RHS of are two independent Likelihood Ratio extrapolations using the two end-points.

We define  $\varphi^*$  as the  $\varphi$  in this convex combination with the minimum error in the estimate. That is, it minimizes

$$\text{Var}[J(v)] = \varphi^2 \text{Var}[J_1] + (1 - \varphi)^2 \text{Var}[J_2] \tag{17.14}$$

By the first order necessary and sufficient conditions, the optimal  $\varphi^*$  is:

$$\varphi^* = \frac{\text{Var}[J_2]}{\{\text{Var}[J_1] + \text{Var}[J_2]\}} \tag{17.15}$$

Thus, with (17.13) and (17.15) the "best linear" interpolation for any point in interval  $[v_1, v_2]$  is:

$$J(v) = \varphi^* \cdot J_1 + (1 - \varphi^*) \cdot J_2 \tag{17.16}$$

which is the optimal interpolation in the sense of having minimum variance.

### 17.5.5. Conclusions

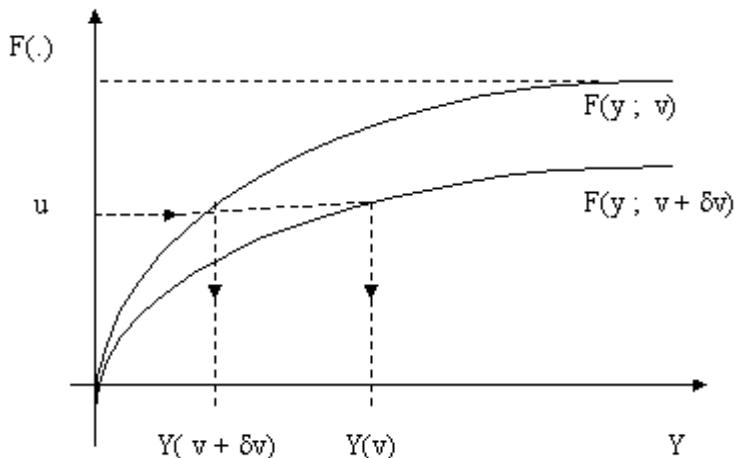
Estimating system performance for several scenarios via simulation generally requires a separate simulation run for each scenario. In some very special cases, such as the exponential density  $f(y; \nu) = \nu e^{-\nu y}$ , one could have obtained the perturbed estimate using Perturbation Analysis directly as follow. Clearly, one can generate random variate  $Y$  by using the following inverse transformation:

$$Y_i = \left( \frac{1}{\nu} \right) \ln \left( \frac{1}{U_i} \right) \quad (17.16)$$

where  $\ln$  is the natural logarithm and  $U_i$  is a random number distributed Uniformly  $[0,1]$ . In the case of perturbed  $\nu$ , the counterpart realization using the same  $U_i$  is

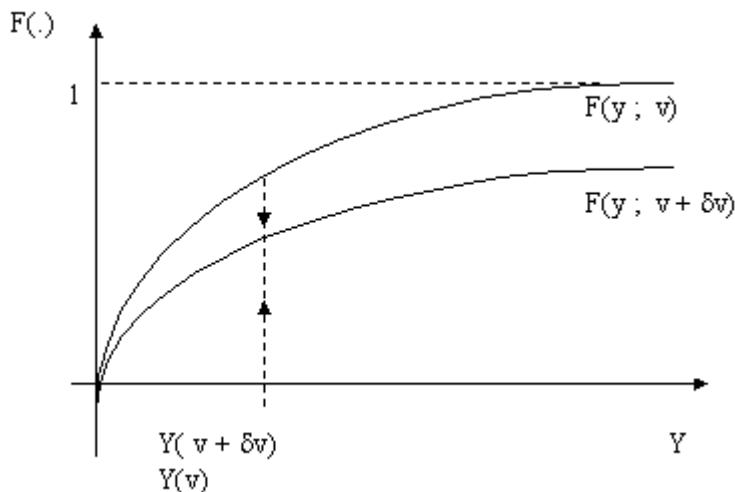
$$Y_i = \left[ \frac{1}{\nu + d\nu} \right] \ln \left( \frac{1}{U_i} \right). \quad (17.17)$$

Clearly, this single run approach is limited, since the inverse transformation is not always available in closed form. Figure 1 illustrates the Perturbation Analysis Method:



**Figure 1.** Parameter (s) Perturbation via Perturbation Analysis Method

Since the Perturbation Analysis Approach has this serious limitation, for this reason, we presented some techniques for estimating performance for several scenarios using a single-sample path, such as the Likelihood Ratio method, which is illustrated in Figure 2.



**Figure 2.** Parameter (s) Perturbation via Likelihood Ratio Method

## 17.6. Simulation using R

### EXAMPLE 3. Monte Carlo Simulation in R

In this example we build a Monte Carlo experiment to compare these approaches. Suppose we have 100 observations with  $x = 0$  and 100 with  $x = 1$ , and suppose that the  $Pr(Y = 1|X = 0) = 0.001$ , while the  $Pr(Y = 1|X = 1) = 0.05$ . Thus the true odds ratio is  $(0.05/0.95)/(0.001/0.999) = 52.8$  and the log odds ratio we want to find is 3.96. But we will rarely observe any  $y=1$  when  $x=0$ . Which of these approaches is most likely to give us acceptable results?

Note that in all of the Markov Chain Monte Carlo (MCMC) analyses we use only 6000 iterations, which is likely too few to trust in practice.

The code is long enough here that we annotate within rather than write much text around the code.

The R procedure used accept the events/trials syntax, so we'll generate example data sets as two observations of binomial random variates with the probabilities noted above.

We want the “weights” version of the data for the `glm()` and `logistf()` functions and need the events/trials syntax for the `elrm()` function and the expanded (one row per observation) version for the `MCMClogit()` function. The `sapply()` function serves to replicate the analysis for each simulated data set. Finally, rather than spelunking through the odds trace output to find the parameter estimates, we used the `str()` function to figure out where they are stored in the output objects and indexes (rather than data set options) to pull out the one estimate we need.

```
# make sure the needed packages are present
require(logistf)
require(elrm)
require(MCMCpack)
# the runlogist() function generates a dataset and runs each
analysis
# the parameter "trial" keeps track of which time we're
calling runlogist()
runlogist = function(trial) {
```

```

# the result vector will hold the estimates temporarily
result = matrix(0,4)
# generate the number of events once
events.0 =rbinom(1,100, .001) # for x = 0
events.1 = rbinom(1,100, .05) # for x = 1
# following for glm and logistf "weights" format
xw = c(0,0,1,1)
yw = c(0,1,0,1)
ww = c(100 - events.0, events.0, 100 - events.1,events.1)
# run the glm and logistf, grab the estimates, and stick
# them into the results vector
result[1]      =      glm(yw      ~      xw,      weights=ww,
binomial)$coefficients[2]
result[2] = logistf(yw ~ xw, weights=ww)$coefficients[2]
# elrm() needs a data frame in the events/trials syntax
elrmdata      =      data.frame(events=c(events.0,events.1),      x
=c(0,1), trials = c(100,100))
# run it and grab the estimate
result[3]=elrm(events/trials ~ x, interest = ~ x, iter =
6000, burnIn = 1000, data = elrmdata, r = 2)$coeffs
# MCMClogit() needs expanded data
x = c(rep(0,100), rep(1,100))
y = c(rep(0,100-events.0), rep(1,events.0), rep(0, 100-
events.1), rep(1, events.1))
# run it and grab the mean of the MCMC posteriors
result[4] = summary(MCMClogit(y~as.factor(x), burnin=1000,
mcmc=6000, b0=0, B0=.04, seed = list(c(781306, 78632467,
364981736, 6545634, 7654654, 4584),trial)))$statistics[2,1]
# send back the four estimates, plus the number of events
# when x=0 and x=1
return(c(trial, events.0, events.1, result))}
```

Note the construction of the `seed=` option to the `MCMClogit()` function. This allows a different seed in every call without actually using sequential seeds.

Now we're ready to call the function repeatedly. We'll do that with the `sapply()` function, but we need to nest that inside a `t()` function call to get the estimates to appear as columns rather than rows, and we'll also make it a data frame in the same command. Note that the

parameters we change within the `sapply()` function are merely a list of trial numbers. Finally, we'll add descriptive names for the columns with the `names()` function (section 1.3.4)

```
res2 = as.data.frame(t(sapply(1:10, runlogist)))
names(res2) <- c("trial", "events.0", "events.1", "glm",
"firth", "exact-ish", "MCMC")
head(res2)
mean(res2[,4:7], na.rm=TRUE)

  trial events.0 events.1      glm      firth exact-ish     MCMC
1     1        0    7 18.72447 2.780246 2.655559 3.736947
2     2        0    8 17.86881 2.916162 2.438431 3.881940
3     3        0    5 18.36672 2.448927 1.981845 3.555314
4     4        0    7 18.72447 2.780246 2.629342 3.840758
5     5        0    5 18.36672 2.448927 2.061048 3.530652
6     6        0    6 18.55962 2.626507 1.930918 3.647824
```

The ordinary logistic estimates are entirely implausible, while the three alternate approaches are more acceptable. The MCMC result has the least bias, but it's unclear to what degree this is a happy coincidence between the odds ratio and the prior precision. The Firth approach appears to be less biased than the exact logistic regression.

In most Monte Carlo experimental settings, one would also be interested in examining the confidence limits for the parameter estimates. Notes and code for doing this can be found here. In a later entry we'll consider plots for the results generated above. As a final note, there are few combinations of event numbers with any mass worth considering. One could compute the probability of each of these and the associated parameter estimates, deriving a more analytic answer to the question. However, this would be difficult to replicate for arbitrary event probabilities and  $N$ s, and very awkward for continuous covariates, while the above approach could be extended with trivial ease.

#### EXAMPLE 4. Simulation of logistic regression output

In R we begin by assigning parameter values for the model. We then generate 1,000 random normal variates (section 1.10.5), calculating the

linear predictor and expected value for each, and then testing vectorwise (section 1.11.2) against 1,000 random uniforms (1.10.3).

```
intercept = 0
beta = 0.5
xtest = rnorm(1000,1,1)
linpred = intercept + xtest*beta
prob = exp(linpred)/(1 + exp(linpred))
runis = runif(1000,0,1)
ytest = ifelse(runis < prob,1,0)
ytest
[1] 0 1 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1 1
0 1 0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1 1 0
[49] 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 1 1 1 1 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1
[97] 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 0
1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 1
.
.
.
[913] 0 0 0 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 1 1 0
0 1 1 0 1 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1
[961] 1 0 1 0 1 0 0 1 1 1 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 0
0 1 0 1 1 0 0 1 1 1 1 0 1 0
```

### EXAMPLE 5. Let's Make a Deal

The Monty Hall problem illustrates a simple setting where intuition often leads to a solution different from formal reasoning. The situation is based on the game show Let's Make a Deal. First, Monty puts a prize behind one of three doors. Then the player chooses a door. Next, (without moving the prize) Monty opens an unselected door, revealing that the prize is not behind it. The player may then switch to the other non-selected door. Should the player switch?

Many people see that there are now two doors to choose between, and feel that since Monty can always open a non-prize door, there's still equal probability for each door. If that were the case, the player might as well keep the original door.

This approach is so attractive that when Marilyn vos Savant asserted that the player should switch (in her Parade magazine column), there were reportedly 10,000 letters asserting she was wrong. There was also an article published in *The American Statistician*. As reported on the Wikipedia page, out of 228 subjects in one study, only 13% chose to switch (Granberg and Brown, *Personality and Social Psychology Bulletin* 21(7): 711-729). In her book, *The Power of Logical Thinking*, vos Savant quotes cognitive psychologist Massimo Piattelli-Palmarini as saying "... no other statistical puzzle comes so close to fooling all the people all the time" and "that even Nobel physicists systematically give the wrong answer, and that they insist on it, and they are ready to berate in print those who propose the right answer."

One correct intuitive route is to observe that Monty's door is fixed. The probability that the player has the right door is  $1/3$  before Monty opens the non-prize door, and remains  $1/3$  after that door is open. This means that the probability the prize is behind one of the other doors is  $2/3$ , both before and after Monty opens the non-prize door. After Monty opens the non-prize door, the player gets a  $2/3$  chance of winning by switching to the remaining door. If the player wants to win, they should switch doors.

The excellent Wikipedia entry referenced above provides additional intuitive tools, as well as variants and history.

One way to prove to yourself that switch is best is through simulation. In fact, even deciding how to code the problem may be enough to convince yourself to switch.

In R, we write two functions. In one, Monty opens a door, choosing at random among the non-chosen doors if the initial choice was correct, or choosing the one non-selected non-prize door if the initial choice was wrong. The other function returns the door chosen by swapping. We use the `sample()` function (section 1.5.2) to randomly pick one value. We then use these functions on each trial with the `apply()` statement (section B.5.3).

```

numsim = 100000
doors = 1:3
opendoor = function(x) { if (x[1]==x[2]) return(sample(doors[-c(x[1])], 1)) else return(doors[-c(x[1],x[2])])}
swapdoor = function(x) { return(doors[-c(x[1], x[2])]) }
winner = sample(doors, numsim, replace=TRUE)
choice = sample(doors, numsim, replace=TRUE)
open = apply(cbind(winner, choice), 1, opendoor)
newchoice = apply(cbind(open, choice), 1, swapdoor)

> cat("without switching, won      ", round(sum(winner==choice)/numsim*100,1), " percent of the time.\n", sep="")
without switching, won 33.4 percent of the time.
> cat("with switching, won      ", round(sum(winner==newchoice)/numsim*100,1), " percent of the time.\n", sep="")
with switching, won 66.6 percent of the time.

```

## 17.7. Simulation using Excel

With Excel's random number generator, **RAND()**, and it's distribution functions, we can build simulation models. The simulation depicted in Figure 3 below represent the operation of a machine over a 52-week period. The machine can be in one of four possible states: Excellent, Good, Average or Bad. We must run a simulation to determine the economic impact of replacing the machine either when it is Bad or Average. Table 1 shows the weekly profit values for the machine when it is in one of the four states.

**Table 1.** Weekly Profit per Machine State

| Machine State | Weekly Profit |
|---------------|---------------|
| Excellent     | \$90.00       |
| Good          | \$70.00       |
| Average       | \$50.00       |
| Bad           | (\$60.00)     |

We also know the probabilities of the machine being in different states and changing states (i.e. from Excellent to Good). These are given in

Table 2. In this scenario, the machine in the Excellent state can only move one state down. For instance the machine in the Excellent state can either stay in that state or change to Good—it cannot go from Excellent to Average or Bad. However, the machine in the Good states can change to Bad, although the probability is low (0.05).

|           | Excellent | Good | Average | Bad  |
|-----------|-----------|------|---------|------|
| Excellent | 0.70      |      |         |      |
| Good      | 0.30      | 0.75 |         |      |
| Average   |           | 0.20 | 0.60    |      |
| Bad       |           | 0.05 | 0.40    | 1.00 |

The process that we are looking at is a Markov Transition, and we can take the given probabilities to create a Markov Transition matrix.

$$T = \begin{bmatrix} 0.70 & 0.00 & 0.00 & 0.00 \\ 0.30 & 0.75 & 0.00 & 0.00 \\ 0.00 & 0.20 & 0.60 & 0.00 \\ 0.00 & 0.05 & 0.40 & 1.00 \end{bmatrix}$$

This situation is modeled in Figure 4. We are running three simulation of the model simultaneously, but we could run more. We do this for the reason we have mentioned regarding autocorrelation.

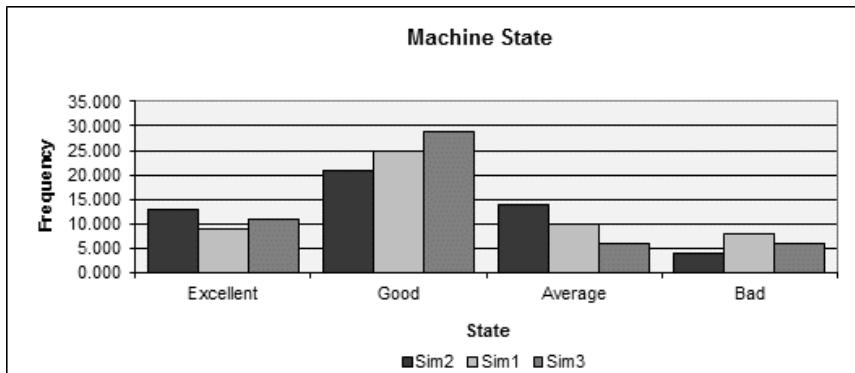
The keys to running this simulation are the values in the transition matrix and the Excel function RAND(). As we have mentioned, RAND() generates pseudo-random numbers from an Uniform(0,1) distribution. Recall that the RAND() function is generating random numbers over a long cycle, so a 52-week (times 3 simulations) is no problem. We also use a decision simulation table (simtable for short) and give the machine replacement states equal chances of occurring,. That is, we replace it at machine state 3 (Average) 50% of the time and at machine state 4 (Bad) 50% of the time. We take the two (ideally more) and use the batched means or regenerative mean methods for computing the grand mean.

Since the Excel workbook can be download from our website, we will not go into the details of building it here. We show this to demonstrate

Excel's ability to run simulations that a little more complex than a Monte Carlo simulation.

|    | A            | B            | C            | D            | E                 | F                | G           | H           | I           | J           |
|----|--------------|--------------|--------------|--------------|-------------------|------------------|-------------|-------------|-------------|-------------|
| 1  | Total Profit | Total Profit | Total Profit | Total Profit |                   |                  |             |             |             |             |
| 2  | Sim1         | Sim2         | Sim3         | GRAND        |                   |                  |             |             |             |             |
| 3  | WeeklyProfit | Total Profit | Total Profit | Total Profit | Avg Cycle Length  | 8.6666667        | 8.6666667   | 6.5         | 7.9444444   |             |
| 4  | \$ 90.00     | \$ 990.00    | \$ 1,260.00  | \$ 1,260.00  | Avg Weekly Profit | \$ 52.75         | \$ 56.15    | \$ 45.71    | 51.537743   |             |
| 5  | \$ 70.00     | \$ 1,330.00  | \$ 1,470.00  | \$ 1,540.00  | Stdev             | 398.32792        | 419.43434   | 402.20197   | 406.65474   |             |
| 6  | \$ 50.00     | \$ 800.00    | \$ 550.00    | \$ 400.00    | lower 95% CI      | \$ (57.64)       | \$ (57.85)  | \$ (89.63)  | -68.373867  |             |
| 7  | \$ (60.00)   | \$ (360.00)  | \$ (360.00)  | \$ (480.00)  | upper 95% CI      | \$ 163.13        | \$ 170.16   | \$ 181.06   | 171.44935   |             |
| 8  |              | \$ 2,760.00  | \$ 2,920.00  | \$ 2,720.00  |                   |                  |             |             |             |             |
| 9  |              |              |              |              |                   |                  |             |             |             |             |
| 10 |              |              |              |              | Total Profit      |                  |             |             |             |             |
| 11 | Frequency    |              |              |              | Sim1              | Sim2             | Sim3        | GRAND       |             |             |
| 12 |              | Sim1         | Sim2         | Sim3         | GRAND             | Mean             | \$ 2,760.00 | \$ 2,920.00 | \$ 2,720.00 | \$ 2,800.00 |
| 13 | Excellent    | 11.000       | 14.000       | 14.000       | 13.000            | Stdev            | \$ 1,950.91 | \$ 2,064.04 | \$ 1,922.62 | \$ 1,979.19 |
| 14 | Good         | 19.000       | 21.000       | 22.000       | 20.667            | 95%CI Lower      | \$ 2,686.47 | \$ 2,842.23 | \$ 2,647.56 | \$ 2,725.42 |
| 15 | Average      | 16.000       | 11.000       | 8.000        | 11.667            | 95%CI Upper      | \$ 2,833.53 | \$ 2,997.77 | \$ 2,792.44 | \$ 2,874.58 |
| 16 | Bad          | 6.000        | 6.000        | 8.000        | 6.667             | Avg Cycle Length | 8.67        | 8.67        | 6.50        | \$ 7.94     |
| 17 |              |              |              |              |                   |                  |             |             |             |             |
| 18 |              |              |              |              | Sim1              | Sim2             | Sim3        | GRAND       |             |             |
| 19 |              |              |              |              | Avg Cycle Length  | 8.67             | 8.67        | 6.50        | 7.94        |             |
| 20 |              |              |              |              | Avg Weekly Profit | \$ 52.75         | \$ 56.15    | \$ 45.71    | 51.54       |             |
| 21 |              |              |              |              | Stdev             | 398.33           | 419.43      | 45.71       | 287.83      |             |
| 22 |              |              |              |              | lower 95% CI      | \$ 43.00         | \$ 43.00    | \$ 28.90    | 38.30       |             |
| 23 |              |              |              |              | upper 95% CI      | \$ 69.31         | \$ 69.31    | \$ 62.53    | 67.05       |             |

**Figure 3.** Machine Maintenance Simulation showing the metric for three runs and their batched mean for total profit, with confidence intervals, average cycle length, and average machine state.



**Figure 4.** The spreadsheet model may be built with the following steps.:.

Enter the input where indicated in Figure 5, including the Transition Matrix, Machine state probabilities, starting state and simtable.

In cell B19 under Random enter =RAND() and copy it down for 52 time periods (weeks).

In cell C19 type =\$E\$8, which is the starting or initial state. The \$ in front of "E" fixes the column to keep it from changing and the \$ in front of "8"

fixed the row. A shortcut is to type E8 and then press the function key “F4”.

|    | A                      | B         | C                | D                         | E                    | F                       |
|----|------------------------|-----------|------------------|---------------------------|----------------------|-------------------------|
| 1  | Machine Maintenance    |           |                  |                           |                      |                         |
| 2  |                        |           |                  |                           |                      |                         |
| 3  | Inputs                 |           |                  |                           |                      | RiskSimTable            |
| 4  | Machine Condition      | Excellent | Weekly Profit    | Scenarios                 | 3                    |                         |
| 5  |                        | Good      | 90               | Replace bad machines      | 4                    |                         |
| 6  |                        | Average   | 70               | Replace average & bad     |                      |                         |
| 7  |                        | Bad       | 50               |                           |                      |                         |
| 8  |                        |           | -60              | Start State               |                      |                         |
| 9  |                        |           |                  | (replacement cost)        | 2                    |                         |
| 10 | Transition probability |           |                  |                           |                      |                         |
| 11 |                        | Excellent | Good             | Average                   | Bad                  | Code                    |
| 12 | Excellent              | 0.7       | 0                | 0                         | 0                    | 1                       |
| 13 | Good                   | 0.3       | 0.75             | 0                         | 0                    | 2                       |
| 14 | Average                | 0         | 0.2              | 0.6                       | 0                    | 3                       |
| 15 | Bad                    | 0         | 0.05             | 0.4                       | 1                    | 4                       |
| 16 |                        |           |                  |                           |                      |                         |
| 17 | Simulation             |           |                  |                           |                      |                         |
| 18 | Week                   | Random    | Start State      | Excellent                 | Good                 | Average                 |
| 19 | 1                      | =RAND()   | =\\$E\$8         | =VLOOKUP(B19,Excellent,2) | =VLOOKUP(B19,Good,2) | =VLOOKUP(B19,Average,2) |
| 20 | 2                      | =RAND()   | =IF(G19=4,1,G19) | =VLOOKUP(B20,Excellent,2) | =VLOOKUP(B20,Good,2) | =VLOOKUP(B20,Average,2) |
| 21 | 3                      | =RAND()   | =IF(G20=4,1,G20) | =VLOOKUP(B21,Excellent,2) | =VLOOKUP(B21,Good,2) | =VLOOKUP(B21,Average,2) |
| 22 | 4                      | =RAND()   | =IF(G21=4,1,G21) | =VLOOKUP(B22,Excellent,2) | =VLOOKUP(B22,Good,2) | =VLOOKUP(B22,Average,2) |
| 23 | 5                      | =RAND()   | =IF(G22=4,1,G22) | =VLOOKUP(B23,Excellent,2) | =VLOOKUP(B23,Good,2) | =VLOOKUP(B23,Average,2) |

**Figure 5.** In cell C20 type =IF(G19=4,1,G19), which tell the model to return to the prior state when the stat turns to Bad=4.

In cell D19 we enter the formula to look up the cumulative transition probabilities. Cells B12 through E15 holds the Transition Matrix (probabilities of state changes), but these are not cumulative probabilities. Cells U2 through Z5 contains the lookup table of cumulative probabilities (see Figure 6). Cells U2 through V3 are named cells, with the name “Excellent”. We name cells by selecting the cell or range of cells and entering a name in Name Box (see Figure 7). In this range of cells, the cumulative probability is on the left (column U) and the state is on the right (column V).

| T | U         | V       | W    | X       | Y | Z       |
|---|-----------|---------|------|---------|---|---------|
| 1 | Excellent |         | Good |         |   |         |
| 2 | Excellent | 0       | 1    |         |   |         |
| 3 | Good      | =U2+B12 | 2    | 0       | 2 |         |
| 4 | Average   |         |      | =W3+C13 | 3 |         |
| 5 | Bad       |         |      | =W4+C14 | 4 | =Y4+D14 |

**Figure 6.** Cumulative probabilities of the Transition Matrix

The formula =Vlookup(B19,Excellent,2) tells Excel to look in cell B19 for the starting state for that week, the go to the range of cells to find the cumulative probability (in column U) that corresponds to the state in cell B19, which can be found in the 2<sup>nd</sup> column of the range of cells named “Excellent” (column V). A similar formula is entered in cells E19 and F19 for “Good” and “Average” respectively.

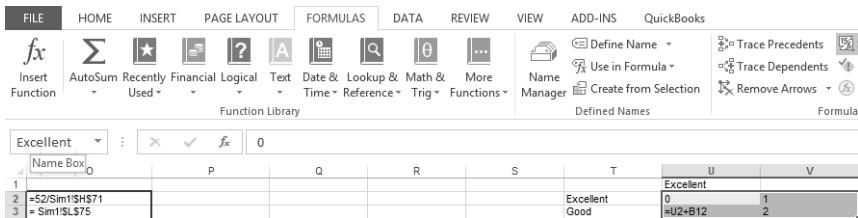


Figure 7. The Name Box is located in the upper left under the File menu tab.

In cell G19, we type the formula for the End State (see Figure 8). =INDEX(D19:F19,C19). To use values returned as an array, we enter the INDEX function as an array formula in a horizontal range of cells for a row, in this case row 19. An array formula is a formula that performs multiple calculations on one or more sets of values, and then returns either a single result or multiple results. Array formulas are enclosed between braces {} and are entered by pressing CTRL+SHIFT+ENTER.) This formula tells Excel to look in cell C19 for a state number, which is used as a column number for the array, and then in cells D19 to F19 for a corresponding column value. For example, if the value in C19 is 2, Excel will look in the 2<sup>nd</sup> column of the array, which is E19, and returns the value in E19.

|    | G                   | H             | I                          | J                       | K                |
|----|---------------------|---------------|----------------------------|-------------------------|------------------|
| 17 |                     |               |                            |                         |                  |
| 18 | End State           | Cycle?        | Cost (Y)                   | Cum Y                   | E(Y)             |
| 19 | =INDEX(D19:F19,C19) | =IF(G19=4,"") | =VLOOKUP(G19,CostLookup,2) | =I19                    | =IF(H19=1,J19,0) |
| 20 | =INDEX(D20:F20,C20) | =IF(G20=4,"") | =VLOOKUP(G20,CostLookup,2) | =IF(H19<>1,I20+J19,I20) | =IF(H20=1,J20,0) |
| 21 | =INDEX(D21:F21,C21) | =IF(G21=4,"") | =VLOOKUP(G21,CostLookup,2) | =IF(H20<>1,I21+J20,I21) | =IF(H21=1,J21,0) |
| 22 | =INDEX(D22:F22,C22) | =IF(G22=4,"") | =VLOOKUP(G22,CostLookup,2) | =IF(H21<>1,I22+J21,I22) | =IF(H22=1,J22,0) |
| 23 | =INDEX(D23:F23,C23) | =IF(G23=4,"") | =VLOOKUP(G23,CostLookup,2) | =IF(H22<>1,I23+J22,I23) | =IF(H23=1,J23,0) |
| 24 | =INDEX(D24:F24,C24) | =IF(G24=4,"") | =VLOOKUP(G24,CostLookup,2) | =IF(H23<>1,I24+J23,I24) | =IF(H24=1,J24,0) |
| 25 | =INDEX(D25:F25,C25) | =IF(G25=4,"") | =VLOOKUP(G25,CostLookup,2) | =IF(H24<>1,I25+J24,I25) | =IF(H25=1,J25,0) |

Figure 8. Formulas for cells G19 to K19.

Cell H19 is an If-Then-Else statement, IF(G19=y,1,"") which says look for the end state in cell G19 and if it is “Bad”=4, replace the machine with a new one (“Excellent”=1), otherwise leave it blank.

In cell I19 we type the formula that looks of the cost for the end state. The formula is another vertical lookup table name CostLookup (see Figure 9). The formula VLOOKUP(G19,CostLookup, 2) tells Excel to look for the value of cell G19, then go to the lookup table and find the correspond value, and then return the value in the second column (the cost or in this case the negative cost, which is the profit).

|    | V           | W   |
|----|-------------|-----|
| 10 | Cost lookup |     |
| 11 | 1           | 90  |
| 12 | 2           | 70  |
| 13 | 3           | 50  |
| 14 | 4           | -60 |

**Figure 9.** CostLookup table.

In cell J19, we enter the formula for the cumulative profit, which for week on is cell I19. For week two we enter the formula  $\text{IF}(H19 <> 1, I20 + J19, I20)$ . This formula goes in cell J20. It tells Excel to look in cell H19, the cell that determines if we replace the machine, and if it is not 1=Excellent, then add the cost in cell I20 to the cost in cell J19, otherwise, do not add the cost, but just enter the previous cost.

Cell K19 holds the expected value of Y,  $\text{IF}(H19 = 1, J19, 0)$ , which tells Excel to look in cell H19 to see if the machine is replaced and if so then return the cumulative cost, otherwise the expected value is 0. For example, if the machine is Bad we replace it and the value in cell H19 will be 1=Excellent, in which case we incur the cost of replacement in J19. If we do incur the cost (-\$60), it is subtracted in the cumulative profit computation.

We'll leave the remaining simulation column for you to figure out. We now turn to the simulation metric calculations on the worksheet name Machine Maint. Figure 10 shows the cell we will be working on.

This worksheet contains the results of all three simulations, which are the same models copied and pasted to different worksheets, and consequently have different results due to the RAND() function. The first metric we will look at is the Total Profit.

| A              | B                     | C                     | D                     | E                 |
|----------------|-----------------------|-----------------------|-----------------------|-------------------|
| 1              | Total Profit          | Total Profit          | Total Profit          | Total Profit      |
| 2              | Sim1                  | Sim2                  | Sim3                  | GRAND             |
| 3 WeeklyProfit | Total Profit          | Total Profit          | Total Profit          |                   |
| 4 90           | =Sim1!\$L\$3          | =Sim2!\$L\$3          | =Sim3!\$L\$3          | =AVERAGE(B4:D4)   |
| 5 70           | =Sim1!\$L\$4          | =Sim2!\$L\$4          | =Sim3!\$L\$4          | =AVERAGE(B5:D5)   |
| 6 50           | =Sim1!\$L\$5          | =Sim2!\$L\$5          | =Sim3!\$L\$5          | =AVERAGE(B6:D6)   |
| 7 -60          | =Sim1!\$L\$6          | =Sim2!\$L\$6          | =Sim3!\$L\$6          | =AVERAGE(B7:D7)   |
| 8              | =SUM(B4:B7)           | =SUM(C4:C7)           | =SUM(D4:D7)           | =AVERAGE(B8:D8)   |
| 9              |                       |                       |                       |                   |
| 10             |                       |                       |                       |                   |
| 11 Frequency   |                       |                       |                       |                   |
| 12             | Sim1                  | Sim2                  | Sim3                  | GRAND             |
| 13 Excellent   | =AVERAGE(Sim1!\$B\$7) | =AVERAGE(Sim2!\$B\$7) | =AVERAGE(Sim3!\$B\$7) | =AVERAGE(B13:D13) |
| 14 Good        | =AVERAGE(Sim1!\$B\$7) | =AVERAGE(Sim2!\$B\$7) | =AVERAGE(Sim3!\$B\$7) | =AVERAGE(B14:D14) |
| 15 Average     | =AVERAGE(Sim1!\$B\$7) | =AVERAGE(Sim2!\$B\$7) | =AVERAGE(Sim3!\$B\$7) | =AVERAGE(B15:D15) |
| 16 Bad         | =AVERAGE(Sim1!\$B\$7) | =AVERAGE(Sim2!\$B\$7) | =AVERAGE(Sim3!\$B\$7) | =AVERAGE(B16:D16) |

**Figure 10.** First set of simulation Metrics

In cell B4, we enter the formula `Sim1!$L$3`, which tell Excel to look on the worksheet called Sim1 and return the value in cell L3, corresponding to the profit for an Excellent machine. The remaining formulas in column B are similar, as are the formulas for the other two simulations.

Cells B13 to B16 are the average values for the frequencies for the states in simulation 1. The formula in cell B13 is `=AVERAGE(Sim1!$B$7)`. Cell B73 in the Sim1 worksheet is where we get the frequency for an Excellent machine. The remaining rows and columns should be self-explanatory.

Turning our attention to the next set of metrics, we refer to Figure 11. These cells hold the Average Cycle Length for a machine before it is replaced, the Average Weekly Profit, the standard deviation of Weekly Profit, and the confidence interval for the Weekly profit. These are located in cells G3 through J7. Cells G12 through J16, hold the same calculation for the Total Profit. The former set of cells only refer to the worksheet containing these metrics for the respective simulation, so we have added cells G19 through J23 to show you what the formulas look like.

The mean value formula for cell B3, refers to cell H71 in the Sim1 worksheet and is 52 weeks divided by the total number of time we replaced a machine. The formula we used in the Sim1 worksheet is `=Average(Sim1!$O$2)`, where cell O2 is the cell in Sim1 that capture the average cycle length. That formula is `=52/H71`.

|                      | F                      | G                      | H                      | I                 | J     |
|----------------------|------------------------|------------------------|------------------------|-------------------|-------|
| 1                    |                        |                        |                        |                   |       |
| 2                    |                        | Sim1                   | Sim2                   | Sim3              | GRAND |
| 3 Avg Cycle Length   | =52/Sim1!\$H\$71       | =52/Sim2!\$H\$71       | =52/Sim3!\$H\$71       | =AVERAGE(G3:I3)   |       |
| 4 Avg Weekly Profit  | = Sim1!\$L\$75         | = Sim2!\$L\$75         | = Sim3!\$L\$75         | =AVERAGE(G4:I4)   |       |
| 5 Stdev              | =Sim1!\$L\$80          | =Sim2!\$L\$80          | =Sim3!\$L\$80          | =AVERAGE(G5:I5)   |       |
| 6 lower 95% CI       | =Sim1!\$L\$81          | =Sim2!\$L\$81          | =Sim3!\$L\$81          | =AVERAGE(G6:I6)   |       |
| 7 upper 95% CI       | =Sim1!\$L\$82          | =Sim2!\$L\$82          | =Sim3!\$L\$82          | =AVERAGE(G7:I7)   |       |
| 8                    |                        |                        |                        |                   |       |
| 9                    |                        |                        |                        |                   |       |
| 10                   |                        | Total Profit           |                        |                   |       |
| 11                   |                        | Sim1                   | Sim2                   | Sim3              | GRAND |
| 12 Mean              | =AVERAGE(Sim1!\$L\$7)  | =AVERAGE(Sim2!\$L\$7)  | =AVERAGE(Sim3!\$L\$7)  | =AVERAGE(G12:I12) |       |
| 13 Stdev             | =STDEVA(Sim1!\$L\$7,1) | =STDEVA(Sim2!\$L\$7,1) | =STDEVA(Sim3!\$L\$7,1) | =AVERAGE(G13:I13) |       |
| 14 95%CI Lower       | =Sim1!\$L\$11-NORMSIN  | =Sim2!\$M\$11-NORMSIN  | =Sim3!\$M\$11-NORMSIN  | =AVERAGE(G14:I14) |       |
| 15 95%CI Upper       | =Sim1!\$L\$11+NORMSIN  | =Sim2!\$M\$11+NORMSIN  | =Sim3!\$M\$11+NORMSIN  | =AVERAGE(G15:I15) |       |
| 16 Avg Cycle Length  | =AVERAGE(Sim1!\$O\$2)  | =AVERAGE(Sim2!\$O\$2)  | =AVERAGE(Sim3!\$O\$2)  | =AVERAGE(G16:I16) |       |
| 17                   |                        |                        |                        |                   |       |
| 18                   |                        | Sim1                   | Sim2                   | Sim3              | GRAND |
| 19 Avg Cycle Length  | =AVERAGE(Sim1!\$O\$2)  | =AVERAGE(Sim2!\$O\$2)  | =AVERAGE(Sim3!\$O\$2)  | =AVERAGE(G19:I19) |       |
| 20 Avg Weekly Profit | =AVERAGE(Sim1!\$O\$3)  | =AVERAGE(Sim2!\$O\$3)  | =AVERAGE(Sim3!\$O\$3)  | =AVERAGE(G20:I20) |       |
| 21 Stdev             | =AVERAGE(Sim1!\$O\$4)  | =AVERAGE(Sim2!\$O\$4)  | =AVERAGE(Sim3!\$O\$4)  | =AVERAGE(G21:I21) |       |
| 22 lower 95% CI      | =Sim1!\$P\$12-NORMSIN  | =Sim2!\$P\$12-NORMSIN  | =Sim3!\$P\$12-NORMSIN  | =AVERAGE(G22:I22) |       |
| 23 upper 95% CI      | =Sim1!\$P\$12+NORMSIN  | =Sim2!\$P\$12+NORMSIN  | =Sim3!\$P\$12+NORMSIN  | =AVERAGE(G23:I23) |       |

**Figure 11.** Simulation metric calculations

We now turn our attention to the calculations of the confidence intervals, which are calculated by

$$CI_{0.975} = \bar{x} \pm z_{0.975} * \frac{\sigma}{\sqrt{n}}$$

The formula in cell G14 is lower CI or

$$=Sim1!$L$7-NORMSINV(0.975)*Sim1!$L$12/SQRT(Sim1!$J$7),$$

that is for Sim1, the

$$\text{AVERAGE}-z_{0.975} * \text{STDEVA}/52 \text{ weeks.}$$

STDEVA is the function for calculating the standard deviation of a sample, as found in cell G12 of Sim1 or =STDEVA(Sim1!\$L\$7,1).

Again, we will leave the rest for you to ponder on your own. However, we leave you with the observation that with Excel, we can do so pretty cool simulations.

## 17.8. Simulation with SCILAB

In SCILIB there are three different approaches for modeling a physical system which are described by Ordinary Differential Equations (ODE): (1) Standard SCILAB programming; (2) Xcos Programming; and (3) Xcos plus

Modelica. In this section, we will demonstrate Xcos Programming using a common physical system, the LHY model for drug abuse. We will use this model to show some of the main features of Xcos.

The considered model is the LHY model used in the study of drug abuse (see Figure 1). This model is a continuous-time dynamical system of drug demand for two different classes of users: light users (denoted by  $L(t)$ ) and heavy users (denoted by  $H(t)$ ) which are functions of time . There is another state in the model that represents the decaying memory of heavy users in the years (denoted by  $Y(t)$ ) that acts as a deterrent for new light users. In other words the increase of the deterrent power of memory of drug abuse reduces the contagious aspect of initiation. This approach presents a positive feedback which corresponds to the fact that light users promote initiation of new users and, moreover, it presents a negative feedback which corresponds to the fact that heavy users have a negative impact on initiation. Light users become heavy users at the rate of escalation  $b$  and leave this state at the rate of desistance  $a$ . The heavy users leave this state at the rate of desistance  $g$ .

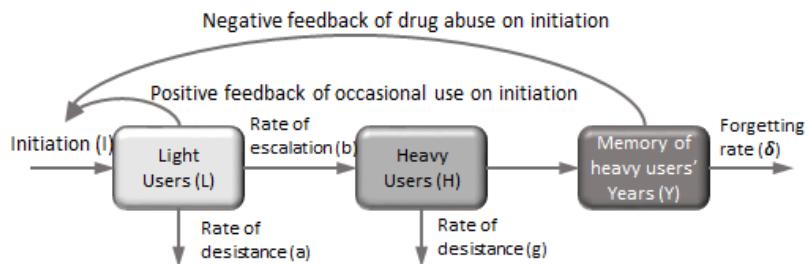


Figure 1.

### Step 1: Mathematical model

The mathematical model is a system of ODE (Ordinary Differential Equation) in the unknowns:

- $L(t)$ , number of light users;
- $H(t)$ , number of heavy users;
- $Y(t)$ , decaying of heavy user years.

The initiation function contains a “spontaneous” initiation  $\tau$  and a memory effect modeled with a negative exponential as a function of the memory of year of drug abuse relative to the number of current light users.

The LHY equations system (omitting time variable for sake of simplicity) is

$$\begin{cases} \dot{L} &= I(L, Y) - (a + b)L \\ \dot{H} &= bL - gH \\ \dot{Y} &= H - \delta Y \end{cases}$$

where the initiation function is

$$I(L, Y) = \tau + L \cdot \max\left\{s_{max}, x^{-q_L^Y}\right\}$$

The problem is completed with the specification of the initial conditions at the time  $t_0$ .

The LHY initial conditions are

$$\begin{cases} \dot{L} &= L_0 \\ \dot{H} &= H_0 \\ \dot{Y} &= Y_0 \end{cases}$$

## **Step 2: Problem data**

### **(Model data)**

a: the annual rate at which light users quit

b: the annual rate at which light users escalate to heavy use

g: the annual rate at which heavy users quit

$\delta$ : the forgetting rate

### **(Model data)**

$$a = 0.163$$

$$b = 0.024$$

$$g = 0.062$$

$$\delta = 0.291$$

### (Initiation function)

$\tau$ : the number of innovators per year

$s$ : the annual rate at which light users attract non-users

$q$ : the constant which measures the deterrent effect of heavy use

$s_{max}$ : the maximum rate of generation for initiation

### (Initiation function)

$$\tau=50000$$

$$s = 0.610$$

$$q = 3.443$$

$$s_{max} = 0.1$$

### (Initial conditions)

$t_0$ : the initial simulation time;

$L_0$ : Light users at the initial time;

$H_0$ : Heavy users at the initial time;

$Y_0$ : Decaying heavy users at the initial time.

### (Initial conditions)

$$t_0 = 1970;$$

$$L_0 = 1.4 \times 10^6;$$

$$H_0 = 0.13 \times 10^2;$$

$$Y_0 = 0.11 \times 10^6.$$

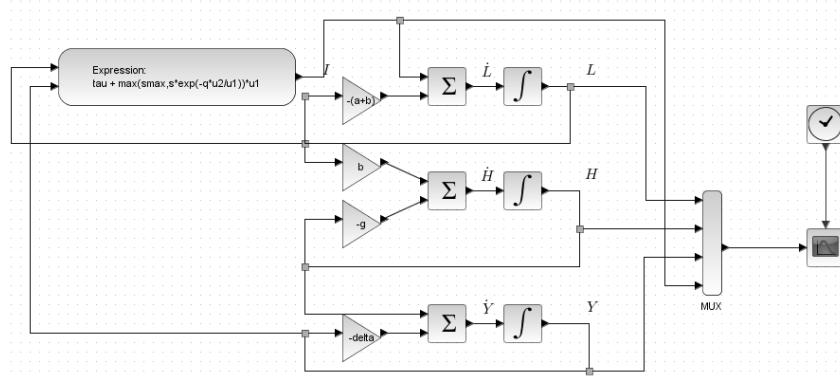
## Step 3: Xcos programming – introduction

Xcos is a free graphical editor and simulator based on Scilab that helps people to model physical systems (electrical, mechanical, automotive, hydraulics, ...) using a graphical user interface based on a block diagram

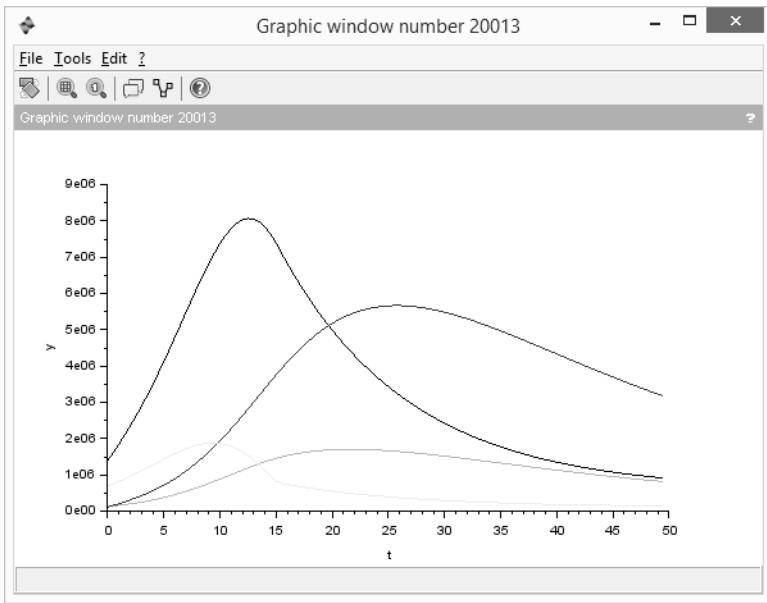
approach. It includes explicit and implicit dynamical systems and both continuous and discrete sub-systems.

This toolbox is particularly useful in control theory, digital and signal processing and model-based design for multidomain simulation, especially when continuous time and discrete time components are interconnected.

As an example of a Xcos diagram, we show in Figure 2 a Xcos model of a LHY model with its graphical output (Figure 3). The first output is relative to the voltage across the capacitor element, while the second one is relative to the current through the voltage generator.



**Figure 2.** LHY Model



**Figure 3.** LHY Model graphical output

#### Step 4: Xcos programming – getting started

Xcos environment can be started from Scilab Console typing

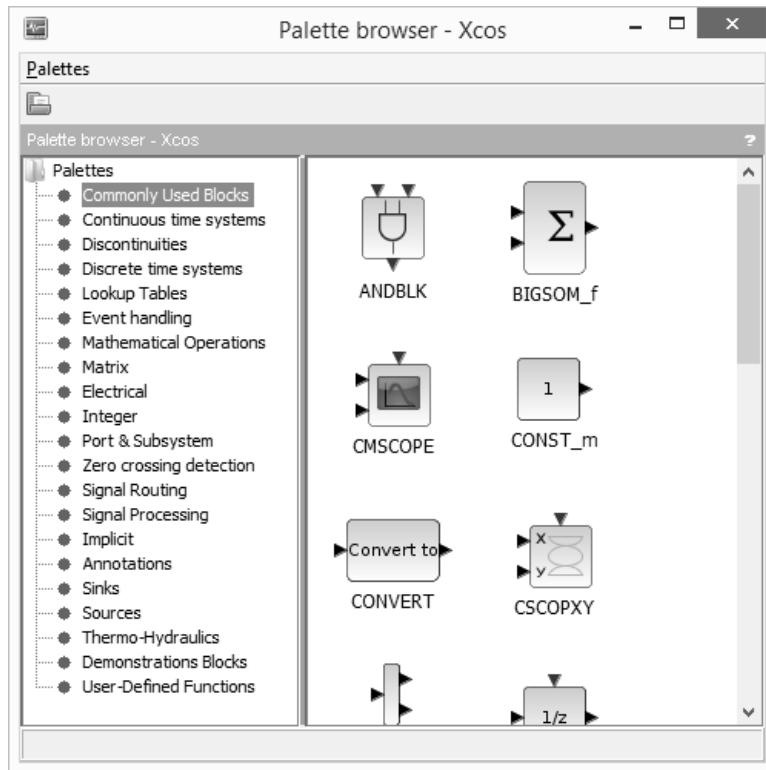
```
--> xcos
```

or clicking on the button  in the Scilab menu bar.

The command starts two windows:

- the palette browser that contains all Xcos available blocks grouped by categories (Figure 4);
- an editor window where the user can drag blocks from the palette browser for composing new schemes (Figure 5).

All Xcos files end with extension “.zcos”. In previous versions of Scilab all Xcos files end with extension “.xcos”.

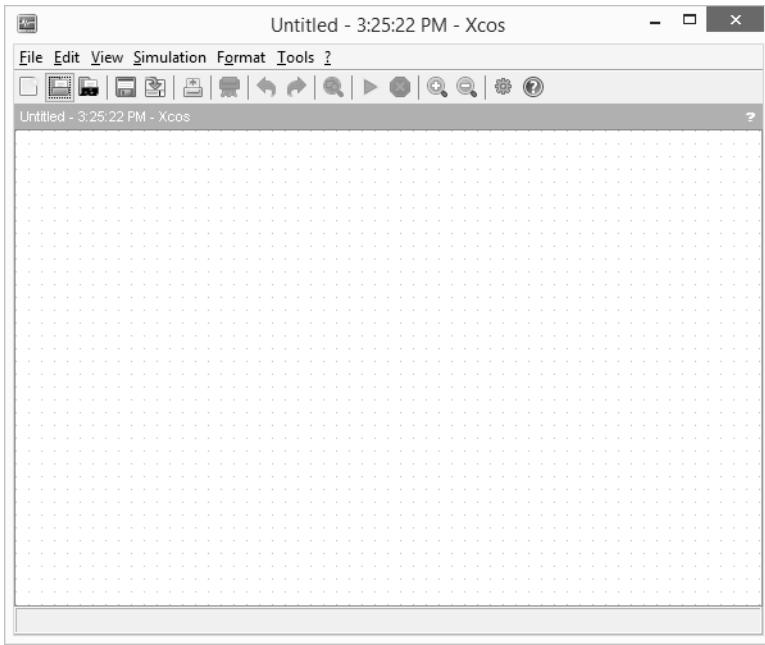


**Figure 4.** Xcos Block Palette

### Step 5: Xcos programming – block types

In Xcos the main object is a block that can be used in different models and projects.

A Xcos block is an element characterized by the following features: input/output ports; input/output activations ports; continuous/discrete time states.



**Figure 5.** Xcos Editor Window

Xcos blocks contain several type of links (see Figure 6):

- Regular links that transmit signals through the blocks ports (black triangle);
- Activation links that transmit activation timing information through the block ports (red triangle);
- Implicit links, see tutorial Xcos + Modelica (black square).

The user should connect only ports of the same type.

Block configuration can be specified from the input mask by double-clicking on the block.



**Figure 6.** Three types of blocks based on connectors

## Step 6: Roadmap

In this section we describe how to construct the LHY model and simulate it in Xcos. We:

- provide a description of all the basic blocks used for the LHY model;
- provide a description of the simulation menu;
- provide a description of how to edit a model;
- construct the LHY scheme;
- test the program and visualize the results.

**Table 1.** LHY Model building steps

| Descriptions        | Steps |
|---------------------|-------|
| Basic blocks        | 9-16  |
| Simulation menu     | 17-19 |
| Editing models      | 20-22 |
| Scheme construction | 23-27 |
| Test and visualize  | 28    |

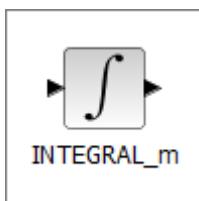
### Step 7: The integral block

Palette: Continuous time systems / INTEGRAL\_m

Purpose: The output of the block  $y(t)$  is the integral of the input  $u(t)$  at the current time step  $t$ .

In our simulation we use this block to recover the variable  $L(t)$  starting from its derivative  $L'(t)$ . The initial condition can be specified in the input mask. Hint: Numerically it is always more robust using the integral block instead of the derivative block.

The integral block



## Step 6: The sum block

Palette: Math operations / BIGSOM\_f

Purpose: The output of the block  $y(t)$  is the sum with sign of the input signals. The sign of the sum can be specified from the input mask with "+1" for "+" and "-1" for "-".

The sum block

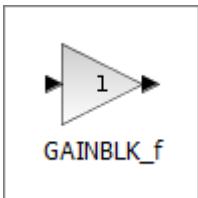


## Step 9: The gain block

Palette: Math operations / GAINBLK\_f

Purpose: The output of the block  $y(t)$  is the input signal  $u(t)$  multiplied by the gain factor. The value of the gain constant can be specified from the input mask.

The gain block



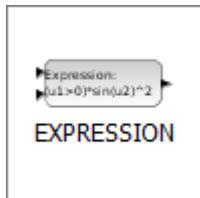
## Step 10: The expression block

Palette: User defined functions / EXPRESSION

Purpose: The output of the block  $y(t)$  is a mathematical combination of the input signals  $u_1, u_2, \dots, u_N$  (max 8). The name,  $u$ , followed by a number, is mandatory. More precisely,  $u_1$  represents the first input port signal,  $u_2$  represents the second input port signal, and so on.

Note that constants that appear in the expression must first be defined in the “context menu” before their use.

The expression block



### Step 11: The clock block

Palette: Sources / CLOCK\_c

Purpose: This block generates a regular sequence of time events with a specified period and starting at a given initialization time. We use this block to activate the scope block (see next step) with the desired frequency.

The clock block



### Step 12: The scope block

Palette: Sinks / CSCOPe

Purpose: This block is used to display the input signal (also vector of signals) with respect to the simulation time. For a better visualization it may be necessary to specify the scope parameters like  $ymin$ ,  $ymax$  values.

The scope block

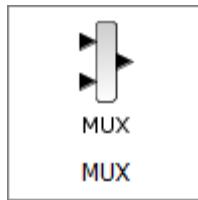


### Step 13: The multiplexer block

Palette: Signal routing / MUX

Purpose: This block merges the input signals (maximum 8) into a unique vector output signal. We use this block for plotting more signals in the same windows. The number of input ports can be specified from the input mask.

The multiplexer block

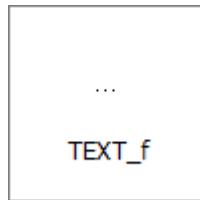


### Step 14: The annotation block

Palette: Annotations palette / TEXT\_f

Purpose: This block permits to add comments to the scheme. Comments can also be in LaTex coding. This block can be also called by a double click of the mouse on the scheme. Examples of LaTex code are  $\$L\$$  for generating  $L$  and  $\$\dot{L}\$$  for generating  $L'(t)$ .

The annotation block



## Step 15: The simulation starting time

Each Xcos simulation starts from the initial time 0 and ends at a specified final time.

The ending simulation time should be specified in the "Simulation/Setup" menu in the "Final integration time" field.

Simulation starting time is 0.

## Step 16: Set simulation parameters

The simulation parameters such as the "final integration time" and solver tolerances can be specified from the "set parameters" (Figure 8) dialog in the "Simulation/Setup" menu (Figure 7).

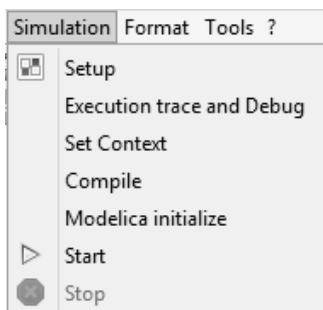


Figure 7. Simulation Setup Menu

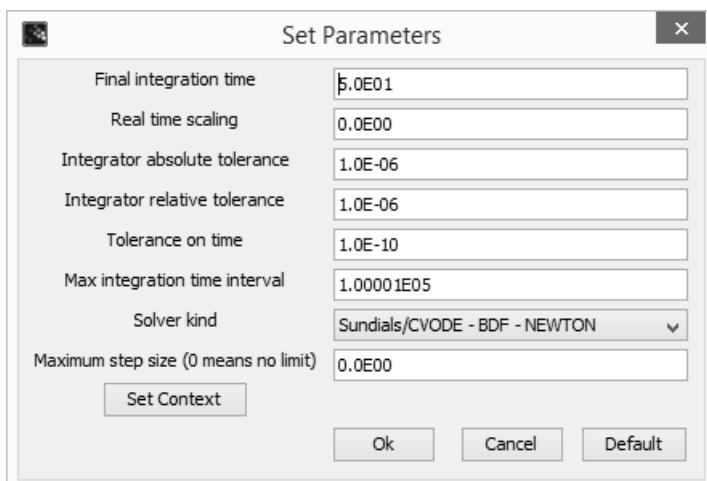


Figure 8. Set Parameters

### Step 17: Set Context

The model constants used in the block definitions can be entered in the "simulation set context". This menu is available from the "Simulation/Set Context" menu (Figure 9).

In our simulation we set here all the model constants and the initial conditions, such that it is easier to change the model value for a new simulation since all the constants are available in a unique place.

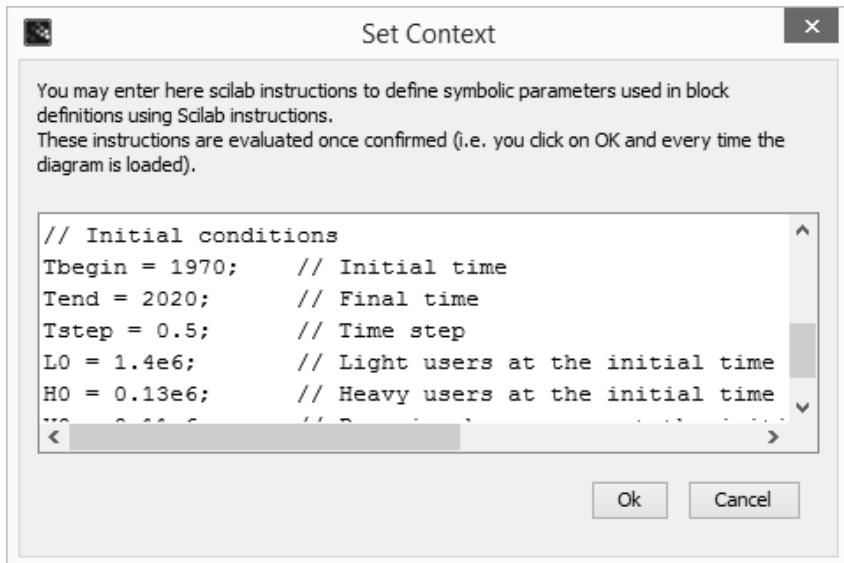


Figure 9. Simulation “Set Context” window

### Step 18: Editing the model: Align blocks

Here we report some hints to improve the visual quality of the connections between blocks. The first step is to drag some elements in the diagram and align them (see Figures 10 and 11).

To make a selection:

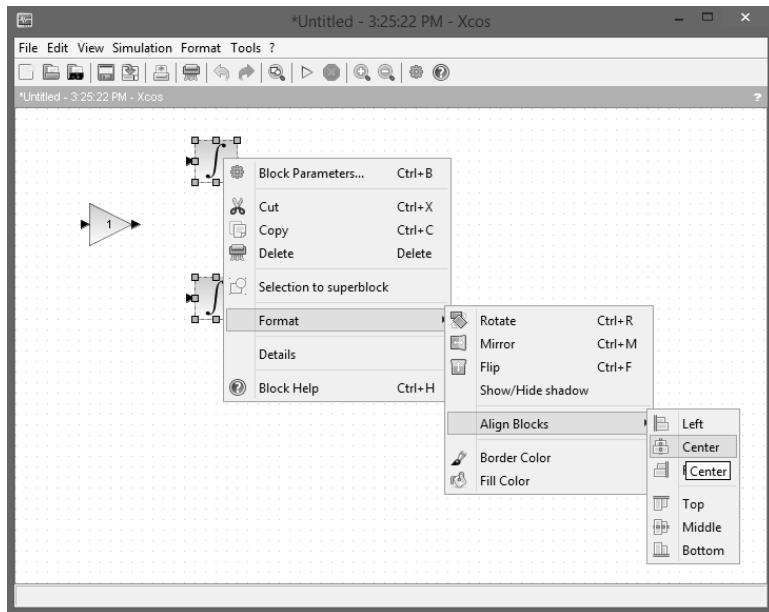
- Left-click where you want to start your selection;
- Hold down your left mouse button and drag the mouse until you have highlighted the area you want;

or

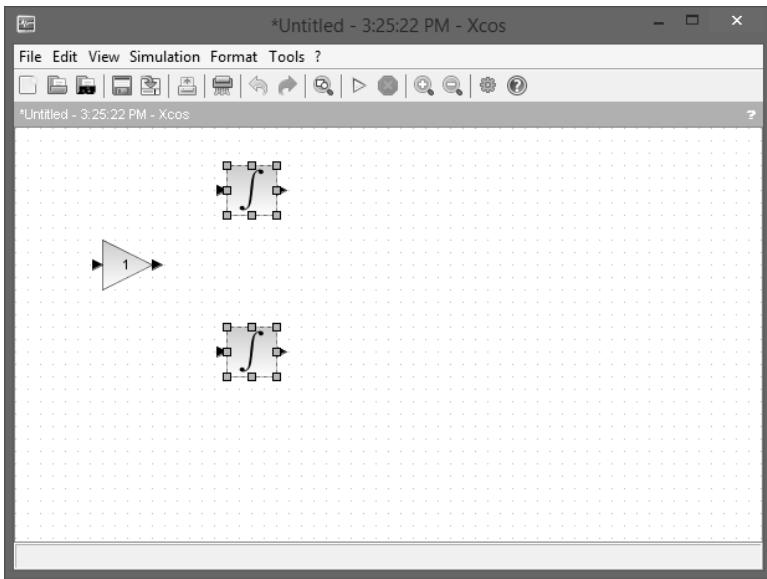
- Left-click of the mouse over the element you want;
- Ctrl + Left-click on elements that you want to select;

To align elements:

- Right-click of the mouse and select: Format -> Align Blocks -> Center.



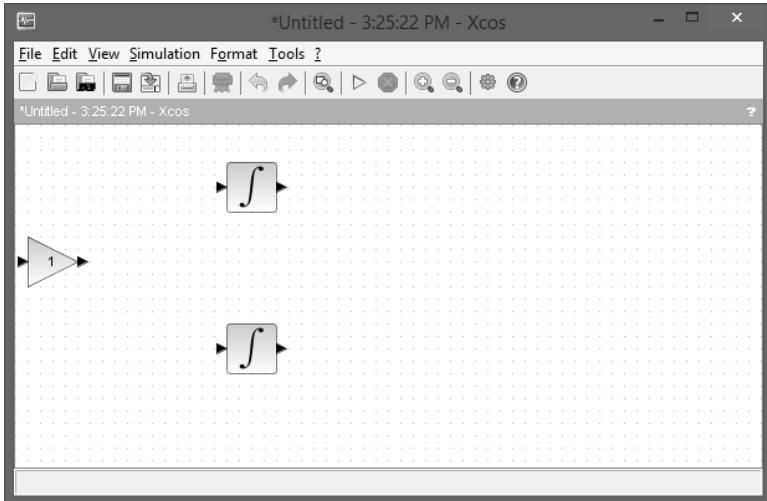
**Figure 10.** Before



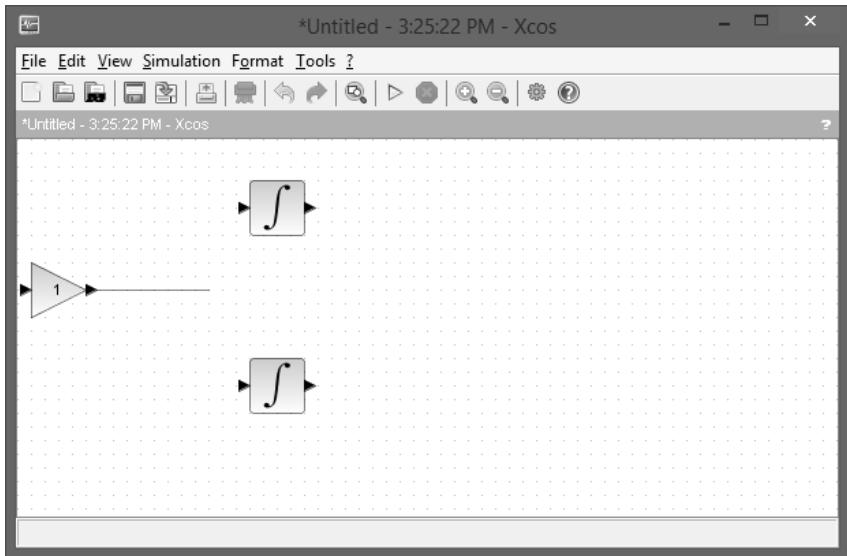
**Figure 11.** After

### Step 19: Editing the model: intermediate points

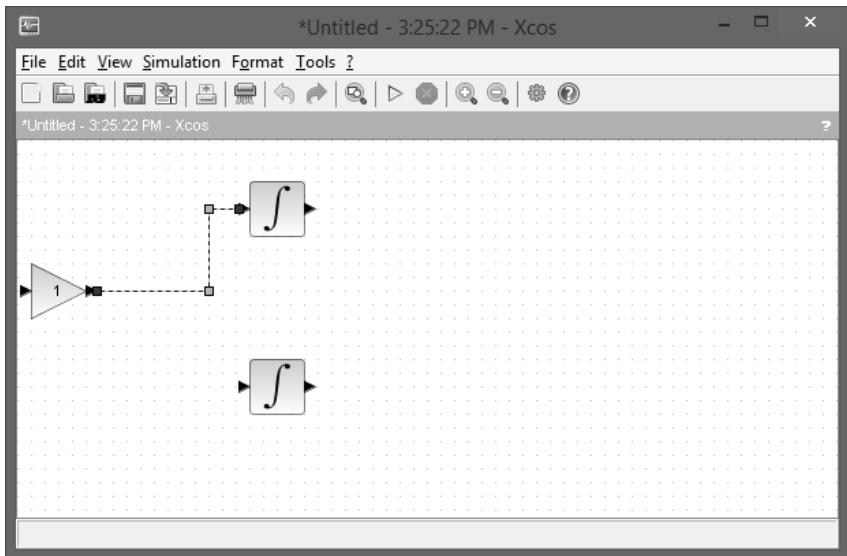
You can link elements specifying the path using a mouse click on intermediate points (See figures 12, 13 and 14).



**Figure 12.** Stating view



**Figure 13.** Intermediate path

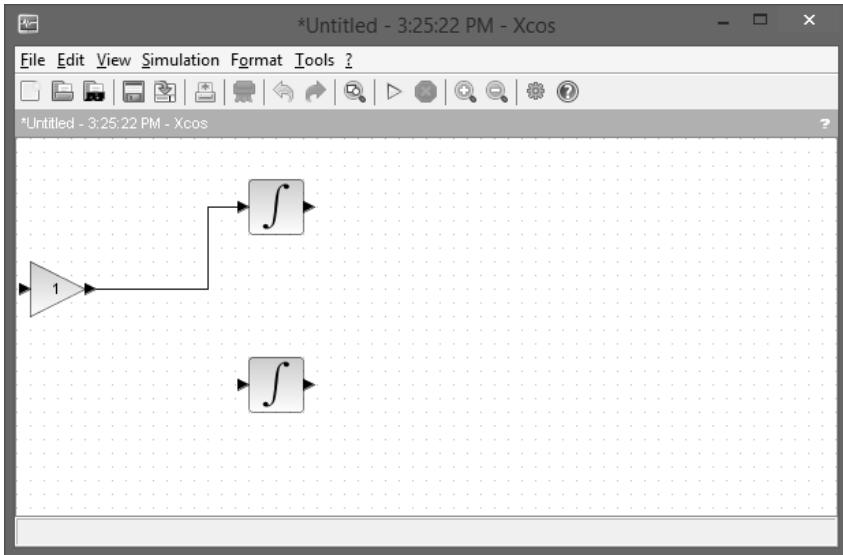


**Figure 14.** Final path

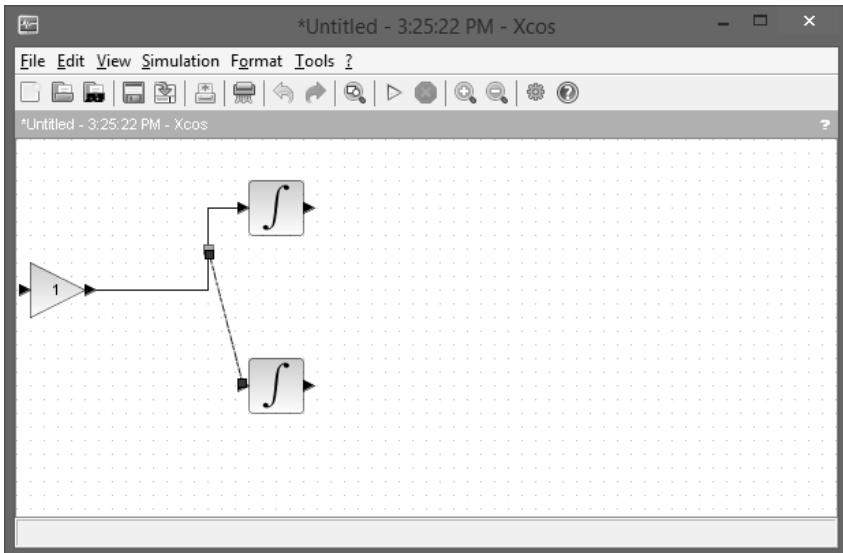
#### Step 20: Editing the model

The same if you want to link elements to an already existing path (start from the element port and draw the path you want). Click with the

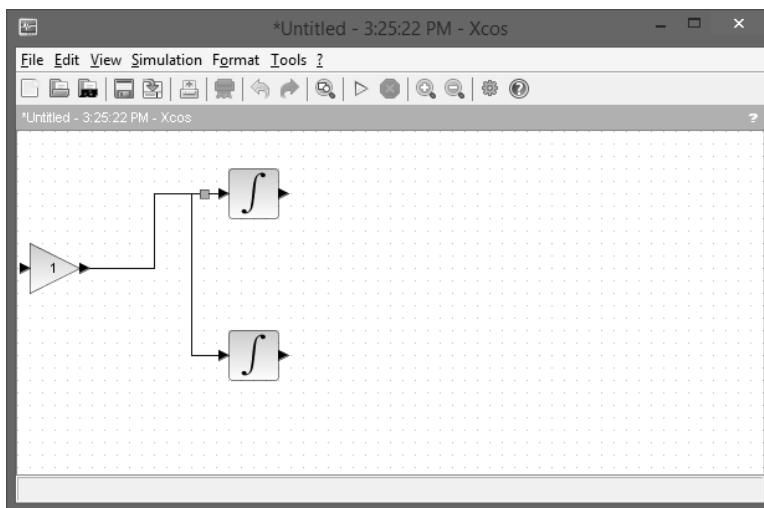
mouse at the right position although the visualization is not correct (see Figures 15, 16 and 17).



**Figure 15.** Starting configuration



**Figure 16.** Intermediate path

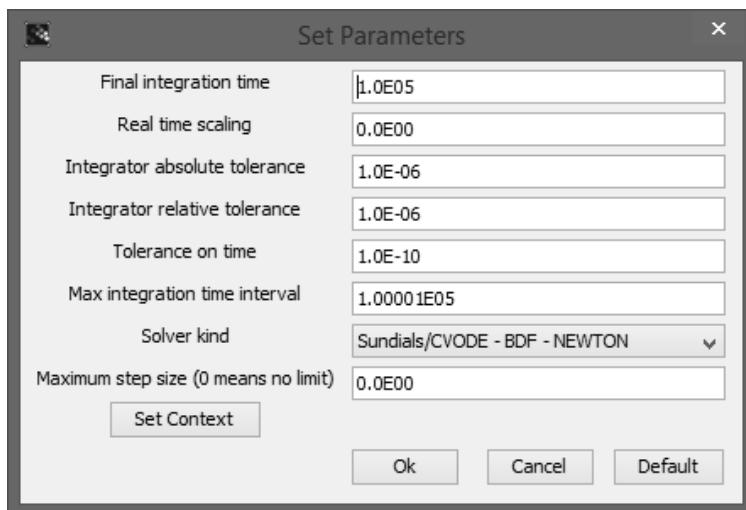


**Figure 17.** Final path

### Step 21: Make the LHY scheme

Set the “Final integration time” to 50 in the "Simulation/Setup" menu;

In our simulation we modify the value of the “Final integration time” to 50 because the initial time in Xcos is 0, which corresponds to the year 1970 of our model. This means that 50 corresponds to the year 2020 of our model (see Figure 18).



**Figure 18.** Set Parameters

## Step 22: Make the LHY scheme

```
Set the model constants in the "Simulation/Set Context"
menu.;

tau = 5e4; // Number of innovators per year (initiation)
s = 0.61; // Annual rate at which light users attract non-
users (initiation)
q = 3.443; // Constant which measures the deterrent effect
of heavy users (initiation)
smax = 0.1; // Upper bound for s effective (initiation)
a = 0.163; // Annual rate at which light users quit
b = 0.024; // Annual rate at which light users escalate to
heavy use
g = 0.062; // Annual rate at which heavy users quit
delta = 0.291; // Forgetting rate
// Initial conditions
Tbegin = 1970; // Initial time
Tend = 2020; // Final time
Tstep = 0.5; // Time step
L0 = 1.4e6; // Light users at the initial time
H0 = 0.13e6; // Heavy users at the initial time
Y0 = 0.11e6; // Decaying heavy user at the initial time
```

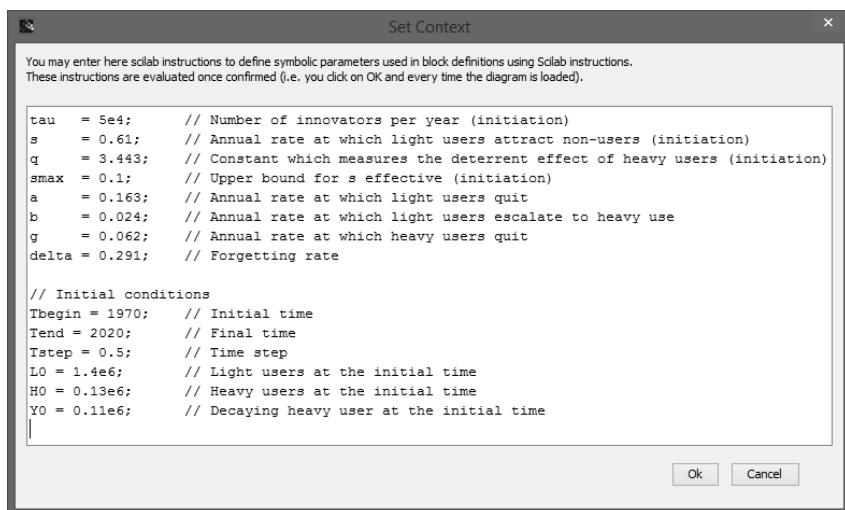
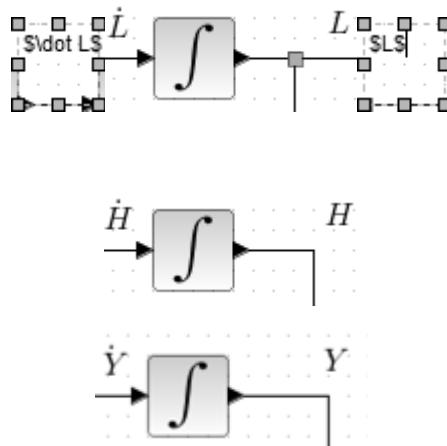


Figure 19. Set Context

## Step 23: Make the LHY scheme

Drag integration blocks and specify for each blocks the appropriate initial conditions (see Figure 20). Add also some annotations for  $\dot{L}$ ,  $L$ ,  $\dot{H}$ ,  $H$ ,  $\dot{Y}$  and  $Y$  (see Step 16).



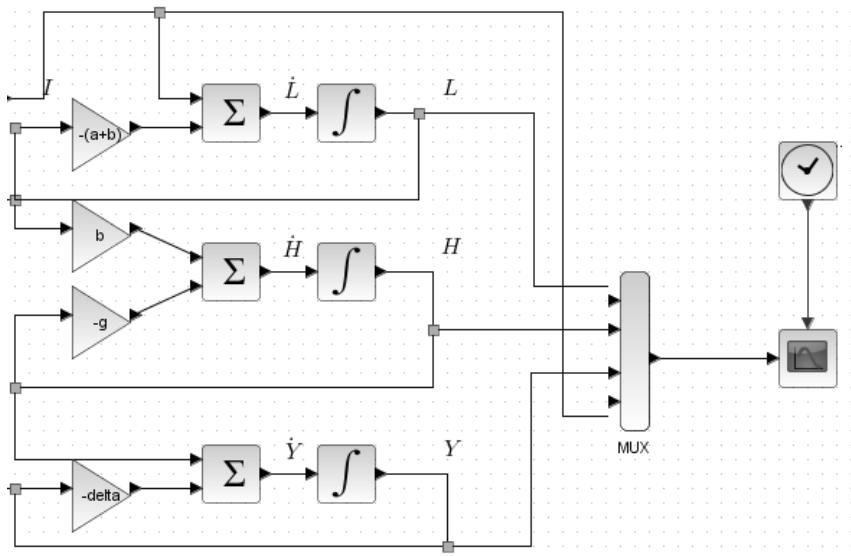
**Figure 20.** The LHY Scheme

#### Step 24: Complete the LHY scheme

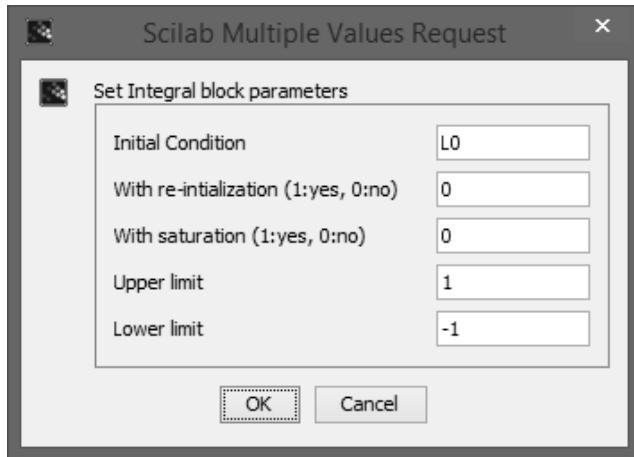
Add:

- Three integral blocks (Figure 22);
- Three sum blocks (Figure 23);
- Four gain blocks with the appropriate constants (Figure 24);
- A multiplexer block (4 ports: L,H,Y,I) (Figure 25);
- A clock block (Period=0.5, Initiation = 0) (Figure 26).
- A scope block (ymin = 0, ymax = 9e+6, Refresh period = 50) (Figure 27);

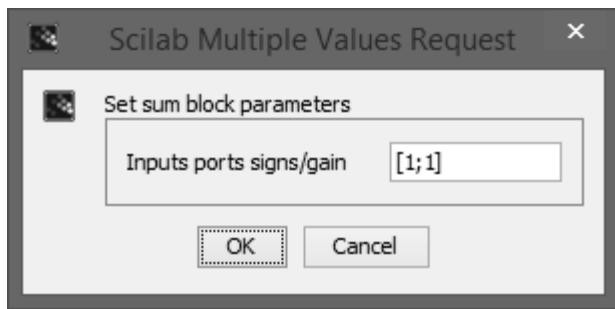
Connect the blocks as shown in the figure. For a more readable diagram it is better to comment blocks and connections using annotation blocks as reported in Figure 21.



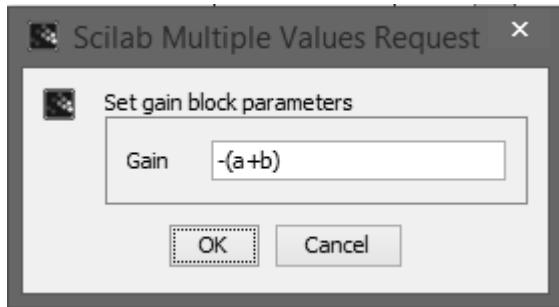
**Figure 21.** Competing the LHY Scheme



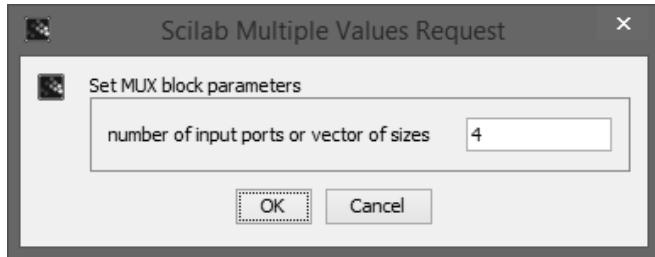
**Figure 22.** Integral block settings for L0, H0 and Y0 (top to bottom)



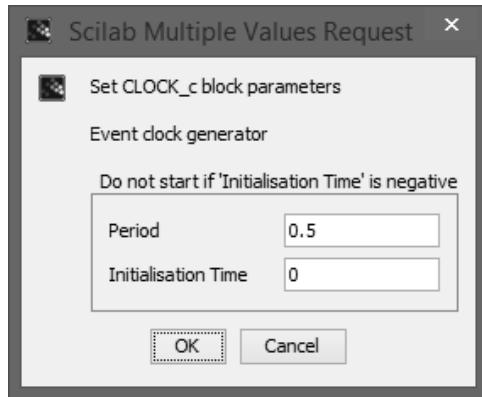
**Figure 23.** Sum block settings (all)



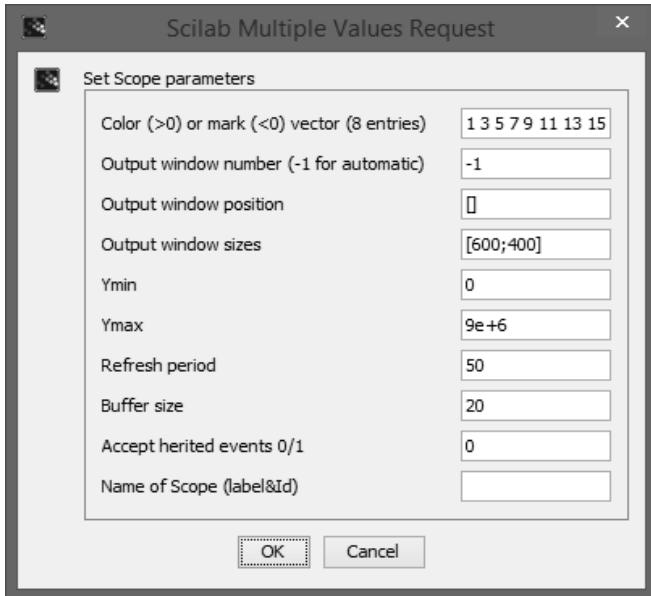
**Figure 24.** Gains block setting  $-(a+b)$ , b, -g and  $-\delta$  (top to bottom)



**Figure 25.** Multiplexor block settings



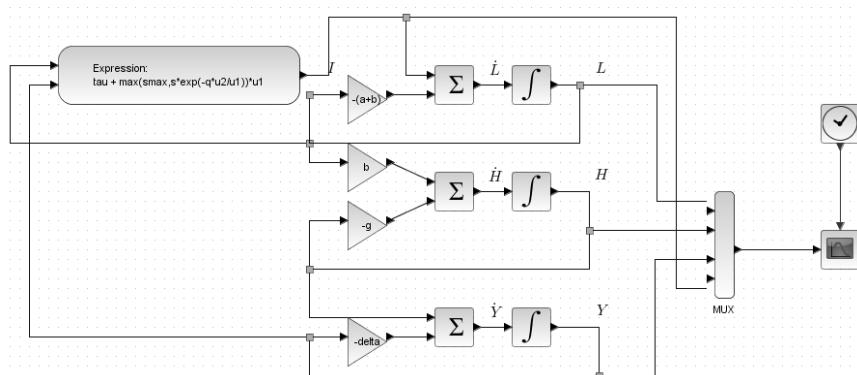
**Figure 26.** Clock block settings



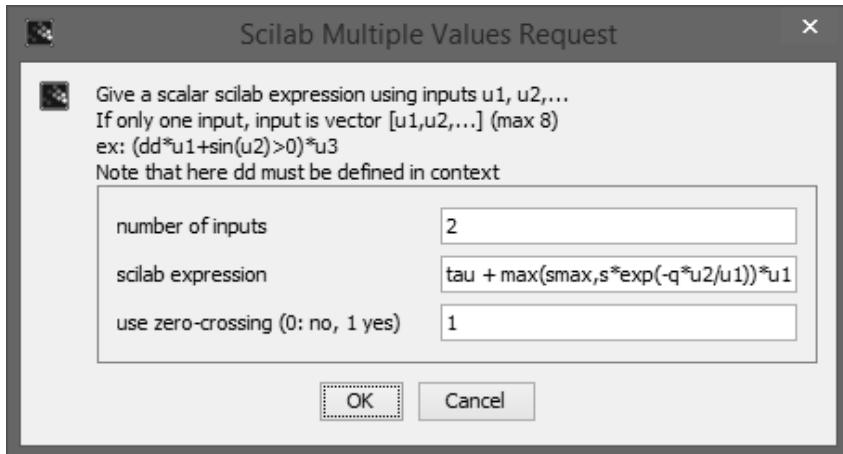
**Figure 27.** Scope block setting

### Step 25: Make the LHY scheme

Add: One expression block (with two inputs and the Scilab expression  $\tau + \max(s_{\max}, s * \exp(-q * u_2 / u_1)) * u_1$ ) (Figures 28 and 29);



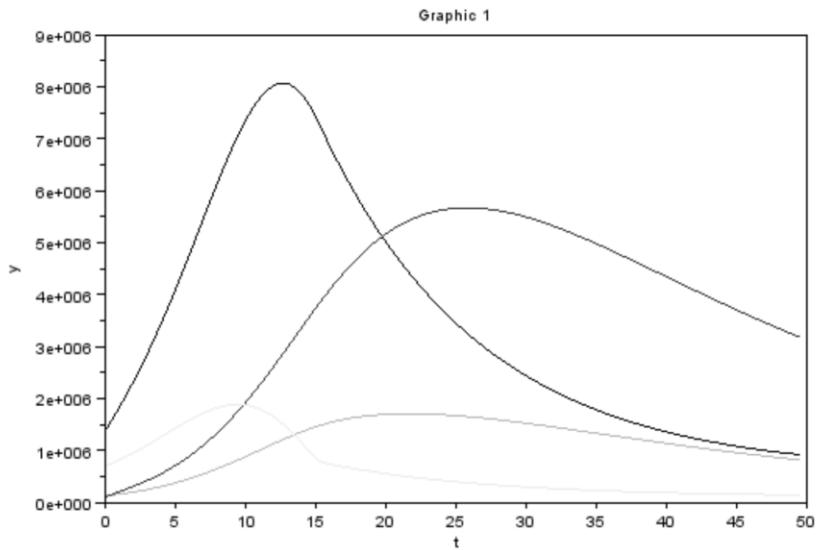
**Figure 28.** Completed LHY Scheme with Expression Block



**Figure 29.** The Expression Block

### Step 26: Running and testing

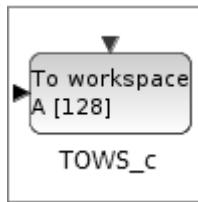
Click on the menu button



### **Step 27: Exercise #1**

Modify the main program file such that it is possible to write data to Scilab environment and plot data from Scilab.

Hint: Use the block in “Sinks” “To workspace” as reported on the right.



### **Step 30: Concluding remarks and References**

In this tutorial we have shown how the LHY model can be implemented in Scilab/Xcos

References for further studies.

1. Scilab Web Page: Available: [www.scilab.org](http://www.scilab.org).
2. D. Winkler, J. P. Caulkins, D. A. Behrens and G. Tragler, "Estimating the relative efficiency of various forms of prevention at different stages of a

drug epidemic," Heinz Research, 2002.  
<http://repository.cmu.edu/heinzworks/211/>.

## 17.9. Review Problems

9. Provided the partially completed Excel spreadsheet, Constructive Constructors, which represents a construction project schedule, complete the simulation inputs using the following information:  
The Activity times are distributed according to a truncated Normal Distribution with mean  $\mu$  and standard deviation  $\sigma$ .
  - a. Set the simulation up to run three simultaneous simulations.  
What is the average project completion time?
  - b. How could you improve your simulation results?
10. Given the Excel spreadsheet Queuing Simulator, and the mean interarrival time of 30 with mean service time 20, run it several times and develop a grand mean for each metric. Then interpret the results.
11. Build an M/M/1 queue model in Excel and run 5 simulations to measure the grand mean.
12. Generate an exponential random variate with interarrival time mean equal to 3 and service time mean equal to 2. Simulate twenty interactions and plot the results.
13. "Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the other doors, opens another door, say No. 3, which has a goat. He then says to you, 'Do you want to pick door No. 2?' Is it to your advantage to take the switch?"
  - a. Suppose the player plays this game  $n = 10$  times. Which door has the car? Hint: Try

```
car(num of choices, num of repetitions, replace=T)  
car
```

- b. Which door is chosen?
- c. Switch and win: the car is not behind the chosen door. If you switch 10 times, what is the probability of winning? Hint: Try

```
switchwin=(door!=car) # what does door!=car do?  
switchwin  
sum(switchwin)/10
```

- d. Not switch and win: the car is behind the chosen door. What is the probability of losing?

14. Try this code for the Monty Hall Problem

```
for(i in 1:1000)  
{  
prize<-sample(doors)[1]  
pick<-sample(doors)[1]  
open<-sample(doors[which(doors!=pick & doors != prize)])[1]  
switchyes<-doors[which(doors != pick & doors != open)]  
if (pick==prize)(xdata=c(xdata,"noswitchwin"))  
if(switchyes==prize)(xdata=c(xdata,"switchwin"))  
}  
length(which(xdata=="switchwin"))  
length(which(xdata=="noswitchwin"))
```

- a. What does the code do?  
b. What is the result?



# 18.What is Discrete Event Simulation?

We cover this topic for information only. The kind of software or the languages used for implementing this kind of simulation is beyond the scope of this text. However, we do have a book dedicated to the subject entitled, *Discrete Event Simulation using ExtendSim*. The following is an excerpt from that book.

## 18.1. Discrete Event Simulation Model Development

In this chapter we will develop a simulation model of a single queue (waiting line) with a single server. Later, we will see that we do not have to simulate this system in order to perform analysis on it—it has a known analytic solution. However, it is instructional to simulate a system that we know something about so that we have confidence in our understanding of discrete event simulation. We will use the model we build in this chapter, and its known solution in performing steady state simulation in Chapter 4 (of *Discrete Event Simulation Using ExtendSim*).

Before we can simulate the single-server queue, we must develop a model of this queuing process. The essence of modeling discrete-event simulation is to determine:

- what variables or components are needed to adequately represent the systems state,
- what events are needed to represent the system changes, and
- what state changes occur within each event, including the details of which events are scheduled and when are they scheduled.

This is usually an iterative process, and there is no one unique model that represents a system. The following sections describe the “what’s” that we need to determine.

### 18.1.1. Event and Event Sequencing

The **state** of a system model is defined to be the collection of variables containing all information necessary to operate the model and record relevant changes in it over time. Any change in any **state variable**

represents a change in the system, and any change in the system is reflected by a change in some state variable ( $s$ ). In a simple queuing system with a single server and a single queue, with identical customers, we could represent the state of the system using two variables,  $(Q,S)$ , where

$Q$  is the number of customers in the queue, and

$S$  represents the status of the server.

### 18.1.2. Events and State Changes

In a **discrete-event dynamic system**, the state changes only at discrete points in time, which mark the occurrence of an **event**. An **event** is any occurrence that causes an instantaneous change in the state of the system. In any system there are only a few distinct type of events. In the simple queuing system there are only three types of events:

- Arrival of a customer
- Beginning of service
- Completion of service

**Event routines** (*transition functions*) specify exactly how a system changes when an event occurs. They update state variables, increment counters, and generate future events.

### 18.1.3. Event Scheduling

In a discrete event simulation, events are generated and scheduled for the future. To implement the dynamics of a discrete-event simulation, a mechanism must be provided to increment time by a **variable** amount between events. A simulation creates an **event notice** to sequence events according to the time of occurrence and apply the associated **event routine** to make the changes to the **system state**. At least two pieces of information must be provided:

- The actual time the event will occur
- The type of event that is schedules to occur

These event notices are stored in a list, called the (*future*) **events lists**, which is ordered by time of occurrence. The process of creating an event

notice, recording information about the event, and placing it in the event list is called **scheduling** the event.

#### 18.1.4. The System Clock

The **system clock** is a special system variable that holds the current system time. The simulation progresses through time in the following way:

- The first event notice on the event list is selected.
- The system clock is set to the event time on this event notice.
- Then the event routine for this type of event is executed to produce the appropriate changes in the system state for this type of event.
- Finally, the event notice is discarded.

This process repeats until the event list is empty or some other signal is given to stop the simulation. The routine that implements this process is called the **timing routine**—the heart of discrete-event simulation.

#### 18.1.5. Queue Taxonomy

We will look at this in more detail in Chapter 3, but before we move on it would be helpful to understand our short-hand for the single server queue. If the distribution of inter-arrival times and service times are exponential (with different means), then the queuing system is known as the “M/M/1” Queue.

#### 18.1.6. Random Number Generation

Before we continue with simulating the M/M/1 queue, there are two other important concepts we need to be familiar with: the generation of random numbers and random variates. Random numbers are used to drive statistical models of processes and to make decisions. The core of these algorithms is the uniformly distributed number between zero and one. That is, a series of independent and identically distributed numbers in the range from 0 to 1, written as IID  $U(0,1)$ . We use deterministic mathematical algorithms to generate a series of numbers that appear random. Since they are not “truly” random, we refer to them as pseudo random numbers. An example of a random number generator is:

$$X(n) = (ax(n - 1) + c)\bmod m$$

$$U(0,1) = x(n)/\max(m) \quad (18.1)$$

Note that random methods tend to generate very skewed sequences of numbers. See Chapter 3, *Probability and Statistics for Simulation*, for more about Random Number Generators, including *ExtendSim 8's Random Number* block.

### 18.1.7. Random Variates

Uniform random numbers drive the selection values from other distributions (random variates). There are three primary methods for generating random variates:

- Inverse Transform Methods assign the U(0,1) number as the Y-axis and traces it back to the X value through the desired cumulative distribution function.
- Acceptance-Rejection Methods use an intermediate function to simplify the selection of numbers and maintain random selection.
- Composition Methods decompose the distribution into a summed series of simpler distributions.

An example of a random variate using the inverse transform method is the exponential distribution:

$$X = \frac{1}{\lambda} \ln(1 - R), R \sim U(0,1) \quad (18.2)$$

The exponential distribution is generally used to represent Interarrival times of customers at a constant rate. See Appendix A for more about Random Variate Generation. *ExtendSim 8* has 35 built-in random variate generators for discrete and continuous probability density functions, and Empirical Table for defining discrete probability distributions.

### 18.1.8. A Generic Simulation Language

In the 1950's, we began seeking languages specifically designed for simulation problems. The General Simulation Program (GSP) was developed in 1960 and was the first Simulation-specific programming language. GSP was developed by K.D. Tocher and D.G. Owen, of General

Electric, and presented at the proceedings of the Second International Conference on Operations Research. The evolved definition of a simulation language defined six key characteristics that the language must possess:

- Be capable of generating random numbers
- Provide for the transformation for statistical distributions
- Be capable of providing list processing
- Provide for statistical analysis
- Be capable of report generation
- Be capable of implementing timing execution

For our step-by-step simulation of the M/M/1 queue, we will use a generic simulation language, as described in **Figure 1: Characteristics of a generic simulation language.**

| Features:  | Event Definition and Scheduling  |
|--|--|
| Constants  | Facilities   |
| Variables  | Event list   |
| Arrays of variables  | System clock: <i>NOW</i>   |
| Arithmetic and logical operators   | Timing routine: <i>simulate</i>  |
| Assignment statements: $A := B + C/D$  | Event routines   |
| Commands or procedures to read in and write out constant and variable values | Commands to:<br>Schedule an event at a given time:<br>Schedule event-type at time $T$<br>Find an event notice in the events list: find event |
| Looping statements: <i>While...Do</i>  | Cancel a scheduled event: cancel event.  |
| Conditional statements: <i>If...Then...Else</i>                              |  |
| Procedures with parameters   |  |
| Functions with parameters  |  |

**Figure 1:** Characteristics of a generic simulation language

A typical simulation program involves both the functions that the language provide and functions that the programmer has to generate. An example of a typical simulation program is shown in **Figure 1.**

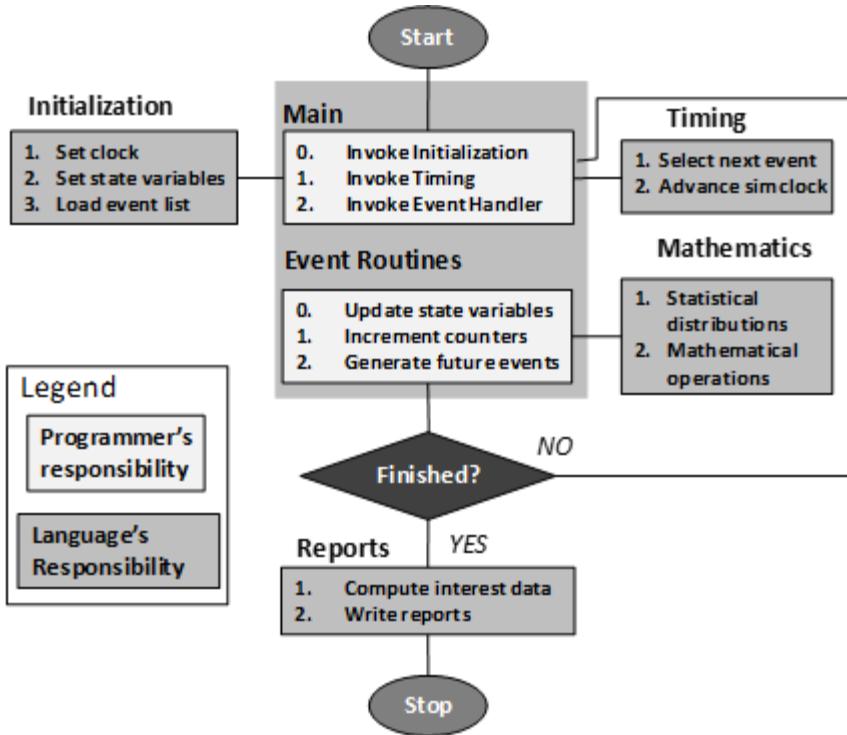


Figure 1: A typical simulation program.

## 18.2. A Single-Server (M/M/1) Queue

The main programs for this system are INITIALIZE and SIMULATE

State variables:

- Number of customers in the queue,  $Q: 0, 1, 2, \dots$
- Status of the server,  $S: \text{idle}$  or  $\text{busy}$
- Since all customers are assumed identical, we do not need to track them individually.

Event Routines:

- Types of events: arrival, begin service, and end service
- NOW is the variable that contains the current system time—the system clock
- Interarrival and service time—routines that return randomly

- sampled values from interarrival and service time distributions
- “Schedule *event-type* at time *T*” means place an event notice with event time *T* and event type *event-type* on the list.
- “Schedule *begin service* at time *NOW*” puts an event notice with time *NOW* and event type *begin service* on the event list.

Initialize the system in the *empty-and-idle state*

### 18.2.1. M/M/1 Queue Event Routines

Event routine arrival

Schedule arrival event at time *NOW* + interarrival time  
 $Q := Q + 1$

IF  $S := \text{idle}$  THEN schedule *begin service* event at time *NOW*

Event routine begin service

$Q := Q - 1$

$S := \text{busy}$

Schedule *end service* event at time *NOW* + service time

Event routine end service

$S := \text{idle}$

IF  $Q > 0$

THEN schedule *begin service* event at time *NOW*

Routine Initialize

$Q := 0$

$S := \text{idle}$

Schedule arrival event at time *NOW*

### 18.2.2. M/M/1 Queue

#### 18.2.2.1 Queue Event Execution and State Changes

The procedure *SIMULATE* is the timing routine that moves the model through time. After *Initialization*, control is turned over to the timing routine, *SIMULATE* which executes the following algorithm:

**SIMULATE**

While the event list is not empty and stop signal has not been received DO:

    Update the clock to the time on the first event notice.

Execute the event routine for the type of event on the first event notice.

Discard the first event notice.

The timing routine before initialization is shown in **Figure 2**.

## Before Initialization

### State Variables

|   |   |     |
|---|---|-----|
| Q | S | NOW |
|   |   |     |

### Event List

| Number | Time | Event Type |
|--------|------|------------|
|        |      |            |

**Figure 2:** The state variables and event list before initialization

When the timing routine takes control, there is one event on the event list as shown **Figure 3**.

### State Variables

|   |      |     |
|---|------|-----|
| Q | S    | NOW |
| 0 | idle | 0.0 |

### Event List

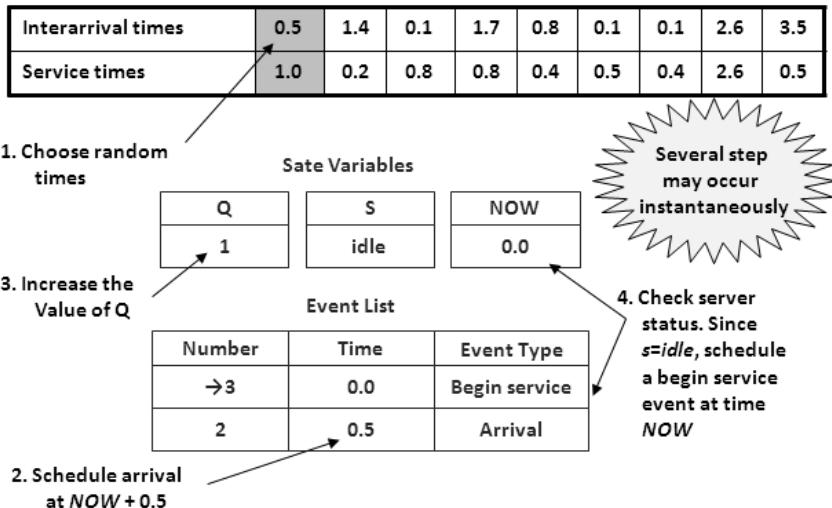
| Number | Time | Event Type |
|--------|------|------------|
| →1     | 0.0  | Arrival    |
| •      | •    | •          |

**Figure 3:** The state variables and event list after initialization

### 18.2.2.2

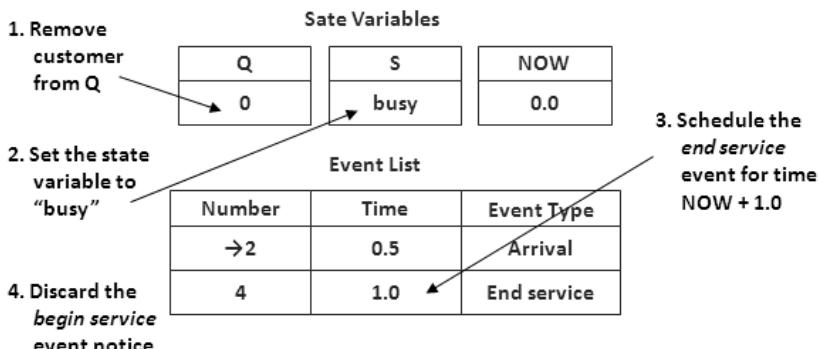
### M/M/1 Queue Step-by-Step Discrete Event Simulation

Randomly sampled observations for interarrival and service times for the single-server queue model are have been generated and appear in **Figure 4**.



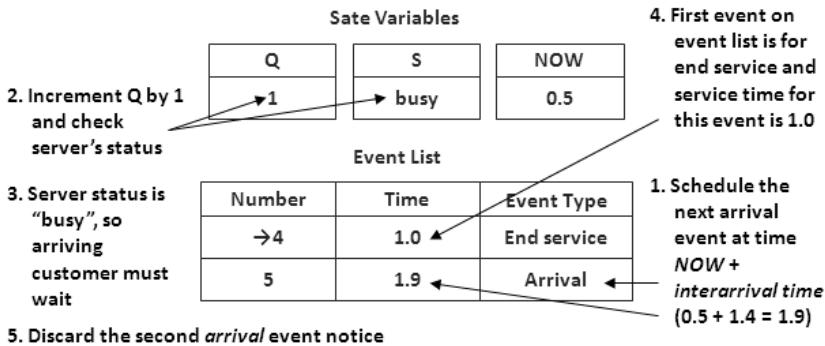
**Figure 4:** Step 1 in the step-by-step simulation of the M/M/1 Queue

The first event notice on the event list now is for the *begin service* event for the first customer. This event occurs at time 0.0, so the clock remains at time 0.0, and the *begin service* event routine is executed. This process is shown in **Figure 5**.



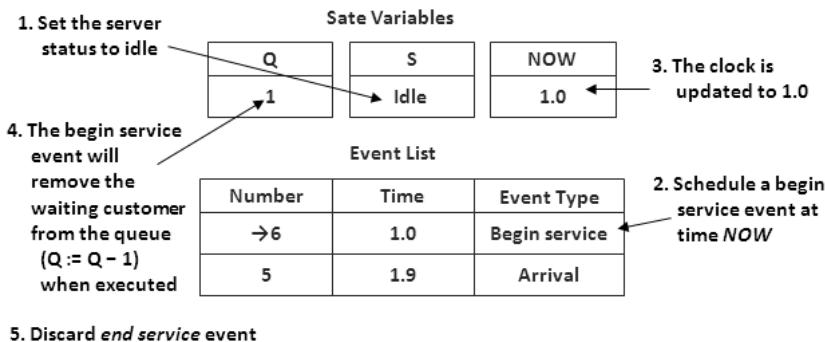
**Figure 5:** Step 2 in the step-by-step simulation of the M/M/1 Queue

Now the first event on the event list is one for the arrival event at time 0.5. After the clock is assigned the event time, 0.5, the arrival event routine is executed. This process is shown in **Figure 6**.



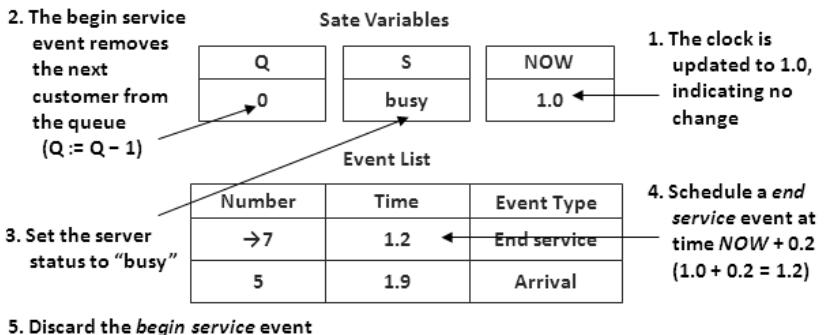
**Figure 6:** Step 3 in the step-by-step simulation of the M/M/1 Queue

The clock is now set for the time of the end service event, 1.0, and the *end service* routine is executed to make the state changes involved with the first customer's completing service. This process is shown in **Figure 7**.



**Figure 7:** Step 4 in the step-by-step simulation of the M/M/1 Queue

The clock is now updated to 1.0, indicating no change, and the begin service event removes the next customer from the queue. This process is shown in **Figure 8**.



© Copyright 2007 Dr. Jeffrey Strickland

**Figure 8:** Step 5 in the step-by-step simulation of the M/M/1 Queue

#### 18.2.2.3 M/M/1 Queue Summary

A summary of the step-by-step simulation of the M/M/1 queue, for 10 events is shown in Figure 10.

Simulation of an M/M/1 Queuing System:

Arriva Rate 0.700

Service Rate 1.000

| Time  | Event Actions(s)   |
|-------|--|
|       | Run trace for 10 events.   |
|       | System initialized to have 0 customers.  |
| 0.000 | Schedule Arrival event at time 0.000<br>-----Start event-----<br>Arrival   |
| 0.500 | Schedule arrival event at time 0.500<br>Add customer to queue. Length is 1<br>Server status is idle.   |
| 0.000 | Schedule Begin Serv event at time 0.000<br>Begin<br>-----Start Event-----<br>Serv  |
| 1.000 | Schedule End Serv event at time 1.000<br>Arrival<br>-----Start Event-----  |
| 1.900 | Schedule Arrival event at time 1.900<br>Add customer to queue. Length is 1<br>Server status is busy.<br>1.000 End<br>-----Start Event-----<br>Serv |
|       | Queue length is 0. Server is idle.   |
|       | Schedule Begin Serv event at time 1.000  |

|                                       |         |
|---------------------------------------|---------|
| 1.000 Begin                           | Serv    |
| -----Start Event-----                 |         |
| Take customer from queue. Length is 0 |         |
| Server is now busy.                   |         |
| Schedule End Serv event at time       | 1.200   |
| 1.200 End                             | Serv    |
| -----Start Event-----                 |         |
| Queue length is 0. Server is idle.    |         |
| Schedule Begin Serv event at time     | 1.900   |
| 1.900                                 | Arrival |
| -----Start Event-----                 |         |
| Schedule Arrival event at time        | 2.000   |
| Add customer to queue. Length is 1    |         |
| Server status is idle.                |         |
| 1.900 Begin                           | Serv    |
| -----Start Event-----                 |         |
| Take customer from queue. Length is 0 |         |
| Server is now busy.                   |         |
| Schedule End Serv event at time       | 2.000   |
| 2.000 End                             | Serv    |
| -----Start Event-----                 |         |
| Queue length is 0. Server is idle.    |         |
| Schedule Begin Serv event at time     | 2.800   |
| 2.000                                 | Arrival |
| -----Start Event-----                 |         |
| Schedule Arrival event at time        | 3.700   |
| Take customer from queue. Length is 1 |         |
| Server is now busy.                   |         |
| Schedule End Serv event at time       | 3.200   |

**Figure 9:** Summary of the step-by-step simulation for 10 events

#### 18.2.2.4 Stopping the Simulation

This process continues until the timing routine breaks out of its loop. If the event routines are executed as given and no signal is sent to the timing routine to quit, then the process will go on forever. Because each *arrival* event schedules the next *arrival* event, the event list always contains one *arrival* event. If we want to stop the simulation after time 10.0 is executed, we replace the statement

Schedule arrival event at time NOW + interarrival time.

With the statement:

```
IF NOW < 10.0
THEN schedule arrival event at time NOW + interarrival
time.
```

### 18.2.3. Event Execution and State Changes Summary

For the event scheduling time advance mechanism to work properly, two things must be done;

1. At least one event must be scheduled when the system is initialized.
2. At least some events must schedule other events.

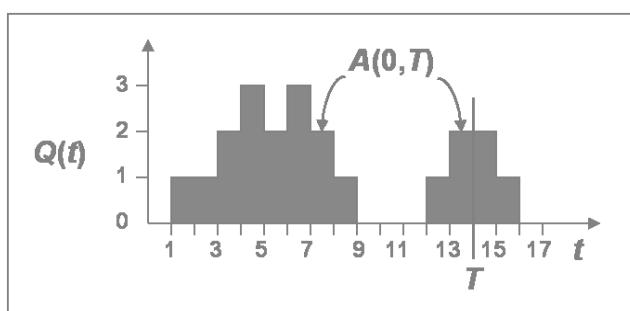
Because each *arrival* event schedules the next *arrival* event at time *NOW* + *interarrival time*, the time *between* arrival events is guaranteed equal to the interarrival time. If the interarrival times are *iid* random variables, this guarantees that times between arrival events are iid.

### 18.2.4. Estimating Mean Number in Queue

The purpose of the simulation is to estimate the mean number of customers in the queue when the system is at steady state. We will discuss estimating the mean or average in more detail in Chapter 3.

The number of customers in the queue, or queue length, at any point in time is contained in the system variable  $Q$ . The parameter we wish to estimate can be expressed as the average area under the graph of  $Q(t)$  (as depicted in **Figure 10**):

$$\mu = \lim_{T \rightarrow \infty} \frac{A(0, T)}{T} \quad (18.3)$$



**Figure 10:** The average area under the graph of  $Q(t)$

Since the system will not reach steady state until after the initial transient period at some point  $t < t_s$ , the point estimate for  $\mu$  in Equation 18.3, computed from  $T$  time units of data, is:

$$\hat{\mu}_T = \frac{A(t_s, t_s + T)}{T}$$

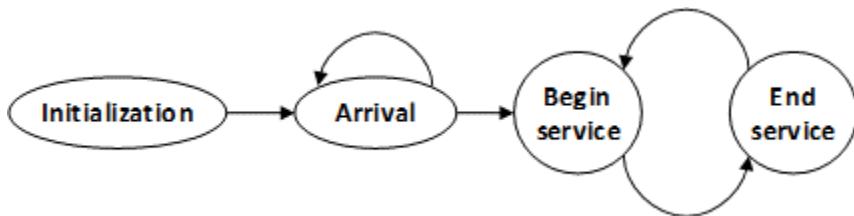
where (18.4)

$$A(t_s, t_s + T) = \sum_{k=1}^K Q_k(t_k - t_{k-1}) \quad (18.5)$$

and where  $Q_k$  represents the number in the queue between the  $(k-1)^{st}$  and  $k^{th}$  events.

### 18.2.5. Event Graphs

The relationship among various events can be displayed graphically using an event graph. The event graph for the M/M/1 queue is shown in **Figure 11**.



**Figure 11:** The event graph for the M/M/1 Queue

See Schruben (1995) and Sargent (1988).

### 18.3. Summary

In a **discrete-event dynamic system**, the state changes only at discrete points in time, which mark the occurrence of an **event**.

An **event** is any occurrence that causes an instantaneous change in the state of the system.

In any system there are only a few distinct type of events.

In the simple queuing system there are only three types of events:

- Arrival of a customer
- Beginning of service
- Completion of service

**Event routines** (*transition functions*) specify exactly how a system changes when an event occurs.

*ExtendSim* has modules that allow us to program event routines with simple modeling constructs, to generate future events and update state variables.

## 18.4. Exercises

1. Perform two more steps of the by-hand simulation presented in this chapter, using the random number provided on page 658.
2. Estimate the average mean number in the queue based on your observation from the by hand simulation, and calculate the point estimate using Equations 18-2 and 18-3.

## 18.5. Discrete Event Simulation using *SCILAB*

Queuing systems can conveniently be modeled in *SCILAB* with **Xcos** using the event mechanism. Xcos is a graphical editor to design hybrid dynamical systems models. Xcos is loaded by typing `xcos` at the command prompt. Doing this brings up a palette window (Figure 1) and a modeling window (see Figure 2). By the way the, image at the top is a simulation that checks matrix output to determine if a signal is properly constructed.

Models can be designed, loaded, saved, compiled and simulated. Consider a simple queuing system with a Poisson arrival process and an exponential service time. This would model a queue at the checkout counter of a store. Customers arrive following a Poisson process law, and the time each customer spends at the counter is assumed to have an exponential law.

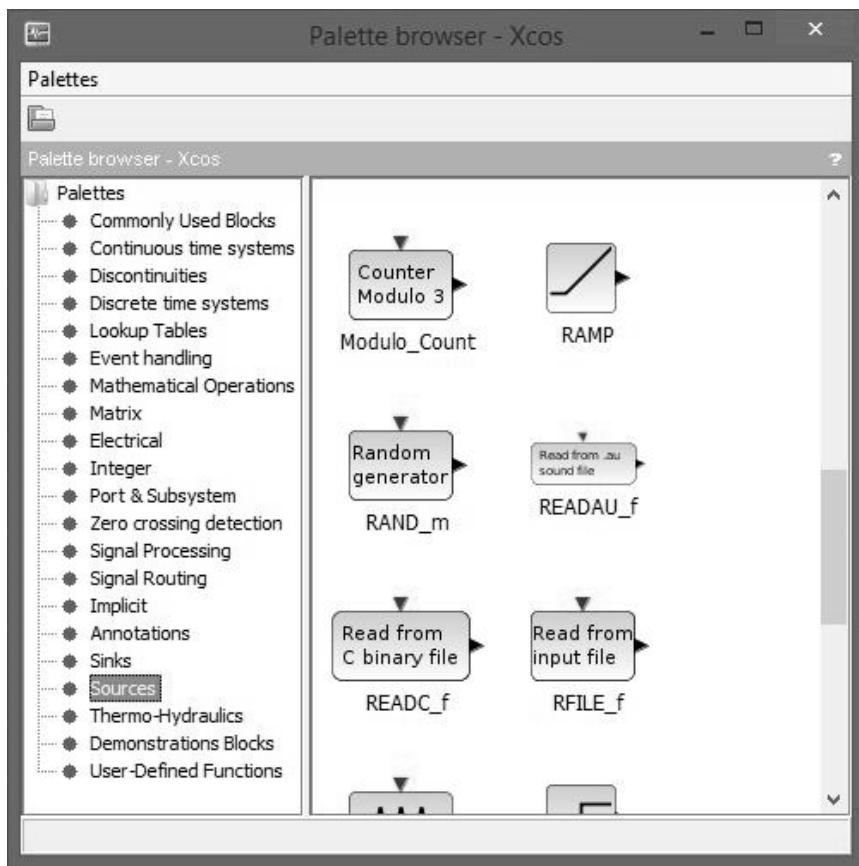
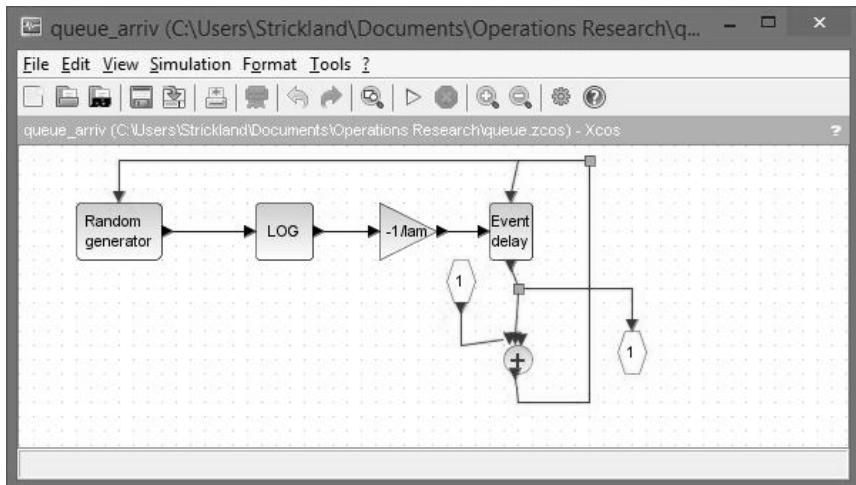


Figure 2. The modeling object pallet

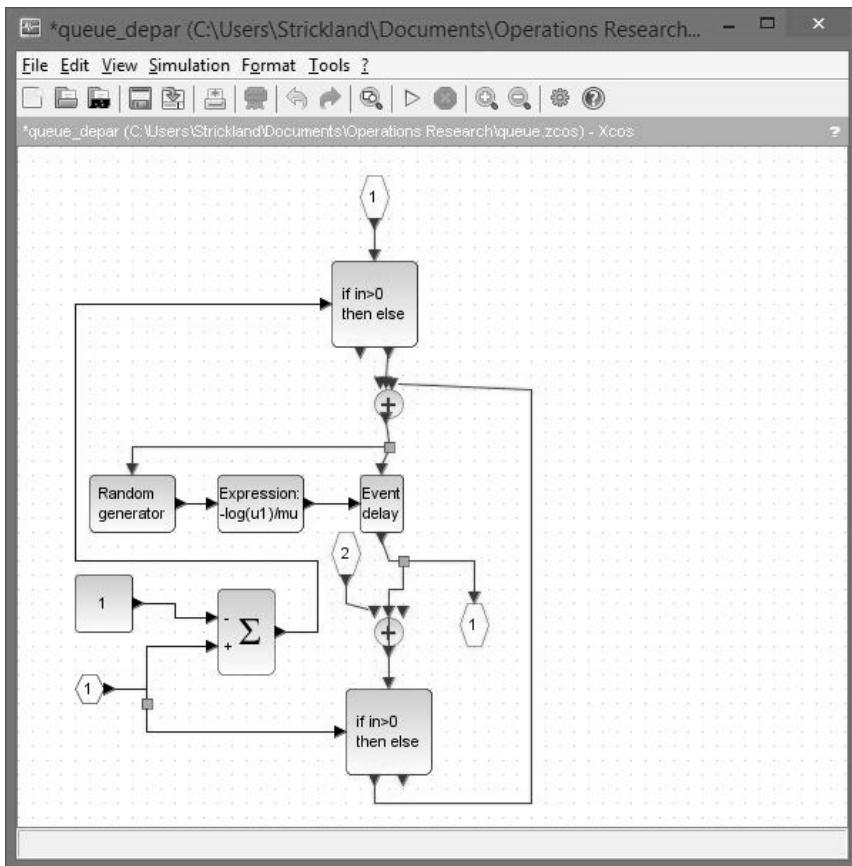
To generate the arrival process, we use the model in Figure 3. The **Random (number) generator** block (from the Sources palette) outputs a uniformly distributed random number between 0 and 1. By taking the **Logarithm** (from the Mathematical Operations palette) and multiplying by  $-1/\lambda$  (use a **Gain** block from the Mathematical Operations palette), we generate a random number with an exponential law with parameter  $\lambda$ . By feeding the output of the **Event Delay** block from the Event handling palette) to its input activation port (from the Port & Subsystems palette), we generate a sequence of events where the time between two events is an independent random variable with exponential law. The result is a Poisson process with parameter  $\lambda$ . The input activation to the Super block starts the process (no event firing is initially programmed on

the output of the event delay block). Superblocks are sub-model objects and are formed by selecting the blocks that will be in the sub-model and choosing **Edit | Selection to superblock**. The "plus (+)" block in Figure 2 is the **Clock summation** block found in the Event handling palette. If you right click on any of these blocks you can find help for the block.



**Figure 3.** The arrival process sub-model (in the modeling window)

Generating the departure process (process of serving a customer) is more complicated. As long as the queue is not empty, the time between two events is an independent random variable. But when the queue is empty, no departure event should be generated. The process should be inactive. To implement this process, we have to allow the restart of the process by an external event (the arrival of a customer in an empty queue). This is the reason why the model of the departure events in Figure 4 has two input activations. One is for initialization and the other is for restart. Note the use of the **Mathematical Expression** block (from the User defined functions palette) to allow for modeling more general waiting laws. But in this example we have used  $-\log(u1)/\mu$ , which results in an exponential law with parameter  $\mu$ . The **Summation** block is from the Mathematical Operations palette and the **Constant input** block is from the Sources palette. The **If-Then-Else** block is from the Event handling palette.



**Figure 4.** The departure process sub-model

The complete model is given in Figure 5, where the state of the queue is stored in the  $1/z$  block. Its value goes up or down by one, depending on the event. Figure 6 Shows the simulation setup. The setting value are in scientific notation. The Solver kind id LSofar. Called by xcos, LSodar (short for Livermore Solver for Ordinary Differential equations, with Automatic method switching for stiff and nonstiff problems, and with Root-finding) is a numerical solver providing an efficient and stable variable-size step method to solve Initial Value Problems of the form:

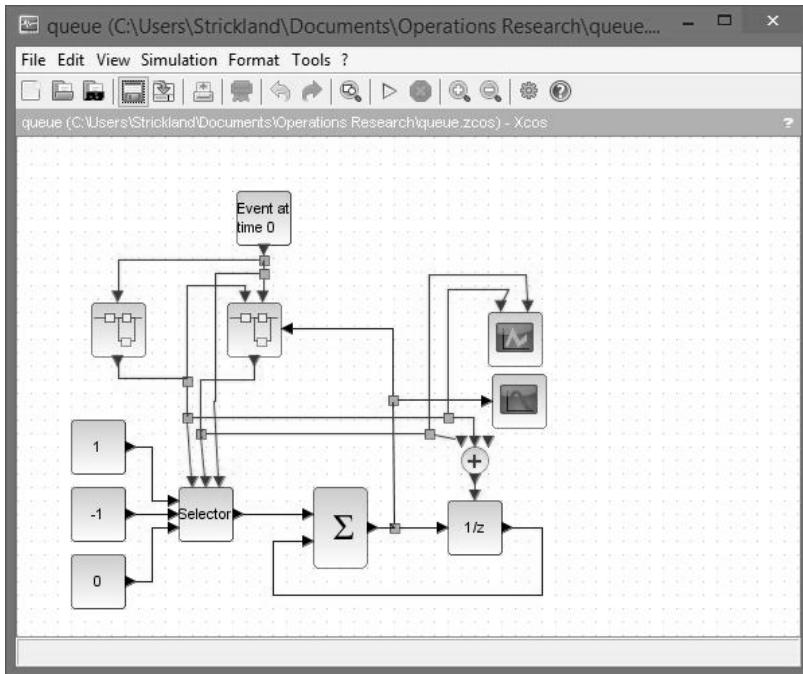
$$\dot{y} = f(y, t), y(t_0) = y_0, y \in \mathbb{R}^n$$

The simulation result is given in Figure 8 (**Single display scope** in the Sinks palette), where we see the evolution of the queue, and in Figure 8

(**Event scope** in the Events handling palette), where the events are displayed (dark lines correspond to arrival events). The parameters are defined in the context as follows:

```
Lam = 0.3;  
Mu = 0.35;  
z0 = 6;
```

where  $z_0$  is used as the initial state of the  $1/z$  block.



**Figure 5.** The complete model

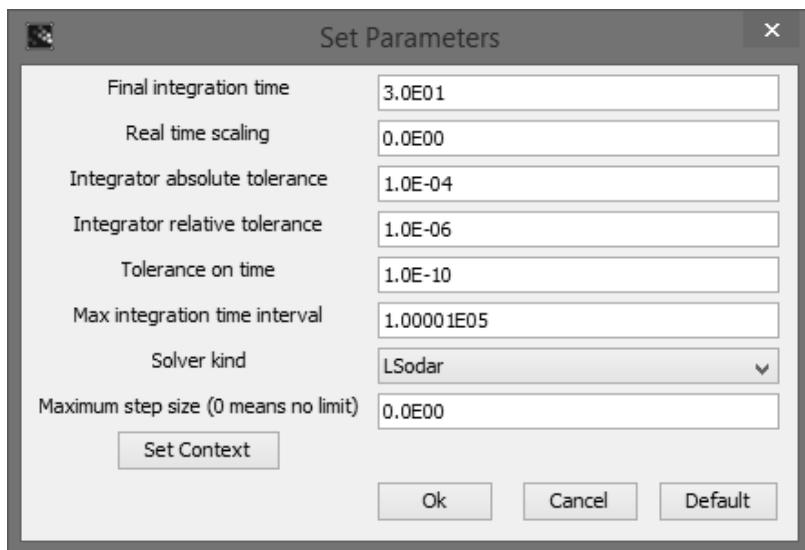


Figure 6. The Simulation Setup dialog window.

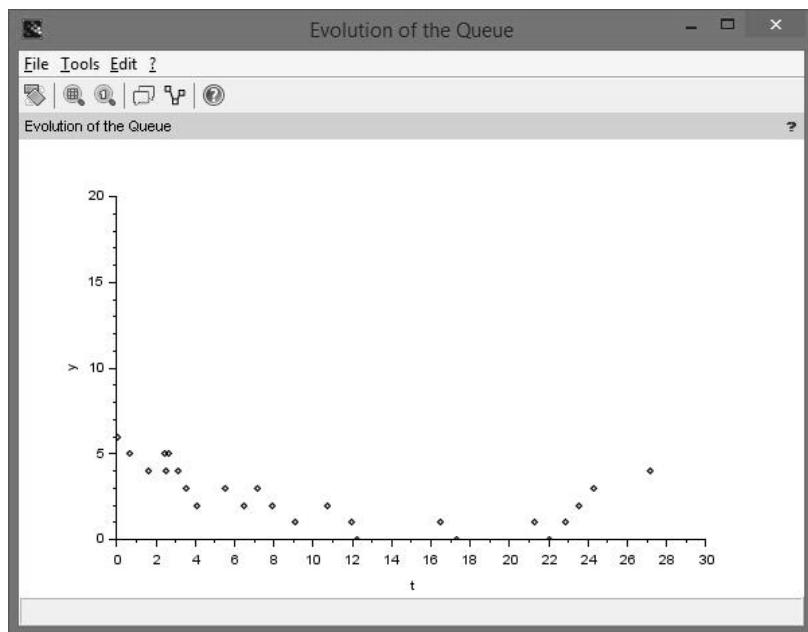
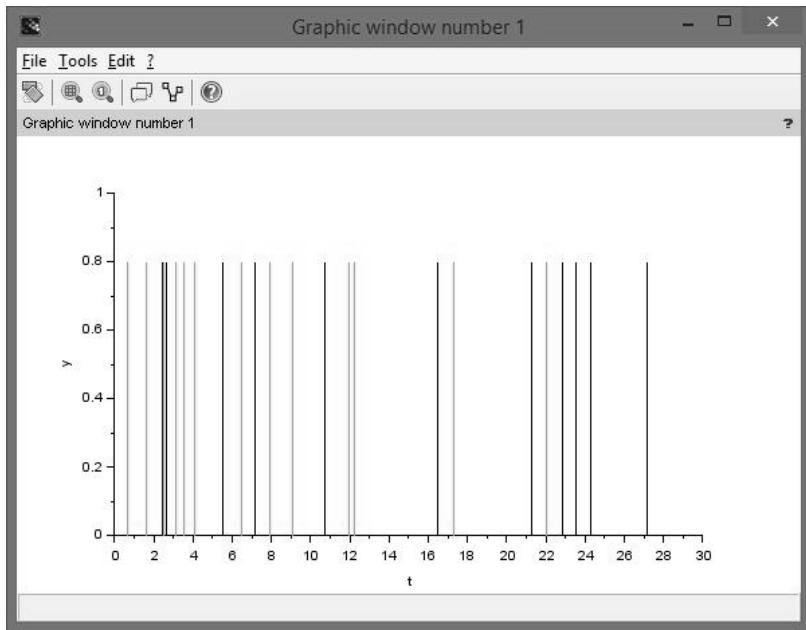


Figure 7. The evolution of the queue



**Figure 8.** The arrival and service events

Later today I will have the complete model uploaded to my website so that it can be downloaded. THE END!

## 18.6. Review Problems

15. Rebuild the discrete event simulation in Section 18.5 using *SCILAB* and Xcos.
16. **Bank Lobby.** A bank lobby has four tellers – Alice, Mary, Jeff, and Doris – with similar working characteristics. The customer arrival process varies over time. The average number of arrivals per hour is 10, 20, 40, 36, 27, 32, 18, and 4 for each of the 8 one-hour periods from bank opening to closing. During each period the arrival process is Poisson. In addition, customers can arrive in groups of more than one. For each arrival instance, there is:
  - a 75% probability that it is a single customer
  - a 20 % probability that the group consists of two customers
  - a 5% probability that the group consists of three customers

A single queue servers all four tellers. When a customer enters the lobby, he or she will join the queue if the total number of customers in the lobby—that is the number of customers being served plus the number waiting in the queue—is less than 10. Otherwise, the customer will balk. The service time for the customer depends on the number of transaction to be processed for the customer. Processing time for each transaction has an Erlang distribution with mean 1.08 and the number of stages (parameter k) equal to 2. Eight hours of the bank lobby operation will be simulated: from 9 A.M. to 5 P.M. The probability distribution of number of transactions per customer is:

|                        |    |    |    |    |   |   |
|------------------------|----|----|----|----|---|---|
| Number of transactions | 1  | 2  | 3  | 4  | 5 | 6 |
| Probability (%)        | 20 | 30 | 22 | 15 | 8 | 5 |

Construct a discrete event simulation for this situation using *SCILAB* and Xcos.

17. The Circuit Card Manufacturing Process. The existing process is as follows:

- The company uses five types of circuit cards which arrive in fixed-size batches according to a predefined schedule.
- The schedule for each card type repeats every 120 minutes. For example:

| Card type | Arrives at (minutes from start) | Batch size |
|-----------|---------------------------------|------------|
| 1         | 0                               | 20         |
| 2         | 20                              | 30         |
| 3         | 40                              | 25         |
| 4         | 60                              | 30         |
| 5         | 80                              | 25         |
| 1         | 120                             | 20         |
| 2         | 140                             | 30         |

- The first operation is a single cleaning station that requires at

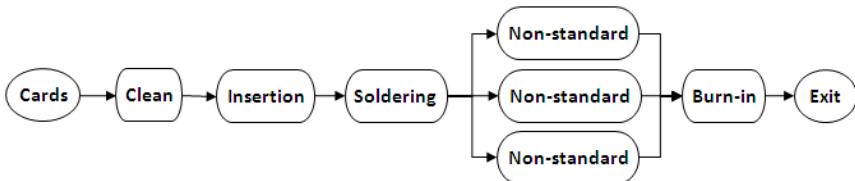
least 36 seconds, at most 54 seconds, and usually 48 seconds to clean each piece.

- The clean cards are put onto an automatic insertion machine which installs most of the elec-tronic components. This machine can work on up to 6 circuit cards at one time and requires 5 minutes to process each card.

Once the components have been installed they are soldered into place in a 10 foot long wave solder machine which has a capacity of 30 cards on its belt and moves at 1 foot per minute. There are three stations which install any nonstandard components that the automatic insertion machine is unable to handle. The amount of time required for this operation varies by card type:

| Card type process time (minutes) |     |
|----------------------------------|-----|
| 1                                | 2.5 |
| 2                                | 2.0 |
| 3                                | 2.5 |
| 4                                | 3.0 |
| 5                                | 2.0 |

The final step is a burn-in test. In this process, the cards are combined into groups of 24 which are put into an oven and power-cycled for 20 minutes. A process flow model is shown in Figure 7.



**Figure 7.** Simple Process Diagram



## Appendix A – Installing R

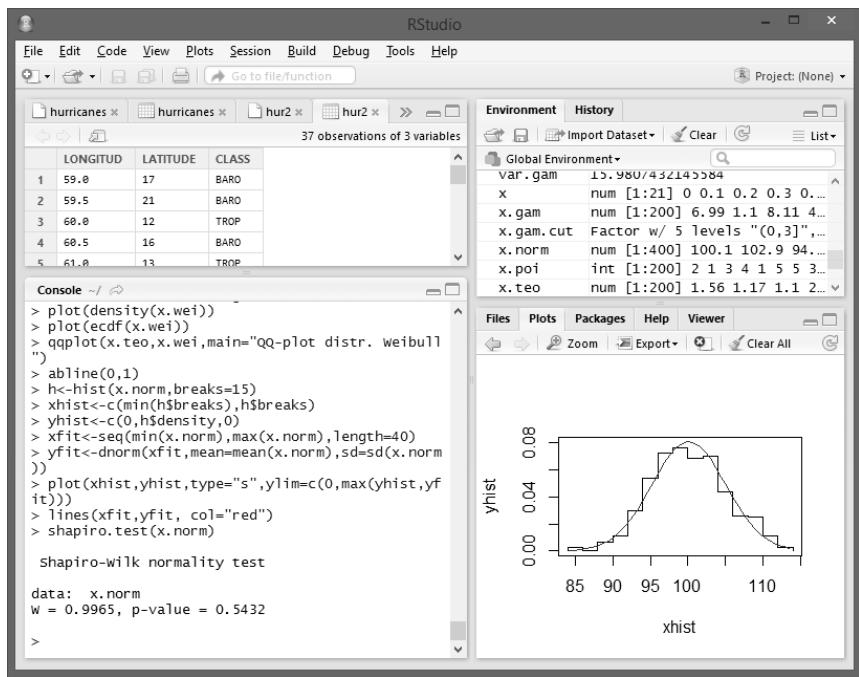
---

I assume that you will use a PC or MAC, that you will perform the installation on a hard disc, and that you have access to the internet. First go to the site called CRAN (Comprehensive R Archive Network). The URL address is <http://cran.r-project.org/> or simply type CRAN into Google. Once you are at the website, you need to ‘Download and Install R’ by running the appropriate precompiled binary distributions. Click to choose between Linux, Mac Os or Windows, and then follow the instructions. You want the ‘base’ package and you want to run the setup program which will have a name like R\*.exe (on a OC) or R\*.dmg (on a Mac). When asked, say you want to ‘Run’ the file (rather than ‘Save’ it).

You can begin using R without any further actions. However, I use a Graphical User Interface (GUI) called *R Studio*, and I recommend that you do the same, or spend some frustrating hours at the onset. You can get *R Studio* at this URL:

<http://www.rstudio.com/products/rstudio/download/>

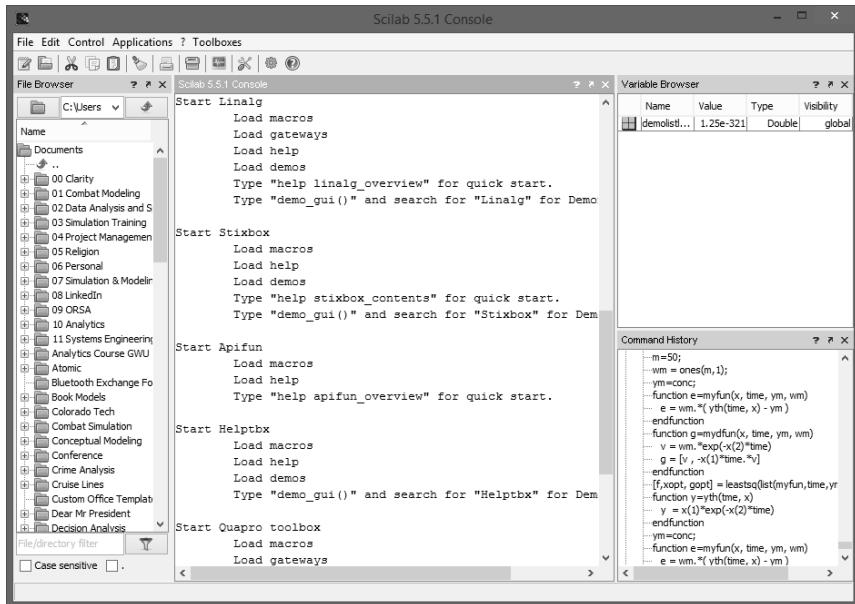
Alternatively, you can type *R Studio* into Google. Select the installation for your system as you did with R. When you run R Studio, you will see something like Figure 1, except the workspace will not have R code and there will be no graph. Plus I reduced the size for the screenshot.



**Figure 1.** R Studio window with four panes (from upper left): Data view, Environment/History view, Console (this is where you enter R code), and Viewer (for Plots, Help, Packages and Files)

## Appendix B – Installing *SCILAB*

To download *SCILAB*, go to <http://www.SCILAB.org/download/5.5.1> (version 5.5.1 is the latest as of the time this book was published) or simply type *SCILAB* into Google and go from there. Again, select the installation that is appropriate for your system (Mac or PC). If you are not sure whether to install the 32-bit or 64-bit version, then start with the 32-bit version. Your system will inform you which one you really need upon installation (if you are installing the wrong one). The GUI for *SCILAB* is native to the program, so there is no additional installation. Figure 1 shows the *SCILAB* window (I reduced the size for a screenshot).



**Figure 1.** *SCILAB* window with four panes (from left to right): Directory, Console, Variable Browser and Command History.

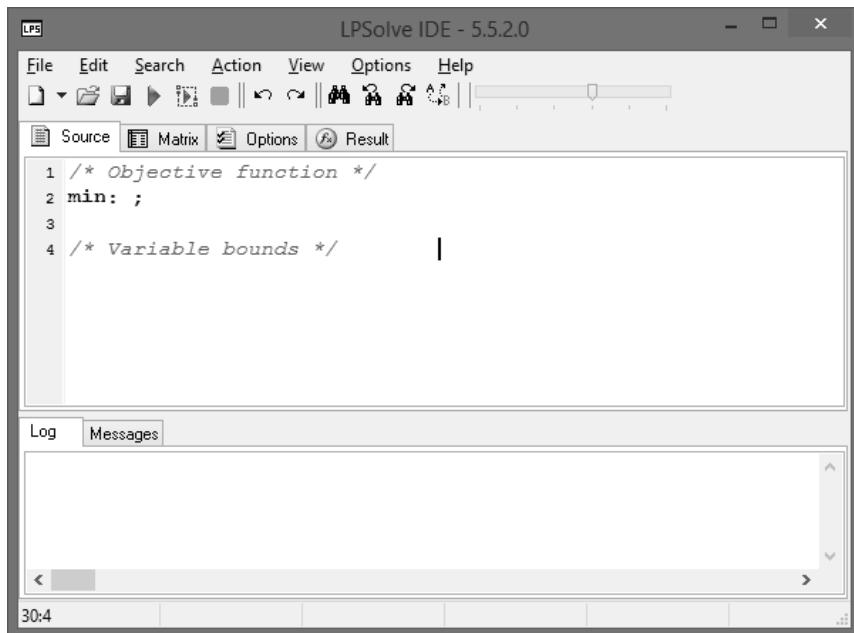
## Appendix C – Installing LPSolve

---

To install LPSolve, go to the following URL:

<http://LPSolve.sourceforge.net/5.5/distribution.htm>

You can also type LPSolve into Google. You will want to install the LPSolve IDE (Integrated Development Interface) for Windows. If you are using a Mac, you are at a disadvantage (as is the case with many Operations Research software packages). You can still download and use LPSolve, but you will just have the Console (no GUI). Alternatively, you could download the appropriate version and follow the rather complex instructions for calling LPSolve from MATLAB, Octave, Python, etc. The Windows IDE is shown in Figure 1.



**Figure 1.** LPSolve IDE in Windows with two panes (top to bottom): Console (Source, Matrix, Options and Results) and Log/Messages

# References

---

- Albright, C.S. and Winston, W.L. (2005). *Spreadsheet Modeling and Application: Essentials of Practical Management Science*. Wadsworth Group. ISBN 0-534-38032-8
- Ali M. (2000). "Probabilistic Systems Analyses: Part I," University of the Witwatersrand, 2000.
- Crawley, M.J. (2007). *The R Book*. John Wiley & Sons, Ltd. ISBN 978-0-470-51024-7
- Evans, Hastings, and Peacock (2000), *Statistical Distributions*, 3rd. Ed., John Wiley and Sons.
- Hayter, A.J. (2002). *Probability and Statistics for Engineers and Scientists*. 2<sup>nd</sup> Edition. Wadsworth Group. ISBN 0-534-38669-5.
- Stephens, M. A. (1974). EDF Statistics for Goodness-of-fit and Some Comparisons, *Journal of the American Statistical Association*, Vol. 69, pp. 730-737.
- Tukey, John (1977), *Exploratory Data Analysis*, Addison-Wesley.
- Schruben, L. (1995). *Graphical Simulation and Modeling using SIGMA for Windows*, 3rd Edition. The Scientific Press.
- Sargent, R. G. (1988). Event graph modeling for simulation with an application to flexible manufacturing systems. *Management Science*, 34 1231–1251.
- Strickland, J.S. (2010). *Discrete Event Simulation Using ExtendSim 8*. Lulu.com. ISBN 978-0-557-72821-3.
- Strickland, J.S. (2015). *Predictive Analytics using R*. Lulu.com. ISBN 978-1-312-84101-7.
- Winston, W.L. (2004). *Operations Research Applications and Algorithms*. 4<sup>th</sup> Edition. Wadsworth Group. ISBN 0-534-38058-1.

## Works Cited

---

Akbarzadeh, V. (2013). *Line Estimation*. From Machine Learning Made Simple: <http://mlmadesimple.com/2014/05/07/line-estimation/>

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185. doi:10.1080/00031305.1992.10475879

Altschul, J. H., Sebastian, L., & Heidelberg, K. (2004). *Predictive Modeling in the Military: Similar Goals, Divergent Paths*. SRI Foundation.

Amemiya, T. (1985). *Advanced econometrics*. Boston: Harvard University Press.

Bailey, K. (1994). Numerical Taxonomy and Cluster Analysis. In *Typologies and Taxonomies* (p. 34).

Barkin, E. (2011). *CRM + Predictive Analytics: Why It All Adds Up*. New York: Destination CRM. Retrieved 2014 from <http://www.destinationcrm.com/Articles/Editorial/Magazine-Features/CRM---Predictive-Analytics-Why-It-All-Adds-Up-74700.aspx>

Ben-Gal I. Dana A., S. N. (2014). Efficient Construction of Decision Trees by the Dual Information Distance Method. *Quality Technology & Quantitative Management (QTQM)*, 11(1), 133-147. From <http://www.eng.tau.ac.il/~bengal/DID.pdf>

Bhandari, M., & Joensson, A. (2008). *Clinical Research for Surgeons*. Thieme.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. From <http://www.hua.edu.vn/khoa/fita/wp-content/uploads/2013/08/Pattern-Recognition-and-Machine-Learning-Christophe-M-Bishop.pdf>

Bliss, C. I. (1934). The Method of Probits. *Science*, 79(2037), 38–39. doi:10.1126/science.79.2037.38

Boyd, C. R., Tolson, M. A., & Copes, W. S. (1987). Evaluating trauma care: The TRISS method. Trauma Score and the Injury Severity Score. *The Journal of trauma*, 27(4), 370–378.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123-140. From <http://link.springer.com/article/10.1023%2FA%3A1018054314350>

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. doi:10.1023/A:1010933404324

Burnham, K. P., & Anderson, D. (2002). *Model Selection and Multi-Model Inference*. New York: Springer.

Cattell, R. B. (1943). The description of personality: Basic traits resolved into clusters. *Journal of Abnormal and Social Psychology*, 38, 476–506. doi:doi:10.1037/h0054116

Chen, J., Zhu, J., Wang, Z., Zheng, X., & Zhang, B. (2013). Scalable Inference for Logistic-Normal Topic Models. In *Advances in Neural Information Processing Systems* (p. 2445{2453).

Chib, S., & Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4). doi:10.1080/00031305.1995.10476177

Chong, J., Jin, Y., & Phillips, M. (2013). *The Entrepreneur's Cost of Capital: Incorporating Downside Risk in the Buildup Method*. Pasadena: MacroRisk.com. Retrieved 2014 from <http://www.macrorisk.com/wp-content/uploads/2013/04/MRA-WP-2013-e.pdf>

Cohen, J., Cohen, P., West, S., & Aiken, L. (2002). Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences (3rd ed.). Routledge.

Conz, N. (2008). *Insurers Shift to Customer-focused Predictive Analytics Technologies*. New York: Insurance & Technology. Retrieved 2014 from <http://www.insurancetech.com/business-intelligence/insurers-shift-to-customer-focused-predi/210600271>

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273. doi:10.1007/BF00994018

Da, Y., & Xiurun, G. (2005). An improved PSO-based ANN with simulated annealing technique. In T. e. Villmann, *New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks*. New York: Elsevier.

doi:10.1016/j.neucom.2004.07.002

Das, K., & Vidyashankar, G. (2006). *Competitive Advantage in Retail Through Analytics: Developing Insights, Creating Value*. New York: Information Management. Retrieved 2014 from <http://www.information-management.com/infodirect/20060707/1057744-1.html>

Davidson, R., & Mackinnon, .. G. (1993). *Estimation and inference in econometrics*. Oxford University Press.

Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal (British Computer Society)*, 20(4), 364–366. doi:doi:10.1093/comjnl/20.4.364

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag.

Draper, N., & Smith, H. (1998). *Applied Regression Analysis* (3rd ed.). New York: John Wiley.

Eckerson, W. (2007). *Extending the Value of Your Data Warehousing Investment*. The Data Warehouse Institute. Retrieved 2014 from [http://tdwi.org/articles/2007/05/10/predictive-analytics.aspx?sc\\_lang=en](http://tdwi.org/articles/2007/05/10/predictive-analytics.aspx?sc_lang=en)

Estivill-Castro, V. (2002, June 20). Why so many clustering algorithms — A Position Paper. *ACM SIGKDD Explorations Newsletter*, 4(1), 65–75. doi:doi:10.1145/568574.568575

Fletcher, H. (2011). *The 7 Best Uses for Predictive Analytics in Multichannel Marketing*. Philadelphia: Target Marketing. Retrieved 2014 from <http://www.targetmarketingmag.com/article/7-best-uses-predictive-analytics-modeling-multichannel-marketing/1#>

Freund, Y., & Schapire, R. E. (1999). Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3), 277. doi:10.1023/A:1007662407062

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1), 1-67. doi:10.1214/aos/1176347963

Geisser, S. (1993). *Predictive Inference: An Introduction*. New York: Chapman &

Hall.

George, E., & McCulloch, R. (1993). Variable Selection via Gibbs Sampling. *Journal of the American Statistical Association*, 88(423). doi:10.1080/01621459.1993.10476353

Greene, W. (2011). *Econometric Analysis* (7th ed.). New York: Prentice Hall.

Hale, G., & Hale, G. (2006). *Uneasy Partnership: The Politics of Business and Government in Canada*. Toronto: University of Toronto Press.

Hardin, J., & Hilbe, J. (2003). *Generalized Estimating Equations*. London: Chapman and Hall/CRC.

Harrell, F. (2010). *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York: Springer-Verlag.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized Additive Models*. London: Chapman & Hall/CRC.

Hayashi, F. (2000). *Econometrics*. Princeton: Princeton University Press.

Hayward, R. (2004). Clinical decision support tools: Do they support clinicians? *FUTURE Practice*, 66-68.

Hopfield, J. (2007). Hopfield network. *Scholarpedia*, 2(5), 1977. doi:10.4249

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression* (2nd ed.). New York: Wiley-Interscience Publication.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York: Springer.

Johnson, M., & Raven, P. (1973). Species number and endemism: The Galapagos Archipelago revisited. *Science*, 179, 893-895.

King, G. (2014, January). Restructuring the Social Sciences: Reflections from Harvard's Institute for Quantitative Social Science. *PS*, 165-172. doi:doi:10.1017/S1049096513001534

Kohonen, T., & Honkela, T. (2007). Kohonen Network. *Scholarpedia*, 2(1), 1568. doi:10.4249

- Korn, S. (2011). *The Opportunity for Predictive Analytics in Finance*. San Diego: HPC Wire. Retrieved 2014 from [http://www.hpcwire.com/2011/04/21/the\\_opportunity\\_for\\_predictive\\_analytics\\_in\\_finance/](http://www.hpcwire.com/2011/04/21/the_opportunity_for_predictive_analytics_in_finance/)
- Luenberger, D. (1984). *Linear and Nonlinear Programming* (2nd ed.). Reading, MASS: Adisin-Wesley.
- Łukaszyk, S. (2004). A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, 33, 299-300. Retrieved 2014
- Mark, J., & Goldberg, M. (2001). Multiple Regression Analysis and Mass Assessment: A Review of the Issues. *The Appraisal Journal*, 89–109.
- McCullagh, P., & Nelder, J. (1989). *Generalized Linear Models* (2nd ed.). Boca Raton: Chapman and Hall/CRC.
- McDonald, M. (2010). *New Technology Taps 'Predictive Analytics' to Target Travel Recommendations*. Oyster Bay: Travel Market Report. Retrieved 2014 from <http://www.travelmarketreport.com/technology?articleID=4259&LP=1>
- McKay, L. (2009, August). The New Prescription for Pharma. *Destination CRM*. Retrieved 2014 from <http://www.destinationcrm.com/articles/Web-Exclusives/Web-Only-Bonus-Articles/The-New-Prescription-for-Pharma-55774.aspx>
- Menard, S. (2002). *Applied Logistic Regression* (2nd ed.). SAGE.
- Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.
- Myers, J. H., & Forgy, E. W. (1963). The Development of Numerical Credit Evaluation Systems. *J. Amer. Statist. Assoc.*, 49(12), 799–806. doi:10.1080/01621459.1963.10500889
- Nelder, J., & Wedderburn, R. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3), 370–384. doi:10.2307/2344614
- Nyce, C. (2007). *Predictive Analytics White Paper*. Malvern: American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America.

Retrieved 2014 from  
<http://www.theinstitutes.org/doc/predictivemodelingwhitepaper.pdf>

Palei, S. K., & Das, S. K. (2009). Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach. *Safety Science*, 47, 88. doi:10.1016/j.ssci.2008.01.002

Parry, G. (1996, November–December). The characterization of uncertainty in Probabilistic Risk Assessments of complex systems. *Reliability Engineering & System Safety*, 54(2-3), 119–1. Retrieved 2014 from  
<http://www.sciencedirect.com/science/article/pii/S0951832096000695>

Peduzzi, P., Concato, J., Kemper, E., Holzner, T., & Feinstein, A. (1996). A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology*, 49(12), 1373–9. doi:10.1016/s0895-4356(96)00236-3

Prentice, R., & Pyke, R. (1979). Logistic disease incidence models and case-control studies. *Biometrika*, 66(3), 403-411. doi:10.1093/biomet/66.3.403

Prinzie, A. V. (2008). Random Forests for multiclass classification: Random MultiNomial Logit. *Expert Systems with Applications*, 34(3), 1721–1732. doi:10.1016/j.eswa.2007.01.029

Rao, C. (1973). *Linear statistical inference and its applications* (2nd ed.). New York: John Wiley & Sons.

Rennie, J., Shih, L., Teevan, J., & Karger, D. (2003). *Tackling the poor assumptions of Naive Bayes classifiers*. ICML. Retrieved 2014 from  
<http://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf>

Riedmiller, M. (2010). *Machine Learning: Multi Layer Perceptrons*. Albert-Ludwigs-University. Freiburg: AG Maschinelles Lernen. Retrieved 2014 from  
[http://ml.informatik.uni-freiburg.de/\\_media/teaching/ss10/05\\_mlps.printer.pdf](http://ml.informatik.uni-freiburg.de/_media/teaching/ss10/05_mlps.printer.pdf)

Rokach, L., & Maimon, O. (2008). *Data mining with decision trees: theory and applications*. World Scientific Pub Co Inc.

Rosenblatt, F. (1958). Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. *Psychological Review*, 65(6), 386–408.

doi:10.1037/h0042519

Schiff, M. (2012, March 6). *BI Experts: Why Predictive Analytics Will Continue to Grow*. Renton: The Data Warehouse Institute. Retrieved 2014 from <http://tdwi.org/Articles/2012/03/06/Predictive-Analytics-Growth.aspx?Page=1>

Senn, S. (2003). A conversation with John Nelder. *Statistical Science*, 18(1), 118–131. doi:10.1214/ss/1056397489

Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)*, 16(1), 30–34. doi:doi:10.1093/comjnl/16.1.30

Stasinopoulos, M., & Rigby, R. (2007). Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, 23(7). From <http://www.jstatsoft.org/v23/i07>

Stevenson, E. (2011, December 16). Tech Beat: Can you pronounce health care predictive analytics? *Times-Standard*. Retrieved 2014 from [http://www.times-standard.com/business/ci\\_19561141](http://www.times-standard.com/business/ci_19561141)

Steyerberg, E. W. (2010). *Clinical Prediction Models*. New York: Springer.

Strano, M., & Colosimo, B. (2006). Logistic regression analysis for experimental determination of forming limit diagrams. *International Journal of Machine Tools and Manufacture*, 46(6). doi:10.1016/j.ijmachtools.2005.07.005

Strickland, J. (2005). *Random Variate Generation: @Risk Simulation Modeling*. Fort Lee: US ALCM. From [http://simulation-educators.com/uploads/2/7/7/2/2772366/sim\\_04.pdf](http://simulation-educators.com/uploads/2/7/7/2/2772366/sim_04.pdf)

Strickland, J. (2013). *Introduction to Crime Analysis and Mapping*. Lulu.com. From <http://www.lulu.com/shop/jeffrey-strickland/introduction-to-crime-analysis-and-mapping/paperback/product-21628219.html>

The CLUSTER Procedure: Clustering Methods. (n.d.). In *SAS/STAT 9.2 Users Guide*. SAS Institute. Retrieved Retrieved 2014-12-26

The DISTANCE Procedure: Proximity Measures. (n.d.). In *SAS/STAT 9.2 Users Guide*. SAS Institute. Retrieved Retrieved 2014-12-26

Tryon, R. C. (1939). Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality. Edwards Brothers.

Walsh, B. (2004, April 26). *Markov Chain Monte Carlo and Gibbs Sampling*. From Lecture Notes for EEB 581: <http://web.mit.edu/~wingated/www/introductions/mcmc-gibbs-intro.pdf>

Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 50(301), 236–244. doi:doi:10.2307/2282967

Weisberg, S. (2005). *Applied Linear Regression* (3rd ed.). New York: Wiley.

Wilks, S. S. (1938). Weighting systems for linear functions of correlated variables when there is no dependent variable. *Psychometrika*, 3(1), 23-40. doi:10.1007/BF02287917

Willey, G. R. (1953). Prehistoric Settlement Patterns in the Virú Valley, Peru. Bureau of American Ethnology.

Wood, S. (2006). Generalized Additive Models: An Introduction with R. London: Chapman & Hall/CRC.

Yarnold, P. R., & Soltysik, R. C. (2004). *Optimal Data Analysis*. American Psychological Association.

Zeger, S., Liang, K., & Albert, P. (1988). Models for Longitudinal Data: A Generalized Estimating Equation Approach. *Biometrics*, 44(4), 1049–1060. doi:10.2307/2531734

Zhang, e. a. (2013). Agglomerative clustering via maximum incremental path integral. In *Pattern Recognition*.

Zhang, e. a. (October 7–13, 2012). Graph degree linkage: Agglomerative clustering on a directed graph. *12th European Conference on Computer Vision*. Florence, Italy. From <http://arxiv.org/abs/1208.5092>



# Index

---

## A

|                                     |   |
|-------------------------------------|---|
| A. K. Erlang                        | 580   |
| <i>Acceptance-Rejection Methods</i> | 664   |
| activity on arc                     | 272, 275  |
| activity on node                    | 273   |
| adjusted R <sup>2</sup>             | 457   |
| Akaike information criterion        | 457   |
| analytics                           | 1, 70, 377, 380, 382  |
| Anderson-Darling test               | 363, 374, 428, 431, 435, 436, 567   |
| ANN                                 | 530   |
| annihilator matrix                  | 445, 452  |
| ANOVA                               | 503   |
| approximately uncorrelated          | 596, 597  |
| arrival                             | 361, 362, 662, 666, 667, 670, 671, 672, 675, 681                          |
| arrival process                     | 580, 581, 582, 584, 587, 675, 676, 677, 681                               |
| arrival rate                        | 583, 584, 588, 589, 590   |
| event                               | 670, 672, 673, 679  |
| interarrival time                   | 373, 565, 582, 583, 584, 587, 588, 589, 603, 658, 663, 664, 667, 672, 673 |
| schedule arrival                    | 667, 672, <i>See</i> arrival  |
| attributes                          | 384, 518  |
| augmented matrix                    | 30, 31, 44  |
| autocorrelation                     | <i>See</i> correlation  |
| average                             | 332, 333, 343, 344, 355, 361, 673, 681                                    |

## B

|                              |   |
|------------------------------|---|
| backward                     | 386   |
| balking                      | 582   |
| basic feasible solution      | 160, 161, 162, 163, 164, 165, 166, 167, 168, 170, 171, 173, 175, 176, 178, 179, 180, 181, 182, 185, 186, 192, 193, 194, 196, 197, 207, 209, 210 |
| basic variable               | 40, 41, 159, 163, 168, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 182, 184, 189, 192, 193, 236, 243, 244, 245, 260                  |
| Batched Means                | 594, 595, 596, 597, 602, 626  |
| Bayes' rule                  | 338, 339  |
| Bayesian logistic regression | 532   |

|                                |  |
|--------------------------------|--|
| Bernoulli variable             | 486, 506, 531  |
| best linear unbiased estimator | 449  |
| Big M method                   | 192, 195, 199, 200, 201                                    |
| binary classification model    | 538  |
| binary integer program         | 291, 292   |
| binary predictor               | 505  |
| binary variable                | 491, 493, 499, 505, 506, 542, 547, 551                     |
| binding constraints            | 88, 94, 97, 129, 180, 212, 241, 259, 265, 305              |
| binomial data                  | 494  |
| bivariate data                 | 363, 367   |
| branch-and-bound               | 10, 293, 294, 298, 299                                     |
| branching                      | 293, 295, 296, 297, 298                                    |
| <b>C</b>                       |  |
| canonical form                 | 170, 171, 175, 176, 178, 179, 180, 181, 182, 195, 488, 489 |
| canonical link function        | 489, 490, 492  |
| capital budgeting model        | 107  |
| CART                           | 383  |
| Central Limit Theorem          | 345, 355, 359, 360, 374                                    |
| Certainty Assumption           | 81   |
| chaining phenomenon            | 393  |
| Chi-square estimator           | 511  |
| Chi-square test                | 71, 363, 374, 428, 429, 430, 431, 437                      |
| Chi-square value               | 512  |
| Cholesky factorization         | 59   |
| cluster analysis               | 380, 387, 388  |
| clustering                     | 380, 381, 382, 387, 388, 389, 390, 392, 393, 402, 406, 407 |
| biclustering                   | 389  |
| connectivity based             | 390, 391   |
| hard clustering                | 389  |
| hierarchical clustering        | 388, 389, 390, 391   |
| overlapping                    | 389  |
| soft clustering                | 389  |
| strict partitioning            | 389  |
| subspace                       | 389  |
| clustering algorithms          | 381, 388, 390  |
| collectively exhaustive        | 338  |
| combinatorial curse            | 300  |
| conditional probability        | 335, 336   |
| confidence interval            | 575, 592, 595, 596, 597, 602, 616, 631                     |
| confidence level               | 454  |

|                                  |   |
|----------------------------------|---|
| <b>constrained least squares</b> | 452   |
| constraints                      | 10, 12, 66, 67, 68, 69, 70, 73, 76, 77, 78, 79, 81, 82, 88, 90, 91, 94, 95, 98, 102, 104, 105, 106, 112, 117, 122, 126, 129, 137, 138, 139, 140, 143, 144, 145, 146, 147, 148, 149, 150, 153, 154, 156, 157, 158, 159, 160, 163, 165, 167, 170, 180, 182, 184, 187, 188, 192, 195, 196, 204, 214, 220, 221, 222, 223, 230, 231, 236, 237, 239, 241, 242, 243, 244, 248, 254, 256, 257, 258, 259, 260, 264, 265, 285, 292, 295, 301, 305, 310, 311, 313, 317, 318, 319, 326, 519 |
| continuous variable              | 507, 528  |
| convergence                      | 510, 511, 532   |
| convex set                       | 88, 89, 129, 160  |
| correlation                      | 345, 457, 495, 510, 514, 538, 541, 568, 569, 570, 572   |
| autocorrelation                  | 574, 575, 576, 592, 626   |
| Correlated Output Data           | 575   |
| correlation tests                | 567, 574  |
| Cox and Snell R2                 | 514   |
| Cramér–Rao bound                 | 450   |
| Cramer-Von Mises test            | 435   |
| critical path                    | 271, 280, 282, 283, 287, 288, 289   |
| critical path method             | 271, 275, 285, 286, 287, 288, 289   |
| cross-sell                       | 385   |
| cumulative distribution function | 493, 524, 534   |
| curve fitting                    | 363, 367  |
| customer attrition               | 384   |
| Customer Lifetime Value          | 377   |
| <b>D</b>                         |   |
| data mining                      | 385   |
| decision variables               | 65, 66, 67, 68, 69, 70, 73, 74, 76, 77, 78, 79, 80, 83, 88, 91, 95, 102, 120, 121, 129, 130, 169, 211, 221, 243, 256, 257, 259, 260, 263, 291, 292, 293, 294, 298, 299, 301, 302  |
| degenerate                       | 163   |
| degrees of freedom               | 374, 428, 430, 433, 445, 468, 513, 516, 532, 543, 544, 548, 549, 596  |
| dendrogram                       | 390, 398, 399, 400, 401, 402  |
| dependent variable               | 441, 445, 452, 455, 456, 460, 497, 505, 506, 507, 508, 511, 515, 518, 525, 538  |
| <i>design matrix</i>             | 440, 498, 502   |
| determinant                      | 13, 52, 58  |
| deterministic model              | 67  |
| deviance                         | 499, 500, 501, 503, 504, 512, 513, 514, 516, 542, 543, 544, 548, 549  |

|  |   |
|--|---|
| diagonal elements                                | 42, 50, 58  |
| discrete event simulation                        | 67, 564, 565, 567, 576, 610, 611, 613, 661, 662, 682  |
| <i>Discrete Event Simulation Using ExtendSim</i> | 661, 689  |
| Distributed Interactive Simulation               | 565   |
| Distribution                                     | 574   |
| Bernoulli  | 486, 490, 491, 493, 519, 520  |
| binomial distribution                            | 486, 491, 531   |
| Chi-square                                       | 450, 501, 512, 513, 516, 517  |
| cumulative distribution function                 | 341, 342, 350, 354, 493, 664  |
| density function                                 | 343   |
| empirical distributions                          | 415, 417  |
| Erlang distribution                              | 579, 589, 682   |
| exponential density function                     | 361   |
| exponential distribution                         | 352, 361, 373, 487, 490, 579, 583, 584, 588, 618, 664   |
| frequency distribution                           | 568   |
| gamma distribution                               | 361, 422, 423, 424, 429, 487, 490, 534  |
| Gaussian distribution                            | 344, 415, 418, 532  |
| multinomial distribution                         | 491   |
| Normal distribution                              | 344, 345, 348, 350, 351, 355, 356, 357, 360, 373, 414, 435, 443, 457, 485, 493, 532, 533, 534, 535, 574 |
| Poisson distribution                             | 362, 414, 418, 426, 429, 486, 495, 496, 584   |
| probability density function                     | 341, 342  |
| probability distribution                         | 341, 349, 358, 584  |
| sampling distribution                            | 346, 347, 348, 356, 357, 358, 359, 360, 374   |
| Standard Normal distribution                     | 348, 357, 454, 524  |
| stationary                                       | 594   |
| Uniform Density Function                         | 566   |
| uniform distribution                             | 567, 568  |
| Weibull distribution                             | 361, 364, 414, 435  |
| Divisibility Assumption                          | 80, 106, 114, 291   |
| dual problem                                     | 12, 248, 249, 250, 251, 252, 253, 255, 256, 258, 260  |
| dual variable                                    | 231, 236, 259, 261, 283   |
| dummy variables                                  | 519   |
| Durbin–Watson statistic                          | 457   |
| dynamic model                                    | 65, 66  |
| dynamic programing models                        | 67  |
| <b>E</b>   |   |
| early event time                                 | 275   |

|                                    |   |
|------------------------------------|---|
| elementary row operation           | 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 53, 54, 174, 175, 177   |
| empirical distribution function    | 414, 431, 432   |
| entering variable                  | 172, 173, 174, 176, 178, 180, 182, 195, 208   |
| ERO                                | <i>See</i> elementary row operation   |
| error sum of squares               | 444   |
| Euclidean distance                 | 391, 392  |
| Event                              | 272, 278, 282, 284, 285, 334, 335, 336, 337, 342, 361, 515, 661, 662, 663, 667, 668, 670, 671, 672, 673, 674  |
| discrete-event                     | 674   |
| event graph                        | 674   |
| event list                         | 662, 663, 665, 667, 668, 670  |
| event notice                       | 662, 663, 665, 667, 668, 669  |
| event routine                      | 662, 663, 665, 666, 667, 668, 669, 670, 672, 675  |
| event schedule                     | 661, 663, 665, 667, 673, 682  |
| event time                         | 663   |
| Event schedules                    | <i>See</i> event schedule   |
| Excel                              | 7, 10, 11, 24, 110, 153, 220, 230, 231, 232, 233, 239, 241, 247, 264, 288, 290, 313, 315, 317, 325, 328, 369, 567, 572, 590, 591, 597, 625, 626, 628, 629, 630, 631, 632, 658           |
| Excel Solver                       | 230, 232, 322, 325, 328   |
| exogeneity                         | 441, 443, 448   |
| Expectation-maximization algorithm | 388   |
| expected value                     | 332, 341, 342, 346, 356, 485, 487, 489, 491, 496, 520, 521  |
| experiment                         | 333, 334, 335, 338, 363, 365, 366, 441, 564, 565, 575, 607, 620   |
| explanatory variables              | 458, 459, 497, 498, 507, 509, 518, 519, 520, 521, 524, 529  |
| exponential fitting                | 479   |
| <i>ExtendSim</i>                   | 363, 567, 675   |
| extreme point                      | 88, 89, 90, 160, 161, 162, 163, 227   |
| <b>F</b>                           |   |
| feasible region                    | 70, 81, 82, 85, 86, 87, 88, 89, 90, 92, 93, 95, 97, 98, 99, 100, 101, 102, 129, 130, 141, 160, 161, 162, 163, 164, 165, 168, 183, 186, 189, 191, 202, 222, 223, 226, 227, 258, 295, 613 |
| feasible solution                  | 85, 95, 160, 161, 168, 176, 178, 221, 222, 223, 254   |
| finite set                         | 89, 299   |
| finite source models               | 581   |
| fitted values                      | 444, 459  |
| fitting distributions              | 413, 417, 436   |
| fixed charge                       | 310, 313, 322   |

|  |   |
|--|---|
| forecasting  | 515   |
| free float   | 281   |
| frequency density                                    | 414, 437  |
| <i>F</i> -statistic                                  | 458, 468  |
| <i>F</i> -test                                       | 450, 513  |
| <b>G</b>   |   |
| G/G/1 Queue  | 588   |
| Gauss–Jordan method                                  | 13, 28, 31, 32, 34, 36, 37, 38, 40, 41, 42, 44, 174                       |
| Gauss–Jordon form                                    | 53  |
| Gauss–Markov theorem                                 | 448, 449, 450   |
| general linear model                                 | 485, 492, 498, 499  |
| <b>generalized additive model</b>                    | 496, 497  |
| generalized additive model location, scale and shape | 497   |
| <b>Generalized estimating equations</b>              | 495   |
| generalized least squares                            | 443   |
| <b>Generalized linear mixed models</b>               | 496   |
| generalized linear model                             | 67, 485, 486, 487, 491, 492, 493, 495, 497, 498, 521, 524                 |
| George E.P. Box                                      | 64  |
| GI/G/1 Queue   | 595   |
| Gibbs sampling                                       | 492, 493, 532, 533, 535   |
| Gleason model  | 463   |
| goal programming                                     | 202, 212  |
| goodness-of-fit                                      | 374, 413, 415, 425, 426, 427, 428, 436, 438, 445, 457, 511, 514, 567, 568 |
| goodness-of-fit tests                                | 413, 427  |
| graphical solution                                   | 86, 93, 97, 129, 164, 221   |
| <b>H</b>   |   |
| Hessian matrix                                       | 492   |
| heteroscedasticity                                   | 442, 456, 459, 467  |
| heuristic models                                     | 64  |
| hierarchical   | 1, 390, 391, 393  |
| hierarchical clustering                              | 391, 393  |
| High Level Architecture                              | 565   |
| histograms   | 414, 417, 427, 608  |
| holistic   | 1, 2, 3   |
| homoscedastic  | 439, 449  |
| <b>Hosmer–Lemeshow test</b>                          | 515   |
| hyperplane   | 444   |

|                                 |   |
|---------------------------------|---|
| <b>Hypothesis Test</b>          |   |
| Alternate Hypothesis            | 373   |
| Critical Region                 | 373   |
| level of significance           | 374   |
| Null Hypothesis                 | 373   |
| Statistical hypotheses          | 373   |
| Test Statistic                  | 373, 374  |
| <b>hypothesis testing</b>       | 346, 373, 427, 445  |
| <br><b>I</b>                    |   |
| identity matrix                 | 42, 50, 58  |
| IID                             | 589, 663, 673   |
| independence                    | 53, 79, 80, 159, 163, 286, 288, 289, 335, 340, 345, 348, 356, 357, 359, 360, 361, 385, 441, 442, 443, 450, 487, 489, 497, 504, 505, 506, 514, 515, 518, 519, 531, 575, 584, 585, 587, 588, 590, 592, 594, 596, 597, 602, 613, 615, 617, 663, 676, 677 |
| independent variables           | 441, 515, <i>See explanatory variables</i>  |
| infinite number of solutions    | 28, 36, 38, 41, 82  |
| initial transient period        | 592, 594, 597, 674  |
| inner product                   | 49, 50  |
| input variables                 | <i>See explanatory variables</i>  |
| integer model                   | 66  |
| integer programming             | 81, 107, 322  |
| interarrival time               | <i>See arrival</i>  |
| inverse of a matrix             | 43, 44, 51, 52  |
| <i>Inverse Transform Method</i> | 664   |
| inverse transformation          | 618   |
| involution                      | 249   |
| iso-profit line method          | 223, 229  |
| <br><b>J</b>                    |   |
| joint probabilities             | 338   |
| <br><b>K</b>                    |   |
| k-means                         | 388, 390, 406, 407  |
| Kolmogorov-Smirnov              | 363, 374, 567   |
| Kolmogorov-Smirnov test         | 428, 431, 432, 434, 435, 436, 438, 574  |
| kurtosis                        | 413, 417, 418, 420, 432, 433, 437, 498  |
| <br><b>L</b>                    |   |
| late event time                 | 277   |

|                              |   |
|------------------------------|---|
| latent variable              | 523, 525, 528, 530, 532, 533, 534   |
| Law of Multiplication        | <i>See Multiplicative Rule</i>  |
| Law of Total Probability     | 337   |
| least squares                | 54, 55, 364, 367, 372, 442, 446, 452, 453, 454, 456, 472, 473, 485, 511, 522  |
| least squares estimators     | 453   |
| least squares method         | 439, 485  |
| least-squared error          | 52  |
| Levenshtein distance         | 392   |
| <b>leverage points</b>       | 451   |
| likelihood                   | 338   |
| likelihood function          | 423, 424, 510, 532  |
| likelihood ratio             | 512, 513, 514, 516, 614, 617  |
| likelihood ratio R2          | 514   |
| Likelihood Ratio method      | 619   |
| likelihood-ratio test        | 512, 516  |
| Lilliefors test              | 435, 437  |
| line segment                 | 14, 15, 18, 19, 86, 88, 89, 90, 97, 98, 101, 130  |
| linear algebra               | 13, 47  |
| linear combination           | 447, 451, 485, 487, 489, 507, 508, 520  |
| linear discriminant analysis | 505   |
| linear equation              | 25, 29, 31, 42, 79, 80  |
| linear equations             | 13, 25, 26, 28, 47, 51, 61, 159, 170, 446, 452  |
| linear function              | 78, 79, 129, 137, 260, 261, 262, 440, 507   |
| linear inequalities          | 79  |
| linear model                 | 52, 66, 440, 456, 462, 463, 485, 486, 492, 504, 511, 521  |
| linear predictor             | 487, 489, 491, 496, 520, 521, 523, 528  |
| linear programming           | 11, 25, 28, 41, 73, 74, 75, 77, 78, 79, 80, 81, 82, 83, 87, 88, 89, 90, 91, 92, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 106, 107, 111, 112, 113, 114, 115, 118, 119, 120, 122, 124, 125, 127, 129, 130, 131, 132, 134, 135, 137, 140, 143, 147, 153, 154, 155, 156, 157, 158, 160, 161, 163, 164, 165, 167, 168, 169, 170, 180, 181, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 195, 199, 200, 202, 203, 205, 206, 215, 220, 221, 227, 229, 230, 231, 233, 235, 236, 237, 238, 239, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 255, 256, 258, 259, 260, 261, 263, 266, 267, 268, 269, 282, 283, 284, 285, 291, 292, 293, 294, 295, 298, 299, 301, 305, 308, 313, 315, 317 |
| linear programming models    | 67, 73  |
| linear programming problem   | <i>See linear programming</i>   |
| linear regression model      | 67, 439, 440, 453   |
| linear system                | 25, 26, 27, 29, 30, 31, 32, 36, 37, 38, 40, 41, 42, 44, 159, 160  |

|                              |   |
|------------------------------|---|
| liner model                  | 64  |
| link function                | 485, 486, 487, 489, 492, 493, 496, 507, 508   |
| linkage                      | 391, 392, 393   |
| linkage criteria             | 392, 393  |
| <i>linpro</i>                | 137, 141, 142, 143, 149, 215, 217, 218, 241, 242, 243, 245, 246, 252, 254, 284, 303, 320                                    |
| log-Arrhenius model          | 463, 465, 467   |
| logistic distribution        | 523, 524, 526, 532, 533, 534, 535   |
| logistic function            | 505, 507, 508, 509, 522, 524, 534   |
| logistic regression          | 485, 491, 493, 500, 505, 506, 507, 511, 512, 514, 515, 516, 518, 519, 520, 521, 523, 524, 525, 530, 532, 533, 538, 542, 547 |
| logit                        | 487, 500, 503, 507, 508, 511, 515, 522, 524, 525, 528, 529, 538, 542, 543, 548, 549   |
| logit function               | 493, 508  |
| logit link                   | 493   |
| logit models                 | 493, 494, 524   |
| logit regression             | <i>See logistic regression</i>  |
| log-likelihood               | 456, 457  |
| log-log function             | 493   |
| Long Term Capital Management | 386   |
| lower triangular matrix      | 59  |
| LPSolve                      | 10, 11, 304, 305, 307, 313, 314, 315, 328, 688  |
| <i>LU</i> decomposition      | 55, 59  |

## M

|                        |   |
|------------------------|---|
| M/M/1 Queue            | 588, 591, 592, 594, 595, 603, 658, 663  |
| machine learning       | 530, 532  |
| management science     | 1   |
| marginal distribution  | 534   |
| marginal prices        | 236   |
| marginal probabilities | 338   |
| Markov chain           | 492, 532, 533   |
| Markovian property     | <i>See memoryless property</i>  |
| MASS package           | 425   |
| mathematical models    | 64, 579   |
| MATLAB                 | 10, 305, 688  |
| matrix                 | 13, 14, 17, 19, 20, 21, 23, 27, 28, 30, 31, 32, 36, 37, 38, 39, 40, 42, 43, 44, 45, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 137, 138, 148, 150, 151, 237, 252, 253, 301, 302, 303, 392, 406, 440, 442, 443, 444, 445, 446, 448, 449, 451, 452, 453, 454, 468, 472, 473, 474, 481, 489, 492, 501, 541, 621, 626, 675 |

|                                     |   |
|-------------------------------------|---|
| matrix form                         | 30, 37, 51, 52, 253, 444, 468   |
| matrix multiplication               | 19, 20, 21, 22, 23, 24, 25, 50, 468   |
| matrix operations                   | 49  |
| matrix representation               | 27, 31  |
| maximization problem                | 82, 184, 190, 248, 250, 252, 258, 259, 260, 261   |
| maximum a posteriori                | 522, 532  |
| maximum likelihood                  | 421, 423, 437, 439, 447, 485, 487, 491, 510, 522  |
| maximum likelihood estimates        | 491   |
| maximum likelihood estimation       | 423, 424, 485, 510, 522   |
| maximum likelihood estimator        | 439, 443, 445, 447, 448, 500  |
| mean                                | 322, 333, 344, 345, 346, 347, 348, 349, 350, 351, 356, 357, 358, 360, 361, 362, 373, 374, 413, 422, 566, 568, 574, 576, 577, 673, 682 |
| sample mean                         | 333, 346, 347, 356, 357, 358, 360   |
| memoryless property                 | 583   |
| memoryless property                 | 583, 584  |
| MILP                                | <i>Mixed Integer Linear Programming</i>   |
| minimization problem                | 82, 100, 190, 248, 250, 252, 254, 257, 258, 259, 260, 262   |
| <b>Mixed Integer Linear Program</b> | 292   |
| Mixed Integer Linear Programming    | 10, 11  |
| model                               | 362, 577, 661, 667, 669, 683  |
| model sum of squared                | 458   |
| modeling                            | 5, 6, 7, 8, 63, 71, 286, 301, 369, 383, 384, 385, 497, 520, 523, 587, 607, 608, 609, 610, 611, 612, 661, 675, 676, 677, 689           |
| Monte Carlo                         | 290, 492, 532, 533, 563, 566, 613, 616, 620, 622, 627   |
| Monty Hall problem                  | 623   |
| multicollinearity                   | 439, 443, 510   |
| multinomial logit models            | 495, 525, 530   |
| multiperiod cash management         | 120   |
| multiple linear fitting             | 480   |
| Multiplication Rule                 | 336   |
| multiplicative inverse              | 42, 43  |
| mutually exclusive events           | 338   |
| <b>N</b>                            |   |
| Nagelkerke R <sup>2</sup>           | 514   |
| natural logarithm                   | 507, 508, 510, 512, 618   |
| <b>NbClust package</b>              | 407   |
| negative log-likelihood             | 424   |
| net present value                   | 108, 109, 110, 111, 112, 113, 114, 115, 326, 329  |
| network models                      | 67, 271   |
| Newton–Raphson method               | 491   |

|                                  |  |
|----------------------------------|--|
| Newton's method                  | 510  |
| no solution                      | 28, 36, 37, 40, 42, 54, 160  |
| nomograms                        | 383  |
| nonbasic variable                | 40, 41, 159, 168, 171, 172, 176, 179, 180, 185, 189, 243, 245, 260   |
| nonbinding constraints           | 88, 265  |
| nonlinear model                  | 66   |
| non-parametric                   | 496, 497   |
| non-singular matrix              | 58   |
| Normal PDF                       | 349  |
| normal standardized distribution | 434  |
| normality condition              | 345  |
| normality tests                  | 427, 432   |
| normalized                       | 528  |
| nuisance parameter               | 442  |
| <i>null hypothesis</i>           | 373, 374, 375, 431, 432, 433, 457, 458   |
| null model                       | 512, 516, 544  |
| null space                       | 53, 58   |
| number of runs                   | 602, 603   |
| <b>O</b>                         |  |
| objective function               | 10, 66, 67, 68, 69, 73, 74, 75, 76, 78, 79, 80, 81, 82, 87, 90, 91, 94, 95, 102, 104, 105, 117, 121, 123, 125, 128, 129, 137, 138, 144, 146, 147, 150, 153, 167, 168, 169, 170, 184, 187, 190, 192, 193, 194, 195, 196, 200, 201, 202, 205, 206, 207, 214, 220, 223, 229, 231, 235, 236, 248, 250, 257, 258, 259, 260, 261, 262, 264, 282, 301, 310, 311, 318, 319 |
| Octave                           | 10, 305, 312, 688  |
| odds ratio                       | 507, 514, 516  |
| omnibus tests                    | 427  |
| optimal interpolation            | 618  |
| optimal solution                 | 65, 70, 81-89, 93-95, 98, 10-102, 106, 114, 119, 122, 127, 129, 130, 140, 145, 163, 164, 167, 179, 180-186, 191, 194, 196, 199, 200, 205, 206, 209, 211, 221-229, 230, 231, 237, 244, 247-250, 254, 258, 259, 260, 262, 268, 269, 283, 294, 295, 297, 311, 312, 317, 320, 322, 323, 328  |
| optimization                     | 5, 6, 7, 69, 73, 522, 605  |
| optimization models              | 66, 67, 68, 69, 70, 78   |
| <b>ordinary least squares</b>    | 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 451, 452, 453, 456, 458, 460  |
| orthonormal basis                | 58   |
| outcome                          | 333, 334, 338, 344, 377, 387, 487, 490, 491, 505, 506, 516, 518, 519, 520, 521, 522, 525   |

|                           |  |
|---------------------------|--|
| outcome variable          | <i>See dependent variables</i>   |
| outer product             | 48   |
| output variable           | <i>See dependent variables</i>   |
| <b>P</b>                  |  |
| parallelogram law         | 17, 19   |
| parameter                 | 332, 344, 346, 347, 348, 356, 363, 373, 570, 575, 665, 673, 682                                    |
| parameter estimates       | 421, 422, 423  |
| parametric                | 496, 497, 498  |
| Pareto Principle          | 377  |
| partitioning              | 390, 393, 511  |
| Pearson's Chi-square test | 436  |
| <b>PERT</b>               | <i>See Program Evaluation and Review Technique</i>   |
| Perturbation Analysis     | 618, 619   |
| physical models           | 64   |
| $p$ -norm of a matrix     | 58   |
| point estimates           | 383  |
| Poisson distribution      | <i>See Distribution</i>  |
| Poisson Process           | 361, 362, 675, 676   |
| Poisson random variable   | 583  |
| polynomial fitting        | 476  |
| population                | 331, 332, 333, 343, 344, 345, 346, 347, 355, 356, 357, 359, 360, 373, 374, 413, 575                |
| mean                      | 360  |
| variance                  | 360  |
| posterior                 | 532  |
| posterior distribution    | <i>See posterior</i>   |
| posterior probability     | 338, 492   |
| predecessors              | 271, 272, 274, 276, 277  |
| predictive modeling       | 377, 384, 385  |
| predictive models         | 382, 385   |
| predictor model           | 516  |
| predictor variables       | 365, 496, 505, 506, <i>See explanatory variables</i>   |
| predictors                | 440, 462, 485, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517                          |
| prescriptive model        | 67   |
| primal problem            | 11, 12, 248, 250, 251, 252, 256, 260, 261  |
| prior distribution        | 522, 533   |
| prior probability         | 338  |
| probability               | 334, 336, 337, 341, 342, 343, 344, 346, 347, 353, 357, 361, 362, 373, 374, 384, 566, 664, 681, 682 |

|   |   |
|---|---|
| continuous random variable              | 334   |
| density function                        | 341   |
| discrete random variable                | 334, 341  |
| distribution                            | 340   |
| expected value                          | 332, <i>See</i> probability   |
| random event                            | 359, 361, 362   |
| random variable                         | 333, 334, 341, 342, 344, 345, 357, 374, 575, 673                                    |
| random variate                          | 566, 663, 664   |
| probability density function            | 413   |
| probability distribution                | 340, 341, 348, 349, 357, 373, 487, 497, 520, 522, 528, 566, 579, 582, 583, 585, 682 |
| probability mass function               | 488, 520, 522   |
| probability of an outcome               | 377   |
| probability of success                  | 520   |
| probit models                           | 493, 494, 523, 524  |
| Program Evaluation and Review Technique | 271, 285, 286, 287, 288, 290  |
| project diagram                         | <i>See</i> project network  |
| project network                         | 271, 272, 273, 274, 275, 277, 279, 280, 282, 283, 285, 286, 289                     |
| propensity models                       | 377, 379, 506   |
| propensity to buy                       | 379   |
| propensity to churn                     | 379   |
| propensity to engage                    | 378   |
| propensity to unsubscribe               | 378   |
| Proportionality assumption              | 79  |
| Proportionality Assumption              | 80  |
| pseudo-R <sup>2</sup>                   | 514, 515  |
| pure integer program                    | 291   |
| p-value                                 | 430, 432, 433, 456  |
| <i>p</i> -values                        | 457, 458  |

## **Q**

|                         |  |
|-------------------------|--|
| <i>QR</i> decomposition | 59   |
| quadratic optimization  | 54   |
| Quantile-Quantile plot  | 415  |
| quapro                  | 137  |
| Queue                   | 362, 661, 662, 663, 665, 666, 669, 670, 671, 673, 674, 682 |
| balk                    | 682  |
| FCFS discipline         | 585  |
| LCFS discipline         | 586  |
| mean queue time         | 592  |
| MM1                     | 663, 669, 670, 671, 674                                    |

|   |  |
|---|--|
| priority discipline   | 586  |
| queue discipline  | 585  |
| queue length  | 673  |
| queue time  | 64, 564, 579, 586, 588, 589, 590, 592, 594, 603, 610   |
| queuing system  | 662  |
| queuing theory  | 362  |
| SIRO discipline   | 586  |
| SPT discipline  | 586  |
| queueing models   | 67, 580  |
| queueing system   | 580, 584, 585, 588, 589, 675   |
| queueing theory   | 579, 581   |
| <b>R</b>  |  |
| R 8, 9, 369, 370, 413, 418, 424, 426, 429, 433, 436, 460, 462, 499, 540, 541, 544, 551, 624 |  |
| R functions   | 370  |
| R library   | 370  |
| <i>R studio</i>   | 383  |
| <i>R Studio</i>   | 381, 685, 686  |
| R <sup>2</sup>  | 457, 514   |
| Radon-Nikodym theorem   | 614  |
| RAND()  | 590, 625, 626, 627, 630  |
| random number   | 567, 618, 625, 663, 675, 676   |
| random number generation  | 663  |
| Random Number Generator   | 664  |
| Random Number Generator   |  |
| Additive Congruence Generator   | 567, 568, 569  |
| Excel Random Number Generator   | 572  |
| ExtendSim   | 574  |
| Linear Congruential Generator   | 567, 570   |
| Random Function in Excel  | 571  |
| Runs tests  | 574  |
| <b>Random Number Generator</b>  | 676  |
| random numbers  | 566, 567, 568, 574, 575, 664, 665  |
| pseudo random   | 663  |
| pseudo-random   | 567  |
| random variable   | 286, 287, 288, 332, 333, 334, 340, 341, 342, 343, 348, 349, 352, 357, 358, 361, 362, 374, 375, 423, 428, 454, 485, 523, 575, 577, 586, 587, 676, 677 |
| random variables  | 440, 441, 531  |
| Random Variate Generation   | 664  |

|                           |   |
|---------------------------|---|
| rank of a matrix          | 53, 55, 58, 206, 245, 312, 442, 444, 452  |
| rare events               | 564   |
| ratio test                | 174, 177, 180, 181, 182, 184, 185, 188, 190, 197, 198, 200, 209, 210, 516   |
| reduced cost              | 179, 235, 243, 244, 245, 260  |
| regression coefficient    | 442, 508, 516, 517  |
| regression intercept      | 446   |
| regression models         | 67  |
| regressor                 | 439, 440, 441, 442, 443, 446, 447, 448, 450, 451, 452, 454, 456, 457  |
| reneging                  | 582   |
| residual sum of squares   | 444, 458  |
| residuals                 | 444, 447, 458, 459, 512, 548  |
| response variable         | 363, 367, 382, 440, 454, 463, 465, 467, 468, 485, 486, 490, 493, 494, 495, 496, 497, 499, 501, 505, 511, 518, 540, 542, 547   |
| retention                 | 384, 385  |
| reweighted least squares  | 491   |
| right-hand side           | 77, 88, 118, 129, 170, 171, 181, 182, 184, 185, 188, 192, 195, 196, 197, 199, 200, 209, 210, 214, 259, 260, 439, 500  |
| <b>S</b>                  |   |
| sample                    | 331, 332, 333, 343, 344, 345, 346, 347, 348, 355, 356, 357, 358, 360, 373, 374, 574, 575, 577   |
| standard deviation        | 360   |
| sample size               | 360   |
| Sample Size Determination | 577   |
| sample space              | 333, 334, 335, 337, 374   |
| Sampling distribution     | 345   |
| saturated model           | 512   |
| scalar                    | 15, 16, 17, 18, 20, 21, 27, 28, 47, 48, 60, 440, 446, 488   |
| scalar product            | 15, 16, 20, 27  |
| scale parameter           | 524   |
| scatterplot               | 375, 415, 456, 496  |
| scheduling problem        | 268, 271, 321   |
| Schwarz criterion         | 456, 457  |
| SCILAB                    | 7, 8, 10, 47, 48, 49, 55, 56, 61, 137, 150, 151, 153, 214, 216, 241, 246, 252, 254, 259, 260, 263, 283, 299, 302, 307, 312, 323, 328, 329, 349, 352, 353, 354, 366, 367, 372, 375, 473, 681, 682, 687 |
| SCILIB                    | 632   |
| Score Function            | 616   |
| scoring                   | 492   |
| sensitivity analysis      | 12, 220, 230, 231, 238, 243, 258  |

|                           |   |
|---------------------------|---|
| Sensitivity Report        | 230, 240  |
| Service                   | 576, 662, 666, 667, 675, 682  |
| begin service             | 669, 670  |
| <b>begin service</b>      | 667   |
| end service               | 666, 667, 670   |
| service time              | 663, 666  |
| service times             | 669   |
| service process           | 585   |
| service times             | 585, 586, 587, 588, 589, 603, 658, 666, 667, 675, 682                             |
| set-covering problem      | 321   |
| shadow price              | 218, 231, 235, 236, 237, 239, 241, 243, 254, 259, 260                             |
| <b>Shapiro-Wilk test</b>  | 433   |
| share of wallet           | 378   |
| sign restriction          | 78, 79, 82, 155, 156, 172, 193, 195   |
| SIMAN                     | 607, 608  |
| simplex algorithm         | 28, 73, 153, 159, 168, 170, 171, 176, 178, 180, 181, 182, 183, 185, 186, 192, 207 |
| simplex method            | 10, 192, 201, 231   |
| SIMSCRIPT II.5            | 605, 606  |
| simulation                | 6, 7, 563, 597, 598, 606, 612, 620, 622, 627, 632, 661, 664, 672, 675, 689        |
| Simulation                |   |
| data                      | 575   |
| discrete event simulation | 567, 576, 662   |
| experiments               | 566   |
| generic language          | 664   |
| languages                 | 664   |
| <b>MM1 Queue</b>          | 671   |
| Monte Carlo               | 566   |
| output                    | 575   |
| purpose                   | 673   |
| simulation run            | 575   |
| step-by-step              | 665, 671  |
| stopping                  | 672   |
| simulation architectures  | 565   |
| simulation model          | 563, 609, 616, 661  |
| simulation models         | 67, 564   |
| singular matrix           | 52  |
| skewness                  | 413, 414, 417, 418, 420, 432, 433, 437, 498, 535                                  |
| smoothing                 | 496   |
| software                  | 383   |

|                                  |   |
|----------------------------------|---|
| SOS                              | <i>Special Ordered Sets</i>   |
| sparseness                       | 510   |
| Special Ordered Sets             | 10, 11  |
| splitting                        | 547   |
| square matrix                    | 42, 50, 55, 58, 59  |
| SSR                              | <i>See sum of squared residuals</i>   |
| standard deviation               | 413   |
| standard deviation               | 287, 288, 344, 345, 346, 348, 349, 350, 351, 352, 356, 357, 360, 374, 375, 376, 414, 435, 437, 631, 632 |
| standard error                   | 445, 449, 458, 468, 472, 500, 517   |
| standard error of the mean       | 360   |
| standard form                    | 153, 155, 156, 157, 158, 160, 161, 163, 165, 168, 170, 181, 193, 195, 248, 250, 251, 252, 299, 489      |
| <b>State</b>                     | 357, 661  |
| <b>empty-idle</b>                | 667   |
| instantaneous change             | 674   |
| state changes                    | 661, 662, 670, 674  |
| state variable                   | 661   |
| steady state                     | 576, 661, 673, 674  |
| variables                        | 662, 668, 675   |
| static model                     | 65  |
| stationary                       | 443, 584, 587, 588, 592, 594  |
| Statistic                        | 332, 333, 343, 345, 346, 347, 355, 373, 374, 413, 428, 435, 574, 575                                    |
| confidence interval              | 346, 575  |
| estimate                         | 347, 363, 575, 576, 577, 673, 674   |
| estimator                        | 333, 575  |
| sample statistics                | 359, 360  |
| statistical inference            | 344, 346, 356, 459  |
| statistical model                | 63, 386, 492, 497, 563, 663   |
| statistical tests                | 8, 346, 413, 574  |
| Steady State Simulation          |   |
| initial transient                | 674   |
| number of runs                   | 574   |
| steady-state                     | 386, 594, 595   |
| stepwise                         | 516   |
| stochastic model                 | 67  |
| strong duality theorem           | 250   |
| Student's <i>t</i> -distribution | 532, 534, 535   |
| sum of squared residuals         | 444, 446, 450, 452  |
| sum of squares                   | 445, 458, 511   |
| survival trees                   | 383   |

|                               |   |
|-------------------------------|---|
| symbolic models               | 64  |
| System                        | 575, 661, 662, 666, 667, 673, 674, 675  |
| dynamic system                | 662, 674  |
| queueing system               | 662, 663  |
| state                         | 663   |
| state                         | 661, 662  |
| System clock                  | 663, 665  |
| system state                  | <i>See System</i>   |
| systems dynamics              | 610, 611  |
| <b>T</b>                      |   |
| technological coefficients    | 77  |
| time series model             | 448   |
| timing routine                | 667   |
| total float                   | 279, 280, 281   |
| Transition Matrix             | 627, 628  |
| transpose of a matrix         | 19, 47, 50, 52, 440, 444, 468   |
| traveling salesman problem    | 299, 300, 302   |
| <i>t</i> -statistic           | 457   |
| <i>t</i> -test                | 516   |
| Type-II error                 | 517   |
| <b>U</b>                      |   |
| unbalanced data               | 517   |
| unbiased estimators           | 439, 449, 450   |
| unbounded                     | 95, 100, 101, 102, 130, 148, 165, 186, 189, 190, 191, 192   |
| unique solution               | 28, 30, 31, 35, 40, 41, 42, 44, 130, 147, 162   |
| univariate data               | 365   |
| unrestricted in sign          | 78, 79, 129, 153, 261, 282, 285   |
| Uplift Modeling               | 384   |
| upper triangular matrix       | 59  |
| <b>V</b>                      |   |
| validation                    | 538, 609, 613   |
| variables                     | 344, 661, 665   |
| variance                      | 332, 333, 341, 342, 343, 345, 346, 356, 358, 361, 362, 373, 439, 442, 445, 448, 449, 450, 485, 487, 488, 492, 494, 495, 500, 511, 512, 514, 515, 532, 534, 566, 568, 576, 577 |
| sample variance               | 333, 360  |
| variance reduction techniques | 615   |

|                           |   |
|---------------------------|---|
| vector                    | 14, 15, 16, 18, 19, 21, 25, 26, 27, 47, 48, 49, 50, 51, 55, 57, 58, 59, 137,<br>138, 145, 146, 162, 163, 235, 237, 241, 253, 302, 303, 370, 413, 417, 423,<br>425, 426, 429, 436, 440, 445, 446, 447, 448, 451, 452, 454, 468, 473, 474,<br>477, 479, 490, 491, 497, 498, 521, 529, 613, 614, 621 |
| verification              | 609   |
| V-linkage                 | 393   |
| <b>W</b>                  |   |
| waiting line              | <i>See Queue</i>  |
| waiting times             | <i>See queue times</i>  |
| Wald statistic            | 510, 516, 517   |
| Ward's criterion          | 393   |
| warm-up period            | <i>See initial transient period</i>   |
| What-if analysis          | 612, 614  |
| Wilkinson-Rogers notation | 462   |
| Wilks' Theorem            | 501, 513  |
| <b>X</b>                  |   |
| Xcos                      | 8, 632, 635, 636, 637, 638, 639, 640, 644, 650, 657, 675, 681, 682  |

