

Contents

| | |
|--|-------------|
| About This Book | ix |
| About The Author | xiii |
| Acknowledgments | xv |
| Part 1: Preparation | 1 |
| Chapter 1: SAS Language Preparation | 3 |
| 1.1 Keep It Simple | 3 |
| 1.1.1 Programming Challenge #1 | 5 |
| 1.1.2 Solution | 5 |
| 1.2 Keep Up with the Software | 6 |
| 1.3 Experiment..... | 10 |
| Chapter 2: Shifting Gears: Macro Language | 13 |
| Part 2: Technique | 21 |
| Chapter 3: CALL EXECUTE | 23 |
| 3.1 Basic Rules | 23 |
| 3.2 Achieving the Impossible | 24 |
| 3.3 Multiple CALL EXECUTEs | 26 |
| 3.4 Finally, the Intricacies | 27 |
| 3.4.1 Programming Challenge #2 | 29 |
| 3.4.2 Solution | 30 |
| 3.5 Execute an Experiment..... | 30 |
| 3.6 The Final Intricacy: Macro Variable Resolution | 32 |
| Chapter 4: %SYSFUNC..... | 35 |
| 4.1 Basic Examples | 35 |

| | |
|---|-----------|
| 4.2 Capturing the Program Name | 37 |
| 4.3 Commas and Nested %SYSFUNCS | 38 |
| 4.4 Achieving the Impossible, Revisited | 39 |
| 4.5 Capturing Option Settings..... | 40 |
| 4.6 Efficiency Considerations | 41 |
| 4.7 A Final Example: ZIP Codes..... | 42 |
| 4.7.1 Programming Challenge #3 | 43 |
| 4.7.2 Solution..... | 43 |
| Chapter 5: CALL SYMPUT | 45 |
| 5.1 Leading and Trailing Blanks..... | 46 |
| 5.2 A Similar Lesson, Using Recursion | 47 |
| 5.3 Test Your Skill..... | 47 |
| 5.3.1 Programming Challenge #4a | 47 |
| 5.3.2 Solution..... | 48 |
| 5.3.3 Programming Challenge #4b..... | 48 |
| 5.3.4 Solution..... | 49 |
| 5.3.5 Programming Challenge #4c..... | 49 |
| 5.3.6 Solution..... | 50 |
| 5.4 Function Shifts in the Real World | 51 |
| 5.5 A Key Issue: Extra Blanks | 53 |
| Chapter 6: INTO..... | 55 |
| 6.1 SELECT Executes Immediately | 55 |
| 6.2 Numeric-to-Character Conversion..... | 56 |
| 6.3 SELECTing Multiple Values..... | 56 |
| 6.4 DISTINCT Differences..... | 58 |
| 6.5 How Many Macro Variables? | 58 |
| 6.6 Zero Incoming Observations | 59 |
| 6.6.1 Programming Challenge #5..... | 60 |
| 6.6.2 Solution..... | 60 |
| 6.7 An Unusual Application: Separated by Else | 61 |
| 6.8 Dictionary Tables | 62 |
| 6.9 Extremely Long Lists | 64 |
| 6.10 Blanks vs. Nulls | 66 |

| | |
|--|------------|
| Chapter 7: Macro Quoting..... | 67 |
| 7.1 Why Quoting is Necessary | 67 |
| 7.2 Why Quoting is a Nightmare | 69 |
| 7.3 What Quoting Really Does | 70 |
| 7.4 Peeking Inside the Black Box of Quoting | 72 |
| 7.5 The Final Word on Quoting | 75 |
| Chapter 8: %LOCAL vs. %GLOBAL..... | 77 |
| 8.1 Why Think %Locally?..... | 77 |
| 8.2 Creating Symbol Tables and Macro Variables | 78 |
| 8.2.1 Programming Challenge #6 | 80 |
| 8.2.2 Solution | 81 |
| 8.3 Symbol Tables with CALL SYMPUT | 81 |
| 8.4 Symbol Tables with CALL SYMPUTX..... | 83 |
| 8.5 Choosing the Source Table..... | 84 |
| 8.6 A Persisting Impact..... | 85 |
| Chapter 9: Arithmetic | 87 |
| 9.1 Basic Rules | 87 |
| 9.2 Truth in Numbers, Expressions, and Comparisons | 89 |
| 9.2.1 Programming Challenge #7 | 92 |
| 9.2.2 Solution | 93 |
| Part 3: Execution..... | 95 |
| Chapter 10: Generating Text..... | 97 |
| 10.1 Utilizing Generated Text | 97 |
| 10.2 Counting Words in a String | 99 |
| 10.3 Working with Lists..... | 101 |
| 10.4 Prefer the Macro Solution | 103 |
| Chapter 11: Debugging | 107 |
| 11.1 Store the Generated SAS Code | 108 |
| 11.2 Control Diagnostic Messages..... | 109 |
| 11.3 End When the Error Occurs | 110 |

| | |
|--|------------|
| Chapter 12: Complexity vs. Simplicity | 115 |
| 12.1 Building Job Security | 115 |
| 12.1.1 Programming Challenge #8 | 118 |
| 12.1.2 Solution | 118 |
| 12.2 Ease of Use | 119 |
| 12.3 Checking Parameters | 121 |
| 12.4 Portability | 123 |
| 12.5 Complexity vs. Speed | 124 |
| 12.6 Miscellaneous Applications | 126 |
| Chapter 13: A Final Illusion: Backtesting | 129 |
| 13.1 The Preparation | 129 |
| 13.2 The Steps | 131 |
| 13.3 The Implementation | 135 |
| 13.3.1 Programming Challenge | 137 |
| Index | 139 |