Statistics and Computing

Robert A. Muenchen · Joseph M. Hilbe

R for Stata Users

Springer

Statistics and Computing

Series Editors
J. Chambers
D. Hand
W. Härdle

For other titles published in this series, go to
http://www.springer.com/series/3022

Robert A. Muenchen · Joseph M. Hilbe

R for Stata Users

Springer

Robert A. Muenchen
University of Tennessee
Office of Information Technology
Statistical Consulting Center
916 Volunteer Blvd.
Knoxville TN 37996-0520
Stokeley Management Center
USA
muenchen.bob@gmail.com

Joseph M. Hilbe
7242 W. Heritage Way
Florence Arizona 85132
USA
hilbe@asu.edu

ISBN 978-1-4419-1317-3 e-ISBN 978-1-4419-1318-0
DOI 10.1007/978-1-4419-1318-0
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010921041

© Springer Science+Business Media, LLC 2010
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

While R and Stata have many features in common, their languages are quite different. Our goal in writing this book is to help you translate what you know about Stata into a working knowledge of R as quickly and easily as possible. We point out how they differ using terminology with which you are familiar and we include many Stata terms in the table of contents and index. You can find any R function by looking up its counterpart in Stata and vice versa. We provide many example programs done in R and Stata so that you can see how they compare topic by topic.

When finished, you should be able to use R to:

- Read data from various types of text files and Stata data sets.
- Manage your data through transformations, recodes, and combining data sets from both the add-cases and add-variables approaches and restructuring data from wide to long formats and vice versa.
- Create publication quality graphs including bar, histogram, pie, line, scatter, regression, box, error bar, and interaction plots.
- Perform the basic types of analyses to measure strength of association and group differences and be able to know where to turn to cover much more complex methods.

Who This Book Is For

This book is, of course, for people who already know Stata. It may also be useful to R users wishing to learn Stata. However, we explain none of the Stata programs, only the R ones and how the packages differ, so it is not ideal for that purpose.

This book is based on *R for SAS and SPSS Users* [34]. However, there is quite a bit of additional material covered here, and, of course, the comparative coverage is completely different.

Who This Book Is Not For

We make no effort to teach statistics or graphics. Although we briefly state the goal and assumptions of each analysis, we do not cover their formulas or derivations. We have more than enough to discuss without tackling those topics too. This is also not a book about writing R functions, it is about using the thousands that already exist. We will write only a few very short functions. If you want to learn more about writing functions, we recommend John Chamber's *Software for Data Analysis: Programming with R* [5]. However, if you know Stata, reading this book should ease your transition to more complex books like that.

Practice Data Sets and Programs

All of the programs, data sets, and files that we use in this book are available for download at `http://r4stats.com`. A file containing corrections and clarifications is also available there.

Acknowledgments

We are very grateful for the many people who have helped make this book possible, including the developers of the S language on which R is based, Rick Becker, John Chambers, and Allan Wilks; the people who started R itself, Ross Ihaka and Robert Gentleman; the many other R developers for providing such wonderful tools for free and all the R-help participants who have kindly answered so many questions. Virtually all of the examples we present here are modestly tweaked versions of countless posts to the R-help discussion list, as well as a few Statalist posts. All we add is the selection, organization, explanation, and comparison.

We are especially grateful to the people who provided advice, caught typos, and suggested improvements, including Raymond R. Balise, Patrick Burns, Peter Flom, Chun Huang, Martin Gregory, Warren Lambert, Mathew Marler, Ralph O'Brien, Wayne Richter, Charilaos Skiadas, Andreas Stefik, Phil Spector, Michael Wexler, Graham Williams, Andrew Yee, and several anonymous reviewers.

A special thanks goes to Hadley Wickham, who provided much guidance on his `ggplot2` graphics package. Thanks to Gabor Grothendieck, Lauri Nikkinen, and Marc Schwarz and for the R-Help help discussion that led to Section 10.14: "Selecting First or Last Observations per Group." Thanks to Gabor Grothendieck also for a detailed discussion that lead to Section 10.4: "Multiple Conditional Transformations." Thanks to Michael A. McGuire for his assistance with all things Macintosh.

The first author is grateful to his wife, Carla Foust, and sons Alexander and Conor, who put up with many lost weekends as he wrote this book.

The second author wishes to thank Springer editor John Kimmel for suggesting his participation in this project and his wife, Cheryl, children Heather, Michael and Mitchell, and Sirr for their patience while he spent time away from them working on this book.

Robert A. Muenchen
muenchen.bob@gmail.com
Knoxville, Tennessee
January 2010

Joseph M. Hilbe
hilbe@asu.edu
Florence, Arizona
January 2010

About the Authors

Robert A. Muenchen is a consulting statistician and author of the book, *R for SAS and SPSS Users* [34]. He is currently the manager of Research Computing Support (formerly the Statistical Consulting Center) at the University of Tennessee. Bob has conducted research for a variety of public and private organizations and has co-authored over 50 articles in scientific journals and conference proceedings.

Bob has served on the advisory boards of the SAS Institute, SPSS Inc., the Statistical Graphics Corporation, and *PC Week Magazine*. His suggested improvements have been incorporated into SAS, SPSS, JMP, STATGRAPHICS, and several R packages.

His research interests include statistical computing, data graphics and visualization, text analysis, data mining, psychometrics, and resampling.

Joseph M. Hilbe is Solar System Ambassador with NASA/Jet Propulsion Laboratory, California Institute of Technology, an adjunct professor of statistics at Arizona State, and emeritus professor at the University of Hawaii. He is an elected Fellow of the American Statistical Association and of the Royal Statistical Society and is an elected member of the International Statistical Institute.

Professor Hilbe was the first editor of the *Stata Technical Bulletin*, later to become the *Stata Journal*, and was one of Stata Corporation's first senior statisticians (1991–1993). Hilbe is also the author of a number of textbooks, including *Logistic Regression Models* [21], *Negative Binomial Regression* [23], and with J. Hardin, *Generalized Linear Models and Extensions*, 2nd ed. [18] and *Generalized Estimating Equations* [19].

S-PLUS®is a registered trademark of Tibco, Inc.
SAS®is a registered trademark of SAS Institute.
SPSS®is a registered trademark of SPSS, Inc.
Stata®is a registered trademark of Statacorp, Inc.
STATISTICA®is a trademark of StatSoft, Inc.
Windows 7®, Windows Vista®, and Windows XP®are registered trademarks of Microsoft, Inc.
Macintosh®and Mac OS®are registered trademarks of Apple, Inc.
Copyright ©2010 Robert A. Muenchen and Joseph M. Hilbe. All rights reserved.

Contents

Preface . . . v

1 Introduction . . . 1
1.1 Overview . . . 1
1.2 Similarities Between R and Stata . . . 2
1.3 Why Learn R? . . . 3
1.4 Is R Accurate? . . . 4
1.5 What About Tech Support? . . . 4
1.6 Getting Started Quickly . . . 5
1.7 Programming Conventions . . . 5
1.8 Typographic Conventions . . . 6

2 Installing and Updating R . . . 9
2.1 Installing Add-on Packages . . . 10
2.2 Loading an Add-on Package . . . 10
2.3 Updating Your Installation . . . 14
2.4 Uninstalling R . . . 15
2.5 Choosing Repositories . . . 15
2.6 Accessing Data in Packages . . . 17

3 Running R . . . 19
3.1 Running R Interactively on Windows . . . 19
3.2 Running R Interactively on Macintosh . . . 21
3.3 Running R Interactively on Linux or UNIX . . . 23
3.4 Running Programs That Include Other Programs . . . 25
3.5 Running R in Batch Mode . . . 25
3.6 Graphical User Interfaces . . . 26
3.6.1 R Commander . . . 26
3.6.2 Rattle for Data Mining . . . 29
3.6.3 JGR Java GUI for R . . . 30

4 Help and Documentation 37
4.1 Introduction 37
4.2 Help Files 37
4.3 Starting Help 37
4.4 Help Examples 39
4.5 Help for Functions That Call Other Functions 40
4.6 Help for Packages 41
4.7 Help for Data Sets 42
4.8 Books and Manuals 42
4.9 E-mail Lists 42
4.10 Searching the Web 43
4.11 Vignettes 43

5 Programming Language Basics 45
5.1 Introduction 45
5.2 Simple Calculations 46
5.3 Data Structures 47
5.3.1 Vectors 47
5.3.2 Factors 51
5.3.3 Data Frames 56
5.3.4 Matrices 60
5.3.5 Arrays 63
5.3.6 Lists 63
5.4 Saving Your Work 67
5.5 Comments to Document Your Programs 69
5.6 Controlling Functions (Commands) 70
5.6.1 Controlling Functions with Arguments 70
5.6.2 Controlling Functions with Formulas 72
5.6.3 Controlling Functions with an Object's Class 73
5.6.4 Controlling Functions with Extractor Functions 75
5.7 How Much Output is There? 77
5.8 Writing Your Own Functions (Macros) 81
5.9 R Program Demonstrating Programming Basics 84

6 Data Acquisition 91
6.1 The R Data Editor 91
6.2 Reading Delimited Text Files 93
6.2.1 Reading Comma-Delimited Text Files 94
6.2.2 Reading Tab-Delimited Text Files 95
6.2.3 Missing Values for Character Variables 97
6.2.4 Trouble with Tabs 98
6.2.5 Skipping Variables in Delimited Files 99
6.2.6 Example Programs for Reading Delimited Text Files 100
6.3 Reading Text Data Within a Program 102

6.3.1 The Easy Approach ... 102
6.3.2 The More General Approach ... 104
6.3.3 Example Programs for Reading Text Data Within a Program ... 104
6.4 Reading Fixed-Width Text Files, One Record per Case ... 106
6.4.1 Macro Substitution ... 109
6.4.2 Example Programs for Reading Fixed-Width Text Files, One Record Per Case ... 110
6.5 Reading Fixed-Width Text Files, Two or More Records per Case ... 111
6.5.1 Example Programs to Read Fixed-Width Text Files with Two Records per Case ... 112
6.6 Importing Data from Stata into R ... 113
6.6.1 R Program to Import Data from Stata ... 114
6.7 Writing Data to a Comma-Delimited Text File ... 114
6.7.1 Example Programs for Writing a Comma-Delimited File ... 115
6.8 Exporting Data from R to Stata ... 116

7 Selecting Variables ... 119
7.1 Selecting Variables in Stata ... 119
7.2 Selecting All Variables ... 120
7.3 Selecting Variables Using Index Numbers ... 120
7.4 Selecting Variables Using Column Names ... 123
7.5 Selecting Variables Using Logic ... 124
7.6 Selecting Variables Using String Search ... 126
7.7 Selecting Variables Using $ Notation ... 128
7.8 Selecting Variables Using Component Names ... 129
7.8.1 The `attach` Function ... 129
7.8.2 The `with` Function ... 130
7.8.3 Using Component Names in Formulas ... 130
7.9 Selecting Variables with the `subset` Function ... 131
7.10 Selecting Variables Using List Index ... 132
7.11 Generating Indexes A to Z from Two Variable Names ... 132
7.12 Saving Selected Variables to a New Dataset ... 133
7.13 Example Programs for Variable Selection ... 134
7.13.1 Stata Program to Select Variables ... 134
7.13.2 R Program to Select Variables ... 134

8 Selecting Observations ... 139
8.1 Selecting Observations in Stata ... 139
8.2 Selecting All Observations ... 140
8.3 Selecting Observations Using Index Numbers ... 140
8.4 Selecting Observations Using Row Names ... 143
8.5 Selecting Observations Using Logic ... 145

8.6 Selecting Observations Using String Search ... 148
8.7 Selecting Observations Using the `subset` Function ... 150
8.8 Generating Indexes A to Z from Two Row Names ... 151
8.9 Variable Selection Methods with No Counterpart for Selecting Observations ... 152
8.10 Saving Selected Observations to a New Data Frame ... 152
8.11 Example Programs for Selecting Observations ... 152
8.11.1 Stata Program to Select Observations ... 153
8.11.2 R Program to Select Observations ... 153

9 Selecting Variables and Observations ... 157
9.1 The `subset` Function ... 157
9.2 Selecting Observations by Logic and Variables by Name ... 158
9.3 Using Names to Select Both Observations and Variables ... 159
9.4 Using Numeric Index Values to Select Both Observations and Variables ... 160
9.5 Using Logic to Select Both Observations and Variables ... 161
9.6 Saving and Loading Subsets ... 162
9.7 Example Programs for Selecting Variables and Observations ... 162
9.7.1 Stata Program for Selecting Variables and Observations ... 162
9.7.2 R Program for Selecting Variables and Observations ... 163

10 Data Management ... 167
10.1 Transforming Variables ... 167
10.1.1 Example Programs for Transforming Variables ... 171
10.2 Functions or Commands? The `apply` Function Decides ... 172
10.2.1 Applying the `mean` Function ... 173
10.2.2 Finding N or NVALID ... 176
10.2.3 Example Programs for Applying Statistical Functions ... 178
10.3 Conditional Transformations ... 180
10.3.1 Example Programs for Conditional Transformations ... 182
10.4 Multiple Conditional Transformations ... 183
10.4.1 Example Programs for Multiple Conditional Transformations ... 185
10.5 Missing Values ... 186
10.5.1 Substituting Means for Missing Values ... 188
10.5.2 Finding Complete Observations ... 189
10.5.3 When “99” Has Meaning ... 190
10.5.4 Example Programs to Assign Missing Values ... 192
10.6 Renaming Variables (and Observations) ... 194
10.6.1 Renaming Variables—Advanced Examples ... 196
10.6.2 Renaming by Index ... 197
10.6.3 Renaming by Column Name ... 198

10.6.4 Renaming Many Sequentially Numbered Variable Names . . . 199
10.6.5 Renaming Observations . . . 200
10.6.6 Example Programs for Renaming Variables . . . 200
10.7 Recoding Variables . . . 204
10.7.1 Recoding a Few Variables . . . 205
10.7.2 Recoding Many Variables . . . 205
10.7.3 Example Programs for Recoding Variables . . . 208
10.8 Keeping and Dropping Variables . . . 209
10.8.1 Example Programs for Keeping and Dropping Variables . . . 210
10.9 Stacking/Appending Data Sets . . . 210
10.9.1 Example Programs for Stacking/Appending Data Sets . . . 213
10.10 Joining/Merging Data Sets . . . 214
10.10.1 Example Programs for Joining/Merging Data Sets . . . 217
10.11 Creating Collapsed or Aggregated Data Sets . . . 219
10.11.1 The `aggregate` Function . . . 219
10.11.2 The `tapply` Function . . . 221
10.11.3 Merging Aggregates with Original Data . . . 222
10.11.4 Tabular Aggregation . . . 224
10.11.5 The `reshape` Package . . . 226
10.11.6 Example Programs for Collapsing/Aggregating Data . . . 226
10.12 By or Split-File Processing . . . 228
10.12.1 Comparing Summarization Methods . . . 232
10.12.2 Example Programs for By or Split-file Processing . . . 233
10.13 Removing Duplicate Observations . . . 234
10.13.1 Example Programs for Removing Duplicate Observations . . . 236
10.14 Selecting First or Last Observations per Group . . . 237
10.14.1 Example Programs for Selecting Last Observation per Group . . . 239
10.15 Reshaping Variables to Observations and Back . . . 240
10.15.1 Example Programs for Reshaping Variables to Observations and Back . . . 242
10.16 Sorting Data Frames . . . 243
10.16.1 Example Programs for Sorting Data Sets . . . 246
10.17 Converting Data Structures . . . 247
10.17.1 Converting from Logical to Numeric Index and Back . . . 250

11 Enhancing Your Output . . . 253
11.1 Value Labels or Formats (and Measurement Level) . . . 253
11.1.1 Character Factors . . . 254
11.1.2 Numeric Factors . . . 256

11.1.3 Making Factors of Many Variables 258
11.1.4 Converting Factors into Numeric or Character Variables 260
11.1.5 Dropping Factor Levels 262
11.1.6 Example Programs for Value Labels or Formats 263
11.2 Variable Labels 266
11.2.1 Variable Labels in The `Hmisc` Package 266
11.2.2 Long Variable Names as Labels 267
11.2.3 Other Packages That Support Variable Labels 270
11.2.4 Example Programs for Variable Labels 270
11.3 Output for Word Processing and Web Pages 271
11.3.1 The `xtable` Package 272
11.3.2 Other Options for Formatting Output 274
11.3.3 Example Programs for Formatting Output 275

12 Generating Data 277
12.1 Generating Numeric Sequences 278
12.2 Generating Factors 279
12.3 Generating Repetitious Patterns (Not Factors) 280
12.4 Generating Integer Measures 281
12.5 Generating Continuous Measures 283
12.6 Generating a Data Frame 285
12.7 Example Programs for Generating Data 285
12.7.1 Stata Program for Generating Data 285
12.7.2 R Program for Generating Data 286

13 Managing Your Files and Workspace 291
13.1 Loading and Listing Objects 291
13.2 Understanding Your Search Path 294
13.3 Attaching Data Frames 296
13.4 Attaching Files 298
13.5 Removing Objects from Your Workspace 299
13.6 Minimizing Your Workspace 301
13.7 Setting Your Working Directory 301
13.8 Saving Your Workspace 302
13.8.1 Saving Your Workspace Manually 302
13.8.2 Saving Your Workspace Automatically 303
13.9 Getting Operating Systems to Show You "*.RData*" Files 303
13.10 Organizing Projects with Windows Shortcuts 304
13.11 Saving Your Programs and Output 304
13.12 Saving Your History 305
13.13 Large Data Set Considerations 305
13.14 Example R Program for Managing Files and Workspace 307

14 Graphics Overview 311
14.1 Stata Graphics 312
14.2 R Graphics 312
14.3 The Grammar of Graphics 313
14.4 Other Graphics Packages 315
14.5 Graphics Procedures and Graphics Systems 315
14.6 Graphics Devices 316
14.7 Practice Data: mydata100 318

15 Traditional Graphics 319
15.1 Bar Plots 319
15.1.1 Bar Plots of Counts 319
15.1.2 Bar Plots for Subgroups of Counts 324
15.1.3 Bar Plots of Means 326
15.2 Adding Titles, Labels, Colors, and Legends 327
15.3 Graphics ***Par***ameters and Multiple Plots on a Page 330
15.4 Pie Charts 331
15.5 Dot Charts 333
15.6 Histograms 333
15.6.1 Basic Histograms 334
15.6.2 Histograms Stacked 336
15.6.3 Histograms Overlaid 337
15.7 Normal QQ Plots 341
15.8 Strip Charts 342
15.9 Scatter Plots and Line Plots 347
15.9.1 Scatter plots with Jitter 350
15.9.2 Scatter plots with Large Data Sets 350
15.9.3 Scatter plots with Lines 352
15.9.4 Scatter plots with Linear Fit by Group 353
15.9.5 Scatter plots by Group or Level (Coplots) 354
15.9.6 Scatter plots with Confidence Ellipse 356
15.9.7 Scatter plots with Confidence and Prediction Intervals 357
15.9.8 Plotting Labels Instead of Points 362
15.9.9 Scatter plot Matrices 364
15.10 Dual-Axes Plots 366
15.11 Box Plots 368
15.12 Error Bar Plots 370
15.13 Interaction Plots 370
15.14 Adding Equations and Symbols to Graphs 371
15.15 Summary of Graphics Elements and Parameters 372
15.16 Plot Demonstrating Many Modifications 373
15.17 Example Program for Traditional Graphics 374
15.17.1 Stata Program for Traditional Graphics 375
15.17.2 R Program for Traditional Graphics 375

16 Graphics with `ggplot2` 385
16.1 Introduction 385
16.1.1 Overview `qplot` and `ggplot` 386
16.1.2 Missing Values 387
16.1.3 Typographic Conventions 388
16.2 Bar Plots 389
16.2.1 Pie Charts 392
16.2.2 Bar Charts for Groups 393
16.3 Plots by Group or Level 394
16.4 Presummarized Data 396
16.5 Dot Charts 397
16.6 Adding Titles and Labels 399
16.7 Histograms and Density Plots 400
16.7.1 Histograms 400
16.7.2 Density Plots 401
16.7.3 Histograms with Density Overlaid 401
16.7.4 Histograms for Groups, Stacked 403
16.7.5 Histograms for Groups, Overlaid 404
16.8 Normal QQ Plots 405
16.9 Strip Plots 405
16.10 Scatter Plots and Line Plots 408
16.10.1 Scatter Plots with Jitter 410
16.10.2 Scatter Plots for Large Data Sets 411
16.10.3 Hexbin Plots 414
16.10.4 Scatter Plots with Fit Lines 415
16.10.5 Scatter Plots with Reference Lines 416
16.10.6 Scatter Plots with Labels Instead of Points 420
16.10.7 Changing Plot Symbols 421
16.10.8 Scatter Plot with Linear Fits by Group 422
16.10.9 Scatter Plots Faceted for Groups 422
16.10.10 Scatter Plot Matrix 424
16.11 Box Plots 425
16.12 Error Bar Plots 428
16.13 Logarithmic Axes 430
16.14 Aspect Ratio 430
16.15 Multiple Plots on a Page 431
16.16 Saving `ggplot2` Graphs to a File 433
16.17 An Example Specifying All Defaults 433
16.18 Summary of Graphic Elements and Parameters 435
16.19 Example Programs for ggplot2 436

17 Statistics 453
17.1 Scientific Notation 453
17.2 Descriptive Statistics 454
17.2.1 The `Hmisc` `describe` Function 454
17.2.2 The `summary` Function 456

17.2.3 The `table` Function and Its Relatives ... 457
17.2.4 The `mean` Function and Its Relatives ... 459
17.3 Cross-Tabulation ... 460
17.3.1 The `CrossTable` Function ... 460
17.3.2 The `tables` and `chisq.test` Functions ... 462
17.4 Correlation ... 465
17.4.1 The `cor` Function ... 468
17.5 Linear Regression ... 470
17.5.1 Plotting Diagnostics ... 473
17.5.2 Comparing Models ... 474
17.5.3 Making Predictions with New Data ... 475
17.6 t-Test: Independent Groups ... 476
17.7 Equality of Variance ... 477
17.8 t-Test: Paired or Repeated Measures ... 478
17.9 Wilcoxon Mann-Whitney Rank Sum Test: Independent Groups ... 479
17.10 Wilcoxon Signed-Rank Test: Paired Groups ... 480
17.11 Analysis of Variance ... 481
17.12 Sums of Squares ... 486
17.13 The Kruskal–Wallis Test ... 487
17.14 Example Programs for Statistical Tests ... 489
17.14.1 Stata Program for Statistical Tests ... 489
17.14.2 R Program for Statistical Tests ... 491

18 Conclusion ... 497

Glossary of R jargon ... 499

Comparison of Stata commands and R functions ... 505

Automating Your R Setup ... 507
C.1 Setting Options ... 507
C.2 Creating Objects ... 508
C.3 Loading Packages ... 508
C.4 Running Functions ... 508
C.5 Example .Rprofile ... 510

Example Simulation ... 511
D.1 Stata Example Simulation ... 511
D.2 R Example Simulation ... 512

References ... 513

Index ... 517

List of Tables

5.1 Example formulas in Stata and R. 73
5.2 Modes and classes of various R objects. 74

10.1 Mathematical operators in Stata and R. 168
10.2 Logical operators in Stata and R. 181
10.3 Comparison of summarization functions. 232
10.4 Data conversion functions. 249

11.1 Our practice data set printed in LaTeX. 272
11.2 Linear model results formatted by `xtable` 274

13.1 Workspace management functions 306

14.1 A comparison of R's three main graphics packages 313

15.1 Graphics arguments for use with traditional high-level graphics functions. 344
15.2 Graphics *par*ameters to set or query using only `par()`. 345
15.3 Graphics parameters for both `par` and graphics functions. 346
15.4 Graphics functions to add elements to existing plots. 347

16.1 Comparison of the `qplot` and `ggplot` functions. 388

B.1 Comparison of Stata commands and functions to R functions. 506

List of Figures

2.1 Choosing an R mirror site and package 11
2.2 Display of packages in the library 12
2.3 Display of R package repositories 16
2.4 Display of practice data sets 17
2.5 Display of data sets in the `car` package 18

3.1 R graphical user interface on Microsoft Windows. 20
3.2 R graphical user interface on Macintosh. 22
3.3 R Commander GUI at startup 27
3.4 R Commander's Data Viewer 28
3.5 R Commander GUI in use 28
3.6 Rattle GUI for data mining. 30
3.7 Rattle writing an R program 31
3.8 JGR program editor 31
3.9 JGR offering a list of arguments 32
3.10 JGR's Package Manager 33
3.11 JGR's Object Browser 34
3.12 JGR's DataTable editor 35

4.1 R's main help window 38
4.2 Example results for searching help files 41
4.3 Help window for the `foreign` package. 41

5.1 Help file for the `mean` function. 71

6.1 Adding a new variable in the data editor. 92
6.2 R data editor with practice data 92

10.1 Renaming a variable using R's data editor. 195
10.2 Renaming variables using the `edit` function. 196

12.1 Barplots of generated data 282

12.2 Histograms of generated data 284

14.1 Napoleon's march to Moscow 314

15.1 Unlabeled bar plot using traditional graphics. 320
15.2 Bad bar plot of unsummarized variable q4. 320
15.3 Good bar plot of variable q4 321
15.4 Barplot of gender 322
15.5 Horizontal bar plot 323
15.6 An unlabeled stacked bar plot of workshop. 323
15.7 Stacked bar plot of workshop split by gender 324
15.8 Mosaic plot of workshop by gender from `plot`. 325
15.9 Mosaic plot of workshop by gender from `mosaicplot` 325
15.10 Mosaic plot with three factors 326
15.11 Bar plot of means 327
15.12 Bar plot of means by workshop and gender 328
15.13 Bar plot with title and legend 328
15.14 Bar plots of counts by workshop and gender 332
15.15 Pie chart of workshop 332
15.16 Dotchart of workshop within gender 333
15.17 Histogram of posttest 334
15.18 Histogram of posttest with kernel density curve, tick-marks 335
15.19 Histogram of posttest, males only 336
15.20 Multiframe plot of histograms for all, males 337
15.21 Histogram of posttest overlaid with males 338
15.22 Histogram with breakpoints chosen from another 340
15.23 Normal quantile plot 341
15.24 Strip chart with jitter and stack 343
15.25 Strip chart of posttest by workshop 344
15.26 Scatter plot of pretest and posttest 348
15.27 Scatter plots demonstrating various types 349
15.28 Scatter plots demonstrating jitter 350
15.29 Scatter plots on large data sets 351
15.30 Hexbin plot 351
15.31 Scatter plot with various lines added 352
15.32 Scatter plot with lines and points by gender 354
15.33 Scatter plots conditioned on workshop 355
15.34 Scatter plots conditioned on q1 356
15.35 Scatter plot with confidence ellipse 357
15.36 Scatter plot to build on 358
15.37 Scatter plot simulating confidence intervals 359
15.38 Scatter plot with confidence intervals 362
15.39 Scatter plot using gender as point symbols 363
15.40 Scatter plot plotting character strings as points 363
15.41 Scatter plot matrix 364

15.42 Scatter plot matrix with smoothed fits . . . 366
15.43 Scatter plot with dual axes . . . 367
15.44 box plot of posttest by workshop . . . 368
15.45 box plots of various types . . . 369
15.46 Errorbar plot of posttest by workshop . . . 371
15.47 Interaction plot of posttest by workshop and gender . . . 372
15.48 Plot demonstrating many annotations . . . 373

16.1 Bar plot done with `ggplot2` package . . . 389
16.2 Horizontal bar plot . . . 390
16.3 Stacked bar plot of workshop . . . 391
16.4 Pie chart of workshop . . . 392
16.5 Barplot types of stack, fill, and dodge . . . 394
16.6 Barplots of workshop for each gender . . . 395
16.7 Barplot of presummarized data . . . 397
16.8 Dotchart of workshop by gender . . . 398
16.9 Barplot demonstrating titles and labels . . . 399
16.10 Histogram of posttest . . . 401
16.11 Histogram with smaller bins . . . 402
16.12 Density plot of posttest . . . 402
16.13 Histogram with density curve . . . 403
16.14 Histograms of posttest by gender . . . 404
16.15 Histogram with bars filled by gender . . . 405
16.16 Normal quantile plot . . . 406
16.17 Strip chart with jitter . . . 406
16.18 Strip chart by workshop . . . 408
16.19 Scatter plots demonstrating various line types . . . 409
16.20 Scatter plots showing effect of jitter . . . 411
16.21 Scatter plot showing transparency . . . 412
16.22 Scatter plot with contour lines . . . 414
16.23 Hexbin plot of pretest and posttest . . . 415
16.24 Scatter plot with regression line and confidence band . . . 416
16.25 Scatter plot with regression line but no confidence band . . . 417
16.26 Scatter plot with y=x line added . . . 418
16.27 Scatter plot with vertical and horizontal reference lines . . . 418
16.28 Scatter plot with a multiple vertical reference lines . . . 419
16.29 Scatter plot using labels as points . . . 420
16.30 Scatter plot with point shape determined by gender . . . 421
16.31 Scatter plot showing regression fits determined by gender . . . 422
16.32 Scatter Plots with regression fits by workshop and gender . . . 423
16.33 Scatter plot matrix with lowess fits and density curves . . . 424
16.34 box plot of posttest . . . 425
16.35 box plot of posttest by group with jitter . . . 426
16.36 box plot of posttest by workshop and gender . . . 428
16.37 Error bar plot of posttest by workshop . . . 429

16.38 Multiframe demonstration plot 431
16.39 Scatter plot programmed several ways 434

17.1 Diagnostic plots for regression 474
17.2 Tukey HSD plot 485

1

Introduction

1.1 Overview

R [38] is a powerful and flexible environment for research computing. Written by Ross Ihaka, Robert Gentleman (hence the name “R”), the R Core Development Team, and an army of volunteers, R provides a wider range of analytical and graphical commands than any other software. The fact that this level of power is available free of charge has dramatically changed the landscape of research software.

R is a variation of the S language, developed by John Chambers, Rick Becker, and others at Bell Labs[1]. The Association of Computing Machinery presented John Chambers with a Software System Award and said that the S language “*. . . will forever alter the way people analyze, visualize, and manipulate data. . .*” and went on to say that it is “*. . . an elegant, widely accepted, and enduring software system, with conceptual integrity. . . .*” The original S language is still commercially available as Tibco Spotfire S+. Most programs written in the S language will run in R.

Stata, a product of Stata Corporation, has not yet incorporated an interface to R in its software, but users have already posted programs to use R within the Stata environment. It is expected that more facilities of this sort will be developed in the near future.

For each aspect of R we discuss, we will compare and contrast it with Stata. Many of the topics end with example programs that do almost identical things in both software applications. R programs are often longer than similar Stata code, but this is typically the case because R functions are more specific than Stata commands.

Many R functions will appear familiar to Stata users; that is, R functions such as `lm` or `glm` will appear somewhat similar to Stata’s `regress` and `glm` commands. There are other aspects of the two languages, however, that may

[1] For a fascinating history of S and R, see Appendix A of *Software for Data Analysis: Programming with R* [5].

R.A. Muenchen, J.M. Hilbe, *R for Stata Users*, Statistics
and Computing, DOI 10.1007/978-1-4419-1318-0_1,
© Springer Science+Business Media, LLC 2010

appear more confusing at first. We hope to ease that confusion by focusing on both the similarities and differences between R and Stata in this text. When we examine a particular analysis (e.g., comparing two groups with a t-test) someone who knows Stata will have very little trouble figuring out what R is doing. However, the basics of the R language are very different, so that is where we will spend the majority of our time.

We introduce topics in a carefully chosen order, so it is best to read from beginning to end the first time through, even if you think you do not need to know a particular topic. Later you can skip directly to the section you need. We include a fair amount of redundancy on key topics to help teach those topics and to make it easier to read just one section as a future reference. The glossary in Appendix A defines R concepts in terms that Stata users will understand and provides parallel definitions using R terminology.

1.2 Similarities Between R and Stata

Stata is an excellent statistics package. One of the authors has used Stata for over 20 years and has authored many Stata commands.

Perhaps more than any other two research computing environments, R and Stata share many of the features that make them outstanding:

- Both include rich programming languages designed for writing new analytic methods, not just a set of prewritten commands.
- Both contain extensive sets of analytic commands written in their own languages.
- The pre-written commands in R, and most in Stata, are visible and open for you to change as you please.
- Both save command or function output in a form you can easily use as input to further analysis.
- Both do modeling in a way that allows you to readily apply your models for tasks such as making predictions on new data sets. Stata calls these *postestimation commands* and R calls them *extractor functions.*
- In both, when you write a new command, it is on an equal footing with commands written by the developers. There are no additional "Developer's Kits" to purchase.
- Both have legions of devoted users who have written numerous extensions and who continue to add the latest methods many years before their competitors.
- Both can search the Internet for user-written commands and download them automatically to extend their capabilities quickly and easily.
- Both hold their data in the computer's main memory, offering speed but limiting the amount of data they can handle.

1.3 Why Learn R?

With so many similarities, if you already know Stata, why should you bother to learn R?

- To augment Stata; i.e. to be able to perform statistical analyses that are not available in Stata, but which are available in R. R offers a *vast* number of analytical methods. There are now over 3,000 add-on packages available for R and this number is growing at an exponential rate. Therefore, knowing both gives you a much greater range of tools for analyzing data.
- To stay current with new analytic methods. The majority of statistics textbooks, and journal articles, now being published use either Stata or R for examples. R appears to be used more in many journals. Stata users not understanding R are therefore not able to learn as much from texts or articles using R for examples than they would be if they understood the language.
- If you continue to do all of your data management in Stata, you can learn just enough R to import your data and run the procedures you need.
- R is directly accessible from inside many statistics packages. SAS, SPSS, and STATISTICA offer the ability to run R programs from within their software. This means that when developers write programs in R, they are assured a very wide audience. Roger Newson has written an interface [36] between Stata and R that provides some of this ability. We expect to see more done on this topic in the near future.
- R has been object-oriented since its first version. Many of its commands sense the types of data structures you have and do the best thing for each. For example, once you tell it that gender is a categorical variable, it will take statistically proper actions if you use it as a linear regression predictor. At the time of publication, Stata Corporation had just announced its future move toward object orientation.
- Both languages consist of a core set of functions that are written in the C language. However, only developers at Stata Corporation can modify its most fundamental commands. Every aspect of R is open for anyone to modify in any way they like. This complete flexibility attracts many developers.
- Both R and Stata offer graphics that are flexible, easy to use, and of high quality. However, R also offers the very flexible and powerful Grammar of Graphics approach. As we will see, developers have even gone so far as replacing R's core graphical system.
- R is free. This means, of course, that you can use it for free, but it also means developers know that their work is available to everyone. That helps attract developers and is a major reason that there are so many add-on packages for it.

1.4 Is R Accurate?

When people first learn of R, one of their first questions is "Can a package written by volunteers be as accurate as one written by a large corporation?" People envision a lone programmer competing against a large corporate team. Having worked closely with several software companies over the years, we can assure you that this is not the case. A particular procedure is usually written by one programmer, even at Stata Corporation. A thorough testing process is then carried out by a few people within the company and then more thoroughly by Stata users on publication of the new command or function.

The R Development Core Team runs each release of R through validation suites that have known correct answers to ensure accurate results. They also go through "Alpha," "Beta," and "Release Candidate" testing phases, which are open to the public. Each phase has tighter restrictions on modifications of R. Finally, the production version is released. The details of this process are provided in R: Regulatory Compliance and Validation Issues, A Guidance Document for the Use of R in Regulated Clinical Trial Environments, available at `http://www.r-project.org/doc/R-FDA.pdf` [11].

When bugs are found in Stata, the developers typically make a fix within days. Users are in continual communication with other users and developers through the Statalist. An average of 100 communications are posted daily. Questions are answered by other users or by Stata staff.

R also has open discussions of its known bugs and R's developers fix them quickly too. However, software of the complexity of Stata and R will never be completely free of errors, regardless of its source.

1.5 What About Tech Support?

If a package is free, who supports it?

Stata users may call toll-free or e-mail technical support for problems they experience with the software or for advice on how to run various software commands. The response is near immediate, with a day delay in response being on the high side. Even experienced Stata users sometimes require technical advice for new commands or functions or have difficulties learning new areas of statistics or new methodologies (e.g. the matrix programming). We have always found support to he helpful and friendly.

You can also get support through the Stata Listserver, where it is normal to get assistance from someone the very day you post your request.

R's main source of support is the R-help mailing list. Other users and often developers themselves will often provide immediate help. Sometimes you may obtain different answers from various responders, but that is part of the nature of statistics. For details on the various R e-mail support lists, see Chapter 4, "Help and Documentation."

There are several commercial versions of R available, and the companies that sell them do provide phone support. Here are some of these companies and their web sites:

XL-Solutions Corp., `http://www.experience-rplus.com/`
Revolution Computing, Inc., `http://www.revolution-computing.com/`
Random Technologies, LLC, `http://random-technologies-llc.com/`

1.6 Getting Started Quickly

If you wish to start using R quickly, you can do so by reading fewer than 50 pages of this book. Since you have Stata to do your basic descriptive statistics, you are likely to need R's modeling functions. Here are the steps you can follow to use them.

1. Read the remainder of this chapter and Chapter 2, "Installing and Updating R." Download and install R on your computer.
2. Read the part of Chapter 3, "Running R," that covers your operating system.
3. In Chapter 5, "Programming Language Basics," read Section 5.3.2 about factors, and Section 5.3.3 about data frames.
4. Also in Chapter 5, read Section 5.6.1, "Controlling Functions with Arguments," and Section 5.6.2, "Controlling Functions with Formulas," including Table 5.1, "Example formulas in Stata and R."
5. Read Section 6.6, "Importing Data from Stata."

After reading the pages above, do all your data management in Stata, stripping out observations containing any missing values. Then write out only the variables and observations you need to a comma separated values file, mydata.csv. Assuming your variables are named y, x1, x2,. . . , your entire R program will look something like this:

```
library("Hmisc")  # Contains stata.get function.
library("OtherLibrariesYouNeed")  # If you need any.
mydata <- stata.get("mydata.dta") # imports your Stata file
mymodel <- TheFunctionYouNeed( y ~ x1+x2, data=mydata )
summary(mymodel)
plot(mymodel) # if your function does plots.
```

1.7 Programming Conventions

Although R has many ways to generate practice data and has a variety of example data sets, we will use a tiny practice data set that is easy to enter. We can then manipulate and print it repeatedly so that you can clearly see the changes.