

A Handbook of
Statistical
Analyses
using **Stata**
Third Edition

A Handbook of
Statistical
Analyses
using **Stata**
Third Edition

Sophia Rabe-Hesketh
Brian Everitt

CHAPMAN & HALL/CRC

A CRC Press Company
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Rabe-Hesketh, S.

A handbook of statistical analyses using Stata / Sophia Rabe-Hesketh, Brian S. Everitt.—
[3rd ed.].

p. cm.

Includes bibliographical references and index.

ISBN 1-58488-404-5 (alk. paper)

1. Stata. 2. Mathematical statistics—Data processing. I. Everitt, Brian. II. Title.

QA276.4.R33 2003

519.5'0285'5369—dc22

2003065361

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the CRC Press Web site at www.crcpress.com

© 2004 by CRC Press LLC

No claim to original U.S. Government works

International Standard Book Number 1-58488-404-5

Library of Congress Card Number 2003065361

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

Printed on acid-free paper

Preface

Stata is an exciting statistical package that offers all standard and many non-standard methods of data analysis. In addition to general methods such as linear, logistic and Poisson regression and generalized linear models, Stata provides many more specialized analyses, such as generalized estimating equations from biostatistics and the Heckman selection model from econometrics. Stata has extensive capabilities for the analysis of survival data, time series, panel (or longitudinal) data, and complex survey data. For all estimation problems, inferences can be made more robust to model misspecification using bootstrapping or robust standard errors based on the sandwich estimator. In each new release of Stata, its capabilities are significantly enhanced by a team of excellent statisticians and developers at Stata Corporation.

Although extremely powerful, Stata is easy to use, either by point-and-click or through its intuitive command syntax. Applied researchers, students, and methodologists therefore all find Stata a rewarding environment for manipulating data, carrying out statistical analyses, and producing publication quality graphics.

Stata also provides a powerful programming language making it easy to implement a ‘tailor-made’ analysis for a particular application or to write more general commands for use by the wider Stata community. In fact we consider Stata an ideal environment for developing and disseminating new methodology. First, the elegance and consistency of the programming language appeals to the esthetic sense of methodologists. Second, it is simple to make new commands behave in every way like Stata’s own commands, making them accessible to applied researchers and students. Third, Stata’s emailing list *Statalist*, *The Stata Journal*, the Stata Users’ Group Meetings, and the Statistical Software Components (SSC) archive on the internet all make exchange and discussion of new commands extremely easy. For these reasons Stata is

constantly kept up-to-date with recent developments, not just by its own developers, but also by a very active Stata community.

This handbook follows the format of its two predecessors, *A Handbook of Statistical Analysis using S-PLUS* and *A Handbook of Statistical Analysis using SAS*. Each chapter deals with the analysis appropriate for a particular application. A brief account of the statistical background is included in each chapter including references to the literature, but the primary focus is on how to use Stata, and how to interpret results. Our hope is that this approach will provide a useful complement to the excellent but very extensive Stata manuals. The majority of the examples are drawn from areas in which the authors have most experience, but we hope that current and potential Stata users from outside these areas will have little trouble in identifying the relevance of the analyses described for their own data.

This third edition contains new chapters on random effects models, generalized estimating equations, and cluster analysis. We have also thoroughly revised all chapters and updated them to make use of new features introduced in Stata 8, in particular the much improved graphics.

Particular thanks are due to Nick Cox who provided us with extensive general comments for the second and third editions of our book, and also gave us clear guidance as to how best to use a number of Stata commands. We are also grateful to Anders Skrondal for commenting on several drafts of the current edition. Various people at Stata Corporation have been very helpful in preparing both the second and third editions of this book. We would also like to acknowledge the usefulness of the Stata Netcourses in the preparation of the first edition of this book.

All the datasets can be accessed on the internet at the following Web sites:

- <http://www.stata.com/texts/stas3>
- <http://www.iop.kcl.ac.uk/IoP/Departments/BioComp/stataBook.shtml>

S. Rabe-Hesketh
B. S. Everitt
London

Dedication

To my parents, Birgit and Georg Rabe
Sophia Rabe-Hesketh

To my wife, Mary Elizabeth
Brian S. Everitt

Contents

1 A Brief Introduction to Stata

- 1.1 Getting help and information
- 1.2 Running Stata
- 1.3 Conventions used in this book
- 1.4 Datasets in Stata
- 1.5 Stata commands
- 1.6 Data management
- 1.7 Estimation
- 1.8 Graphics
- 1.9 Stata as a calculator
- 1.10 Brief introduction to programming
- 1.11 Keeping Stata up to date
- 1.12 Exercises

2 Data Description and Simple Inference: Female Psychiatric Patients

- 2.1 Description of data
- 2.2 Group comparison and correlations
- 2.3 Analysis using Stata
- 2.4 Exercises

3 Multiple Regression: Determinants of Pollution in U.S. Cities

- 3.1 Description of data
- 3.2 The multiple regression model
- 3.3 Analysis using Stata
- 3.4 Exercises

4 Analysis of Variance I: Treating Hypertension

- 4.1 Description of data
- 4.2 Analysis of variance model
- 4.3 Analysis using Stata
- 4.4 Exercises
- 5 Analysis of Variance II: Effectiveness of Slimming Clinics**
 - 5.1 Description of data
 - 5.2 Analysis of variance model
 - 5.3 Analysis using Stata
 - 5.4 Exercises
- 6 Logistic Regression: Treatment of Lung Cancer and Diagnosis of Heart Attacks**
 - 6.1 Description of data
 - 6.2 The logistic regression model
 - 6.3 Analysis using Stata
 - 6.4 Exercises
- 7 Generalized Linear Models: Australian School Children**
 - 7.1 Description of data
 - 7.2 Generalized linear models
 - 7.3 Analysis using Stata
 - 7.4 Exercises
- 8 Summary Measure Analysis of Longitudinal Data: The Treatment of Post-Natal Depression**
 - 8.1 Description of data
 - 8.2 The analysis of longitudinal data
 - 8.3 Analysis using Stata
 - 8.4 Exercises
- 9 Random Effects Models: Thought disorder and schizophrenia**
 - 9.1 Description of data
 - 9.2 Random effects models
 - 9.3 Analysis using Stata
 - 9.4 Thought disorder data
 - 9.5 Exercises
- 10 Generalized Estimating Equations: Epileptic Seizures and Chemotherapy**
 - 10.1 Introduction
 - 10.2 Generalized estimating equations

10.3 Analysis using Stata

10.4 Exercises

11 Some Epidemiology

11.1 Description of data

11.2 Introduction to epidemiology

11.3 Analysis using Stata

11.4 Exercises

12 Survival Analysis: Retention of Heroin Addicts in Methadone Maintenance Treatment

12.1 Description of data

12.2 Survival analysis

12.3 Analysis using Stata

12.4 Exercises

13 Maximum Likelihood Estimation: Age of Onset of Schizophrenia

13.1 Description of data

13.2 Finite mixture distributions

13.3 Analysis using Stata

13.4 Exercises

14 Principal Components Analysis: Hearing Measurement using an Audiometer

14.1 Description of data

14.2 Principal component analysis

14.3 Analysis using Stata

14.4 Exercises

15 Cluster Analysis: Tibetan Skulls and Air Pollution in the USA

15.1 Description of data

15.2 Cluster analysis

15.3 Analysis using Stata

15.4 Exercises

Appendix: Answers to Selected Exercises

References

Distributors for Stata

The distributor for Stata in the United States is:

Stata Corporation

4905 Lakeway Drive
College Station, TX 77845

email: stata@stata.com

Web site: <http://www.stata.com>

Telephone: 979-696-4600

In the United Kingdom the distributor is:

Timberlake Consultants

Unit B3, Broomsleigh Business Park
Worsley Bridge Road
London SE26 5BN

email: info@timberlake.co.uk

Web site: <http://www.timberlake.co.uk>

Telephone: 44(0)-20-8697-3377

For a list of distributors in other countries, see the Stata Web page.

Chapter 1

A Brief Introduction to Stata

1.1 Getting help and information

Stata is a general purpose statistics package developed and maintained by Stata Corporation. There are several forms or ‘flavors’ of Stata, ‘Intercooled Stata’, the more limited ‘Small Stata’ and the extended ‘Stata/SE’ (Special Edition), differing mostly in the maximum size of dataset and processing speed. Each exists for Windows (98, 2000, XP, and NT), Unix platforms, and the Macintosh. In this book, we will describe Intercooled Stata for Windows although most features are shared by the other flavors of Stata.

The base documentation set for Stata consists of seven manuals: *Stata Getting Started*, *Stata User’s Guide*, *Stata Base Reference Manuals* (four volumes), and *Stata Graphics Reference Manual*. In addition there are more specialized reference manuals such as the *Stata Programming Reference Manual* and the *Stata Cross-Sectional Time-Series Reference Manual* (longitudinal data analysis). The reference manuals provide extremely detailed information on each command while the *User’s Guide* describes Stata more generally. Features that are specific to the operating system are described in the appropriate *Getting Started* manual, e.g., *Getting Started with Stata for Windows*.

Each Stata command has associated with it a help file that may be viewed within a Stata session using the help facility. Both the help-files and the manuals refer to the *Base Reference Manuals* by [R] **name of entry**, to the *User’s Guide* by [U] **chapter or section number and name**, the *Graphics Manual* by [G] **name of entry**, etc. (see *Stata*

Getting Started manual, immediately after the table of contents, for a complete list).

There are an increasing number of books on Stata, including Hamilton (2004) and Kohler and Kreuter (2004), as well as books in German, French, and Spanish. Excellent books on Stata for particular types of analysis include Hills and De Stavola (2002), *A Short Introduction to Stata for Biostatistics*, Long and Freese (2003), *Regression Models for Categorical Dependent Variables using Stata*, Cleves, Gould and Gutierrez (2004), *An Introduction to Survival Analysis Using Stata*, and Hardin and Hilbe (2001), *Generalized Linear Models and Extensions*. See <http://www.stata.com/bookstore/statabooks.html> for up-to-date information on these and other books.

The Stata Web page at <http://www.stata.com> offers much useful information for learning Stata including an extensive series of ‘frequently asked questions’ (FAQs). Stata also offers internet courses, called *netcourses*. These courses take place via a temporary mailing list for course organizers and ‘attenders’. Each week, the course organizers send out lecture notes and exercises which the attenders can discuss with each other until the organizers send out the answers to the exercises and to the questions raised by attenders.

The UCLA Academic Technology Services offer useful textbook and paper examples at <http://www.ats.ucla.edu/stat/stata/>, showing how analyses can be carried out using Stata. Also very helpful for learning Stata are the regular columns *From the helpdesk* and *Speaking Stata* in *The Stata Journal*; see www.stata-journal.com.

One of the exciting aspects of being a Stata user is being part of a very active Stata community as reflected in the busy *Statalist* mailing list, Stata Users’ Group meetings taking place every year in the UK, USA and various other countries, and the large number of user-contributed programs; see also [Section 1.11](#). *Statalist* also functions as a technical support service with Stata staff and expert users such as Nick Cox offering very helpful responses to questions.

1.2 Running Stata

This section gives an overview of what happens in a typical Stata session, referring to subsequent sections for more details.

1.2.1 *Stata windows*

When Stata is started, a screen opens as shown in [Figure 1.1](#) containing four windows labeled:

- Stata Command
- Stata Results
- Review
- Variables

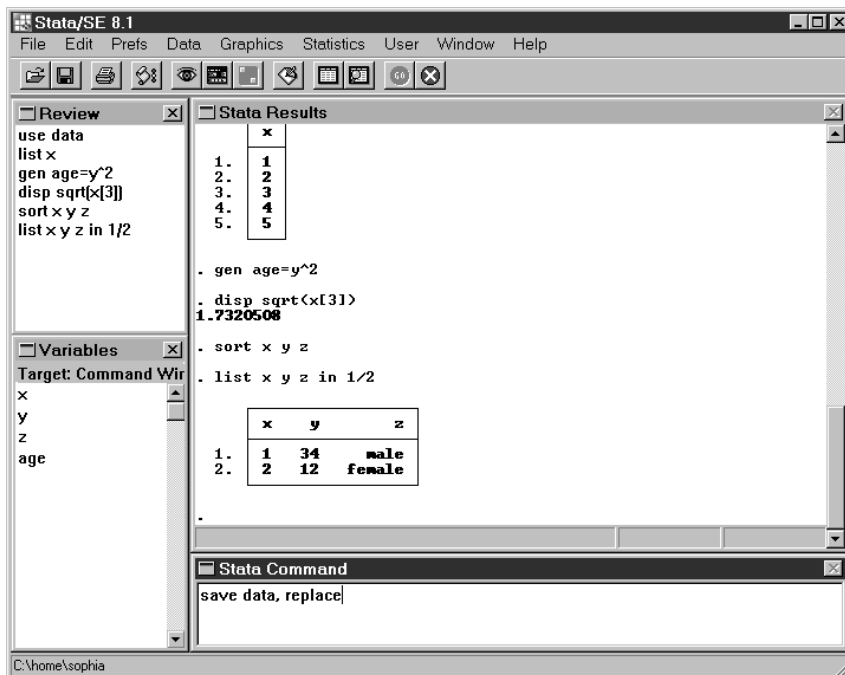





Figure 1.1: Stata windows.

Each of the Stata windows can be resized and moved around in the usual way; the Variables and Review windows can also be moved outside the main window. To bring a window forward that may be obscured by other windows, make the appropriate selection in the **Window** menu. The fonts in a window can be changed by clicking on the menu button  on the top left of that window's menu bar. All these settings are automatically saved when Stata is closed.

1.2.2 Datasets


Stata datasets have the **.dta** extension and can be loaded into Stata in the usual way through the **File** menu (for reading other data formats;

see [Section 1.4.1](#)). As in other statistical packages, a dataset is a matrix where the columns represent variables (with names and labels) and the rows represent observations. When a dataset is open, the variable names and variable labels appear in the Variables window. The dataset may be viewed as a spreadsheet by opening the **Data Browser** with the  button and edited by clicking  to open the **Data Editor**. Both the Data Browser and the Data Editor can also be opened through the **Window** menu. Note however, that nothing else can be done in Stata while the Data Browser or Data Editor are open (e.g. the Stata Command window disappears). See [Section 1.4](#) for more information on datasets.

1.2.3 *Commands and output*

Until release 8.0, Stata was entirely command-driven and many users still prefer using commands as follows: a command is typed in the Stata Command window and executed by pressing the *Return* (or *Enter*) key. The command then appears next to a full stop (period) in the Stata Results window, followed by the output.

If the output produced is longer than the Stata Results window, **--more--** appears at the bottom of the screen. Pressing any key scrolls the output forward one screen. The scroll-bar may be used to move up and down previously displayed output. However, only a certain amount of past output is retained in this window. For this reason and to save output for later, it is useful to open a log file; see [Section 1.2.6](#).

Stata is ready to accept a new command when the prompt (a period) appears at the bottom of the screen. If Stata is not ready to receive new commands because it is still running or has not yet displayed all the current output, it may be interrupted by holding down *Ctrl* and pressing the *Pause/Break* key or by pressing the red **Break** button .

A previous command can be accessed using the *PgUp* and *PgDn* keys or by selecting it from the Review window where all commands from the current Stata session are listed (see [Figure 1.1](#)). The command may then be edited if required before pressing *Return* to execute the command.

Most Stata commands refer to a list of variables, the basic syntax being *command varlist*. For example, if the dataset contains variables **x**, **y**, and **z**, then

```
list x y
```

lists the values of **x** and **y**. Other components may be added to the command; for example, adding *if exp* after *varlist* causes the com-



mand to process only those observations satisfying the logical expression *exp*. Options are separated from the main command by a comma. The complete command structure and its components are described in [Section 1.5](#).

1.2.4 *GUI versus commands*


Since release 8.0, Stata has a Graphical User Interface (GUI) that allows almost all commands to be accessed via point-and-click. Simply start by clicking into the **Data**, **Graphics**, or **Statistics** menus, make the relevant selections, fill in a dialog box, and click **OK**. Stata then behaves exactly as if the corresponding command had been typed with the command appearing in the Stata Results and Review windows and being accessible via *PgUp* and *PgDn*.

A great advantage of the menu system is that it is intuitive so that a complete novice to Stata could learn to run a linear regression in a few minutes. A disadvantage is that pointing and clicking can be time-consuming if a large number of analyses are required and cannot be automated. Commands, on the other hand, can be saved in a file (called a do-file in Stata) and run again at a later time. In our opinion, the menu system is a great device for finding out which command is needed and learning how it works, but serious statistical analysis is best undertaken using commands. In this book we therefore say very little about the menus and dialogs (they are largely self-explanatory after all), but see [Section 1.8](#) for an example of creating a graph through the dialogs.


1.2.5 *Do-files*


It is useful to build up a file containing the commands necessary to carry out a particular data analysis. This may be done using Stata's **Do-file Editor** or any other editor. The Do-file Editor may be opened by clicking  or by selecting **Do...** from the **File** menu. Commands can then be typed in and run as a batch either by clicking into  in the Do-file Editor or by using the command

```
do dofile
```

Alternatively, a subset of commands can be highlighted and executed by clicking into . The do-file can be saved for use in a future Stata session. See [Section 1.10](#) for more information on do-files.

1.2.6 Log files

It is useful to open a log file at the beginning of a Stata session. Press the button , type a filename into the dialog box, and choose **Save**. By default, this produces a SMCL (Stata Markup and Control Language, pronounced ‘smicle’) file with extension `.smcl`, but an ordinary ASCII text file can be produced by selecting the `.log` extension. If the file already exists, another dialog opens to allow you to decide whether to overwrite the file with new output or to append new output to the existing file.

The log file can be viewed in the Stata Viewer during the Stata session (again through ) and is automatically saved when it is closed. Log files can also be opened, viewed, and closed by selecting **Log** from the **File** menu, followed by **Begin...**, **View...**, or **Close**. The following commands can be used to open and close a log file `mylog`, replacing the old one if it already exists:

```
log using mylog, replace
log close
```

To view a log file produced in a previous Stata session, select **File** → **Log** → **View...** and specify the full path of the log file. The log may then be printed by selecting **Print Viewer...** from the **File** menu.

1.2.7 Getting help

Help may be obtained by clicking on **Help** which brings up the menu shown in [Figure 1.2](#). To get help on a Stata command, assuming the command name is known, select **Stata Command...** To find the appropriate Stata command first, select **Search...** which opens up the dialog in [Figure 1.3](#). For example, to find out how to fit a Cox regression, type ‘survival’ under *Keywords* and press **OK**. This opens the Stata Viewer containing a list of relevant command names or topics for which help files or Frequently Asked Questions (FAQs) are available. Each entry in this list includes a blue keyword (a *hyperlink*) that may be selected to view the appropriate help file or FAQ. Each help file contains hyperlinks to other relevant help files. The search and help files may also be accessed using the commands

```
search survival
help stcox
```

Help will then appear in the Stata Results window instead of the Stata Viewer, where words displayed in blue also represent hyperlinks to other



Figure 1.2: Menu for help.

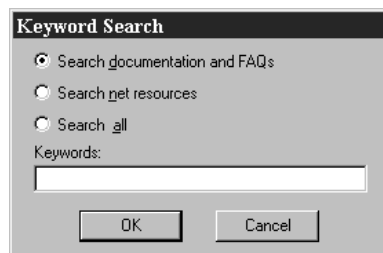



Figure 1.3: Dialog for search.

files.

If the computer running Stata is connected to the internet, you can also search through materials on the internet, to find for instance user-contributed programs by selecting ‘Search net resources’ in the search dialog. The final selection, ‘Search all’ performs a search across the help files, FAQs, and net materials. This is equivalent to using the `findit keyword` command. More refined searches can be carried out using the `search` command (see `help search`). The other selections in the help dialog, **News**, **Official Updates**, **SJ and User-written Programs**, and **Stata Web Site** all enable access to relevant information on the Web (see [Section 1.11](#) on keeping Stata up-to-date).

1.2.8 Closing Stata

Stata can be closed in three ways:

- click on the **Close** button  at the top right-hand corner of the Stata screen
- select **Exit** from the **File** menu
- type `exit`, clear in the Stata Commands window, and press *Return*.

1.3 Conventions used in this book

In this book we will use typewriter font `like this` for anything that could be typed into the Stata Command window or a do-file, that is, command names, options, variable names, etc. In contrast, italicized words are not supposed to be typed; they should be substituted by another word. For example, `summarize varname` means that *varname* should be substituted by a specific variable name, such as `age`, giving `summarize age`. We will usually display sequences of commands as follows:

```
summarize age
drop age
```

If a command continues over two lines, we use `/*` at the end of the first line and `*/` at the beginning of the second line to make Stata ignore the linebreak. An alternative would be to use `///` at the end of the line. Note that these methods are for use in a do-file and do not work in the Stata Command window where they would result in an error. In the Stata Command window, commands can wrap over several lines.

Output taking very little space is displayed immediately following the commands but without indentation and in a smaller font:

```
display 1
```

1

Output taking up more space is shown in a numbered display floating in the text. Some commands produce little notes, for example, the **generate** command prints out how many missing values are generated. We will usually not show such notes.

1.4 Datasets in Stata

1.4.1 *Data input and output*

Stata has its own data format with default extension **.dta**. Reading and saving a Stata file are straightforward. If the filename is **bank.dta**, the commands are

```
use bank
save bank
```

If the data are not stored in the current directory, then the complete path must be specified, as in the command

```
use c:\user\data\bank
```

However, the least error-prone way of keeping all the files for a particular project in one directory is to change to that directory and save and read all files without their pathname:

```
cd c:\user\data
use bank
save bank
```

Data supplied with Stata can be read in using the **sysuse** command. For instance, the famous **auto.dta** data can be read using

```
sysuse auto
```

Before reading a file into Stata, all data already in memory need to be cleared, either by running **clear** before the **use** command or by using the option **clear** as follows:

```
use bank, clear
```

If we wish to save data under an existing filename, this results in an error message unless we use the option **replace** as follows:

```
save bank, replace
```

For large datasets it is sometimes necessary to increase the amount of memory Stata allocates to its data areas from the default of 1 megabyte. For example, when no dataset is loaded (e.g., after issuing the command **clear**), set the memory to 2 megabytes using

```
set memory 2m
```

The **memory** command without arguments gives information on how much memory is being used and how much is available.

If the data are not available in Stata format, they may be converted to Stata format using another package (e.g., Stat/Transfer) or saved as an ASCII file (although the latter option means losing all the labels). When saving data as ASCII, missing values should be replaced by some numerical code.

There are three commands available for reading different types of ASCII data: **insheet** is for files containing one observation (on all variables) per line with variables separated by tabs or commas, where the first line may contain the variable names; **infile** with *varlist* (free format) allows line breaks to occur anywhere and variables to be separated by spaces as well as commas or tabs; **infix** is for files with fixed column format but a single observation can go over several lines; **infile** with a dictionary (fixed format) is the most flexible command since the dictionary can specify exactly what lines and columns contain what information.

Data can be saved as ASCII using **outfile** or **outsheet**. Finally, **odbc** can be used to load, write, or view data from Open Data Base Connectivity (ODBC) sources. See **help infiling** or [U] **24 Commands to input data** for an overview of commands for reading data.

Only one dataset may be loaded at any given time but a dataset may be combined with the currently loaded dataset using the command **merge** or **append** to add observations or variables; see also [Section 1.6.2](#).

1.4.2 Variables

There are essentially two kinds of variables in Stata: *string* and *numeric*. Each variable can be one of a number of storage types that

require different numbers of bytes. The storage types are `byte`, `int`, `long`, `float`, and `double` for numeric variables and `str1` to `str80` (`str244` in Stata/SE) for string variables of different lengths. Besides the storage type, variables have associated with them a name, a label, and a format. The name of a variable `y` can be changed to `x` using

```
rename y x
```

The variable label can be defined using

```
label variable x "cost in pounds"
```

and the format of a numeric variable can be set to ‘general numeric’ with two decimal places using

```
format x %7.2g
```

Numeric variables

A missing values in a numeric variable is represented by a period ‘.’ (system missing values), or by a period followed by a letter, such as `.a`, `.b`, etc. Missing values are interpreted as very large positive numbers with `.` < `.a` < `.b`, etc. Note that this can lead to mistakes in logical expressions; see also [Section 1.5.2](#). Numerical missing value codes (such as ‘–99’) may be converted to missing values (and vice versa) using the command `mvdecode`. For example,

```
mvdecode x, mv(–99)
```

replaces all values of variable `x` equal to –99 by periods and

```
mvencode x, mv(–99)
```

changes the missing values back to –99.

Numeric variables can be used to represent categorical or continuous variables including dates. For categorical variables it is not always easy to remember which numerical code represents which category. Value labels can therefore be defined as follows:

```
label define s 1 married 2 divorced 3 widowed 4 single  
label values marital s
```

The categories can also be recoded, for example

```
recode marital 2/3=2 4=3
```

merges categories 2 and 3 into category 2 and changes category 4 to 3.

Dates are defined as the number of days since 1/1/1960 and can be displayed using a date format such as %d. For example, listing the variable `time` in %7.0g format gives

```
list time
```

	time
1.	14976
2.	200

which is not as easy to interpret as

```
format time %d  
list time
```

	time
1.	01jan2001
2.	19jul1960

See `help dfmt` for other date formats.

String variables

String variables are typically used for categorical variables or identifiers and in some cases for dates (e.g., if the file was saved as an ASCII file from SPSS). In Stata, it is generally advisable to represent these variables by numeric variables, and conversion from string to numeric is straightforward. A categorical string variable (or identifier) can be converted to a numeric variable using the command `encode` which replaces each unique string by an integer and uses that string as the label for the corresponding integer value. The command `decode` converts the labeled numeric variable back to a string variable.

A string variable *string1* representing dates can be converted to numeric using the function `date(string1, string2)` where *string2* is a permutation of "dmy" to specify the order of the day, month, and year in *string1*. For example, the commands

```
display date("30/1/1930","dmy")
```

and

```
display date("january 30, 1930", "mdy")
```

both return the negative value -10928 because the date is 10928 days before 1/1/1960.

1.5 Stata commands

Typing `help language` gives the following generic command structure for most Stata commands:

```
[by varlist:] command [varlist] [= exp] [if exp] [in range]  
[weight] [using filename] [, options]
```

The help file contains links to information on each of the components, and we will briefly describe them here:

`[by varlist:]` instructs Stata to repeat the command for each combination of values in the list of variables *varlist*.

command is the name of the command and can often be abbreviated; for example, the command `display` can be abbreviated as `dis`.

`[varlist]` is the list of variables to which the command applies.

`[=exp]` is an expression.

`[if exp]` restricts the command to that subset of the observations that satisfies the logical expression *exp*.

`[in range]` restricts the command to those observations whose indices lie in a particular range *range*.

`[weight]` allows weights to be associated with observations (see [Section 1.7](#)).

`[using filename]` specifies the filename to be used.

`[, options]` a comma is only needed if *options* are used; options are specific to the command and can often be abbreviated.

For any given command, some of these components may not be available; for example, `list` does not allow `[using filename]`. The

help files for specific commands specify which components are available, using the same notation as above, with square brackets enclosing components that are optional. For example, `help log` gives

```
log using filename [, noproc append replace [text|smcl] ]
```

implying that `[by varlist:]` is not allowed and that `using filename` is required, whereas the three options `noproc`, `append`, `replace` and `[text|smcl]` (meaning `text` or `smcl`) are optional.

The syntax for *varlist*, *exp*, and *range* is described in the next three subsections, followed by information on how to loop through sets of variables or observations.

1.5.1 *Varlist*

The simplest form of *varlist* is a list of variable names separated by spaces. Variable names may also be abbreviated as long as this is unambiguous, e.g., `x1` may be referred to by `x` only if there is no other variable name starting with `x` such as `x` itself or `x2`. A set of adjacent variables such as `m1`, `m2`, and `x` may be referred to as `m1-x`. All variables starting with the same set of letters can be represented by that set of letters followed by a wild card `*`, so that `m*` may stand for `m1 m6 mother`. The set of all variables is referred to by `_all` or `*`. Examples of a *varlist* are

```
x y
x1-x16
a1-a3 my* sex age
```

1.5.2 *Expressions*

There are logical, algebraic, and string expressions in Stata. Logical expressions evaluate to 1 (true) or 0 (false) and use the operators `<` and `<=` for ‘less than’ and ‘less than or equal to’, respectively. Similarly, `>` and `>=` are used for ‘greater than’ and ‘greater than or equal to’. The symbols `==` and `~=` stand for ‘equal to’ and ‘not equal to’, and the characters `!`, `&` and `|` represent ‘not’, ‘and’, and ‘or’, respectively, so that

```
if (y!=2 & z>x)|x==1
```

means ‘if `y` is not equal to 2 and `z` is greater than `x` or if `x` equals 1’. In fact, expressions involving variables are evaluated for each observation

so that the expression really means

$$(y_i \neq 2 \ \& \ z_i > x_i) \mid x_i == 1$$

where i is the observation index.

Great care must be taken in using the `>` or `>=` operators when there are missing data. For example, if we wish to delete all subjects older than 16, the command

```
drop if age>16
```

will also delete all subjects for whom `age` is missing since a missing value (represented by `'.'`, `'.a'`, `'.b'`, etc.) is interpreted as a very large number. It is always safer to accommodate missing values explicitly using for instance

```
drop if age>16 & age<.
```

Note that this is safer than specifying `age!=.` since this would not exclude missing values coded as `'.a'`, `'.b'`, etc.

Algebraic expressions use the usual operators `+`, `-`, `*`, `/`, and `^` for addition, subtraction, multiplication, division, and powering, respectively. Stata also has many mathematical functions such as `sqrt()`, `exp()`, `log()`, etc. and statistical functions such as `chiprob()` and `normprob()` for cumulative distribution functions and `invnorm()`, etc., for inverse cumulative distribution functions. Pseudo-random numbers with a uniform distribution on the $[0,1)$ interval may be generated using `uniform()`. Examples of algebraic expressions are

```
y + x  
(y + x)^ 3 + a/b  
invnorm(uniform()))+2
```

where `invnorm(uniform())` returns a (different) draw from the standard normal distribution for each observation.

Finally, string expressions mainly use special string functions such as `substr(str,n1,n2)` to extract a substring from `str` starting at `n1` for a length of `n2`. The logical operators `==` and `~=` are also allowed with string variables and the operator `+` concatenates two strings. For example, the combined logical and string expression

```
"moon"+substr("sunlight",4,5))=="moonlight"
```

returns the value 1 for ‘true’.

For a list and explanation of all functions, use `help functions`.

1.5.3 *Observation indices and ranges*

Each observation has an index associated with it. For example, the value of the third observation on a particular variable `x` may be referred to as `x[3]`. The macro `_n` takes on the value of the running index and `_N` is equal to the number of observations. We can therefore refer to the previous observation of a variable as `x[_n-1]`.

An indexed variable is only allowed on the right-hand side of an assignment. If we wish to replace `x[3]` by 2, we can do this using the syntax

```
replace x = 2 if _n==3
```

We can refer to a range of observations either using `if` with a logical expression involving `_n` or, more easily, by using `in range`. The command above can then be replaced by

```
replace x = 2 in 3
```

More generally, *range* can be a range of indices specified using the syntax `f/l` (for ‘first to last’) where `f` and/or `l` may be replaced by numerical values if required, so that `5/12` means ‘fifth to twelfth’ and `f/10` means ‘first to tenth’, etc. Negative numbers are used to count from the end, for example

```
list x in -10/1
```

lists the last 10 observations.

1.5.4 *Looping through variables or observations*

Explicitly looping through observations is often not necessary because expressions involving variables are automatically evaluated for each observation. It may however be required to repeat a command for subsets of observations and this is what `by varlist:` is for. Before using `by varlist:`, however, the data must be sorted using

```
sort varlist
```

where *varlist* includes the variables to be used for **by** *varlist*:. Note that if *varlist* contains more than one variable, ties in the earlier variables are sorted according to the next variable(s). For example,

```
sort school class
by school class: summarize test
```

give the summary statistics of **test** for each class. If **class** is labeled from 1 to n_i for the i th school, then not using **school** in the above commands would result in the observations for all classes with the same label being grouped together. To avoid having to sort the data, **bysort** can be substituted for **by** so that the following single command replaces the two commands above:

```
bysort school class: summarize test
```

A very useful feature of **by** *varlist*: is that it causes the observation index **_n** to count from 1 within each of the groups defined by the distinct combinations of the values of *varlist*. The macro **_N** represents the number of observations in each group. For example,

```
sort group age
by group: list age if _n==_N
```

lists **age** for the last observation in each group where the last observation in this case is the observation with the highest **age** within its group. The same can be achieved in a single **bysort** command:

```
bysort group (age): list age if _n==_N
```

where the variable in parentheses is used to sort the data but does not contribute to the definition of the subgroups of observations to which the **list** command applies.

We can also loop through a list of variables or other objects using **foreach**. The simplest syntax is

```
foreach variable in v1 v2 v3 {
    list `variable'
}
```

This syntax uses a *local macro* (see also [Section 1.9](#)) **variable** which takes on the (string) values **v1**, then **v2**, and finally **v3** inside the braces. (Local macros can also be defined explicitly using **local variable v1**).

Enclosing the local macro name in `` `` is equivalent to typing its contents, i.e., ``variable`` evaluates to `v1`, then `v2`, and finally `v3` so that each of these variables is listed in turn.

In the first line above we listed each variable explicitly. We can instead use the more general *varlist* syntax by specifying that the list is of type *varlist* as follows:

```
foreach variable of varlist v* {  
    list `variable`  
}
```

Numeric lists can also be specified. The command

```
foreach number of numlist 1 2 3 {  
    disp `number`  
}
```

produces the output

```
1  
2  
3
```

Numeric lists may be abbreviated by ‘first/last’, here `1/3` or ‘first(increment)last’, for instance `1(2)7` for the list `1 3 5 7`. See `help foreach` for other list types.

For numeric lists, a simpler syntax is `forvalues`. To produce the output above, use

```
forvalues i=1/3 {  
    disp `i`  
}
```

The same output can also be produced using `while` as follows:

```
local i = 1  
while i<=3 {  
    disp `i`  
    local i = `i` + 1  
}
```

Here the local macro `i` was defined using `local i = 1` and then incremented by 1 using `local i = `i` + 1`. See also [Section 1.10](#) on programming. Cox (2002b) gives a useful tutorial on `byvarlist`: and Cox (2002a; 2003) discusses `foreach` and `forvalues` in detail.

1.6 Data management

1.6.1 *Generating and changing variables*

New variables may be generated using the commands **generate** or **egen**. The command **generate** simply equates a new variable to an expression which is evaluated for each observation. For example,

```
generate x = 1
```

creates a new variable called **x** and sets it equal to one. When **generate** is used together with **if** *exp* or **in** *range*, the remaining observations are set to missing. For example,

```
generate percent = 100*(old - new)/old if old>0
```

generates the variable **percent** and sets it equal to the percentage decrease from **old** to **new** where **old** is positive and equal to missing otherwise. The command **replace** works in the same way as **generate** except that it allows an existing variable to be changed. For example,

```
replace percent = 0 if old<=0
```

changes the missing values in the variable **percent** to zeros. The two commands above could be replaced by the single command

```
generate percent = cond(old>0, 100*(old-new)/old, 0)
```

where **cond()** evaluates to the second argument if the first argument is true and to the third argument otherwise.

The command **egen** provides extensions to **generate**. One advantage of **egen** is that some of its functions accept a variable list as an argument, whereas the functions for **generate** can only take simple expressions as arguments. For example, we can form the average of 100 variables **m1** to **m100** using

```
egen average = rmean(m1-m100)
```

where missing values are ignored. Other functions for **egen** operate on groups of observations. For example, if we have the income (variable **income**) for members within families (variable **family**), we may want to compute the total income of each member's family using

```
egen faminc = sum(income), by(family)
```

```

Number of obs.          2  ->    4
Number of variables      3  ->    3
j variable (2 values)    ->   occ
xij variables:
                        x1 x2  ->   x

```

The data now look like this:

```
list
```

	subj	occ	x
1.	1	1	2
2.	1	2	3
3.	2	1	4
4.	2	2	5

We can change the data back again using

```
reshape wide x, i(subj) j(occ)
```

For data in the long shape, it may be required to collapse the data so that each group is represented by a single summary measure. For example, for the data above, each subject's responses can be summarized using the mean, **meanx**, and standard deviation, **sdx**, and the number of nonmissing responses, **num**. This can be achieved using

```
collapse (mean) meanx=x (sd) sdx=x (count) num=x, by(subj)
list
```

	subj	meanx	sdx	num
1.	1	2.5	.707107	2
2.	2	4.5	.707107	2

Since it is not possible to convert back to the original format in this case, the data may be preserved before running **collapse** and restored again later using the commands **preserve** and **restore**.

Other ways of changing the shape of data include dropping observations using

```
drop in 1/10
```

to drop the first 10 observations or

```
bysort group (weight): keep if _n==1
```

to drop all but the lightest member of each group. Sometimes it may be necessary to transpose the data, converting variables to observations and vice versa. This may be done and undone using `xpose`.

If each observation represents a number of units (as after `collapse`), it may sometimes be required to replicate each observation by the number of units, `num`, that it represents. This may be done using

```
expand num
```

If there are two datasets, `subj.dta`, containing subject specific variables, and `occ.dta`, containing occasion-specific variables for the same subjects, then if both files contain the same sorted subject identifier `subj_id` and `subj.dta` is currently loaded, the files may be merged as follows:

```
merge subj_id using occ
```

resulting in the variables from `subj.dta` being expanded as in the `expand` command above and the variables from `occ.dta` being added.

1.7 Estimation

All estimation commands in Stata, for example `regress`, `logistic`, `poisson`, and `glm`, follow the same syntax and share many of the same options.

The estimation commands also produce essentially the same output and save the same kind of information. The stored information may be processed using the same set of *post-estimation commands*.

The basic command structure is

```
[xi:] command depvar [model] [weights], options
```

which may be combined with `by varlist:`, `if exp`, and `in range` as usual. The response variable is specified by `depvar` and the explanatory variables by `model`. The latter is usually just a list of explanatory variables. If categorical explanatory variables and interactions are required, using `xi:` at the beginning of the command enables special notation for `model` to be used. For example,


```
xi: regress resp i.x
```

creates dummy variables for each value of **x** except the lowest value and includes these dummy variables as predictors in the model.

```
xi: regress resp i.x*y z
```

fits a regression model with the main effects of **x**, **y**, and **z** and their interaction **x*y** where **x** is treated as categorical and **y** and **z** as continuous (see `help xi` for further details).

The syntax for the `[weights]` option is

```
weighttype = varname
```

where *weighttype* depends on the reason for weighting the data. If the data are in the form of a table where each observation represents a group containing a total of **freq** observations, using `[fweight=freq]` is equivalent to running the same estimation command on the expanded dataset where each observation has been replicated **freq** times. If the observations have different standard deviations, for example, because they represent averages of different numbers of observations, then **aweight**s is used with weights proportional to the reciprocals of the standard deviations. Finally, **pweight**s is used for probability weighting where the weights are equal to the inverse probability that each observation was sampled. (Another type of weights, **iweight**s, is available for some estimation commands, mainly for use by programmers.)

All the results of an estimation command are stored and can be processed using post-estimation commands. For example, **predict** may be used to compute predicted values or different types of residuals for the observations in the present dataset and the commands **test**, **testparm**, **lrtest** and **lincom** for inferences based on previously estimated models.

The saved results can also be accessed directly using the appropriate names. For example, the regression coefficients are stored in global macros called `_b[varname]`. To display the regression coefficient of **x**, simply type

```
display _b[x]
```

To access the entire parameter vector, use `e(b)`. Many other results may be accessed using the `e(name)` syntax. See the ‘Saved Results’ section of the entry for the estimation command in the *Stata Reference Manuals* to find out under what names particular results are stored. The command

```
ereturn list
```

lists the names and contents of all results accessible via `e(name)`.

Note that ‘r-class’ results produced by commands that are not estimation commands can be accessed using `r(name)`. For example, after `summarize`, the mean can be accessed using `r(mean)`. The command

```
return list
```

list the names and contents of all ‘r-class’ results currently available.

1.8 Graphics

One of the great new features introduced in Stata 8 is the improved graphics. The graphical user interface (GUI) makes it extremely easy to produce a very attractive graph with different line-styles, legends, etc. To demonstrate this, we first simulate some data as follows:

```
clear
set obs 100
set seed 13211
gen x=invnorm(uniform())
gen y = 2 + 3*x + invnorm(uniform())
```

To produce a scatterplot of y versus x via the GUI, select **Twoway graph (scatterplot, line etc.)** from the **Graphics** menu to bring up the dialog box shown in [Figure 1.4](#). Specify x and y in the boxes labeled x and y . This can be done either by typing or by first clicking into the box and then selecting the appropriate variable from the Variables window. To add a label to the x -axis, click into the tab labeled **X-Axis** and type ‘Simulated x ’ in the **Title** box. Similarly, type ‘Simulated y ’ in the **Title** box in the **Y-Axis** tab. Finally, click **OK** to produce the graph shown in [Figure 1.5](#). To change for instance the symbol we would have to plot the graph again, this time selecting a different option in the box labeled **Symbol** under the heading **Marker** in the dialog box (it is not possible to edit a graph). The following command appears in the output:

```
twoway (scatter y x), ytitle(Simulated y) xtitle(Simulated x)
```

The command `twoway`, short for `graph twoway`, can be used to plot scatterplots, lines or curves and many other plots requiring an x and

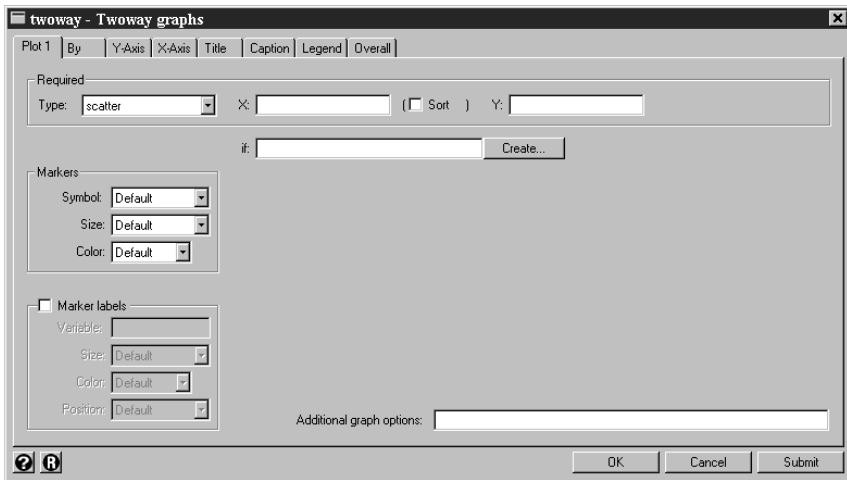


Figure 1.4: Dialog box for twoway graph.

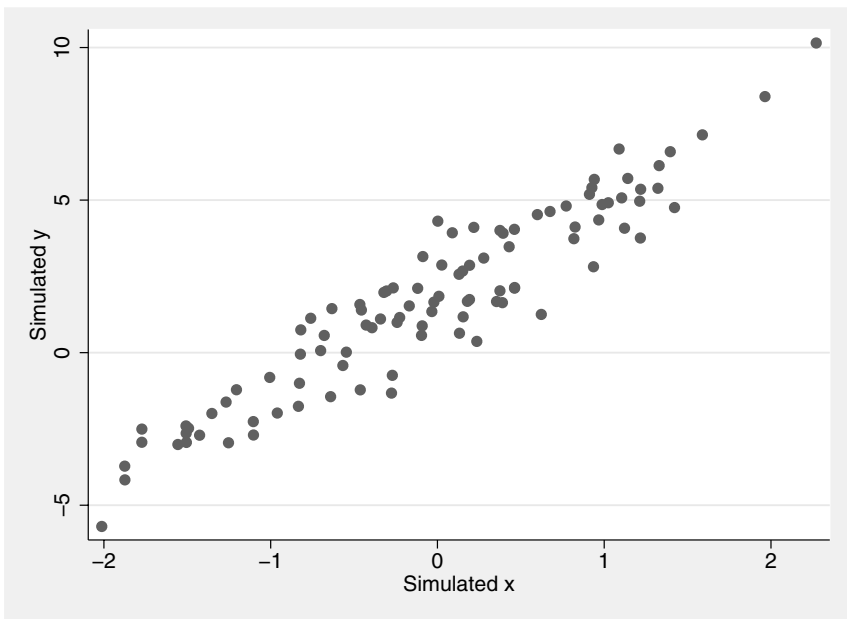


Figure 1.5: Scatterplot of simulated data.

y-axis. Here the *plotype* is **scatter** which requires a *y* and *x* variable to be specified. Details such as axis labels are given after the comma. Help on scatterplots can be found (either in the manual or using help) under ‘graph twoway scatter’. Help on options for **graph twoway** can be found under ‘twoway-options’.

We can use a single **graph twoway** to produce a scatterplot with a regression line superimposed:

```
twoway (scatter y x) (lfit y x),          /*
    */ ytitle(Simulated y) xtitle(Simulated x) /*
    */ legend(order(1 "Observed" 2 "Fitted"))
```

giving the graph in Figure 1.6. Inside each pair of parentheses is a

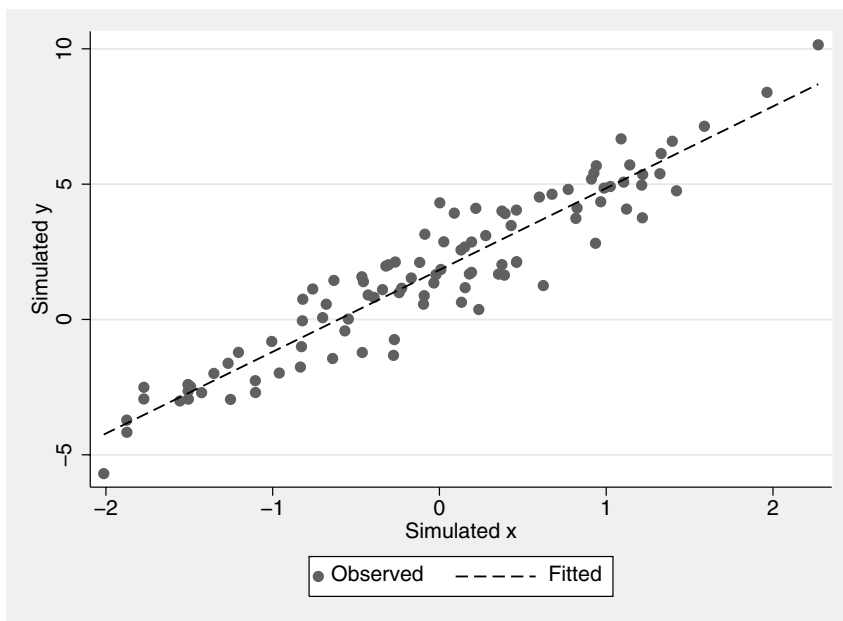


Figure 1.6: Scatterplot and fitted regression line.

command specifying a plot to be added to the same graph. The options applying to the graph as a whole appear after these individual plots preceded by a comma as usual. Here the **legend()** option was used to specify labels for the legend; see the manual or help for ‘legend-option’.

Each plot can have its own *if exp* or *in range* restrictions as well as various options. For instance, we first create a new variable **group**,

taking on the values 1 and 2 and add 2 to the y -values of group 2:

```
gen group = cond(_n < 50,1,2)
replace y = y+2 if group==2
```

Now produce a scatterplot with different symbols for the two groups and separate regression lines using

```
twoway (scatter y x if group==1, msymbol(0))      /*
      */ (lfit y x if group==1, clpat(solid))      /*
      */ (scatter y x if group==2, msymbol(Oh))   /*
      */ (lfit y x if group==2, clpat(dash)),      /*
      */ ytitle(Simulated y) xtitle(Simulated x) /*
      */ legend(order(1 2 "Group 1" 3 4 "Group 2"))
```

giving the graph shown in Figure 1.7. The `msymbol(0)` and `msymbol(Oh)`

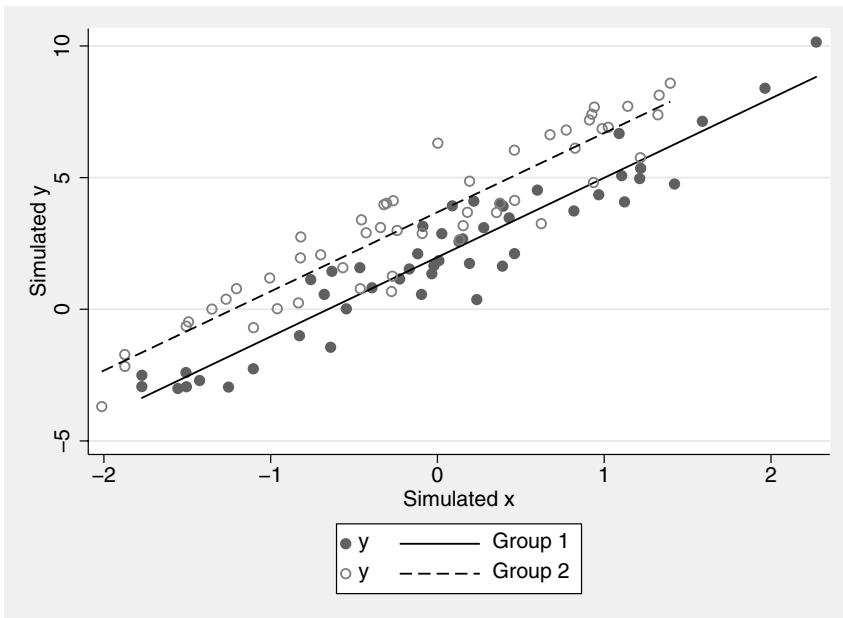


Figure 1.7: Scatterplot and fitted regression line.

options produce solid and hollow circles, respectively, whereas `clpat(solid)` and `clpat(dash)` produce solid and dashed lines, respectively. These options are inside the parentheses for the corresponding plots. The options referring to the graph as a whole, `xtitle()`, `ytitle()`, and

`legend()`, appear after the individual plots have been specified. Just before the final comma, we could also specify *if exp* or *in range* restrictions for the graph as a whole.

Some people find it more convenient to separate plots using `||` instead of enclosing them in parentheses, for instance replacing the first two lines above by

```
twoway scatter y x if group==1, ms(0) || /*  
    */ lfit y x if group==1, clpat(solid)
```

The `by()` option can be used to produce separate plots (each with their own sets of axes) in the same graph. For instance

```
label define gr 1 "Group 1" 2 "Group 2"  
label values group gr  
twoway scatter y x, by(group)
```

produces the graph in Figure 1.8. Here the value labels of `group` are used to label the individual panels.

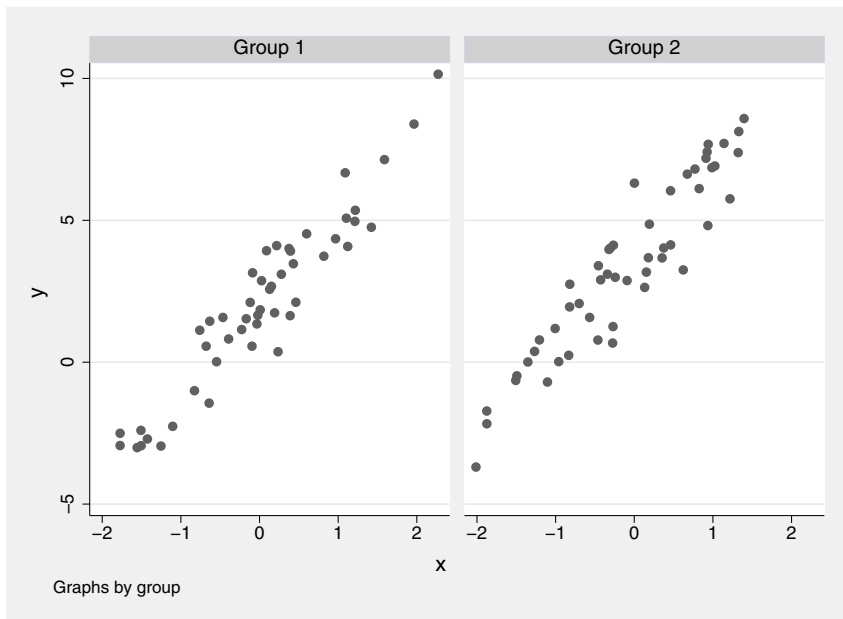


Figure 1.8: Separate scatterplot produced using `by()`.

Other useful graphics commands include `graph twoway` function for plotting a function without having to define any new variables, `graph matrix` for scatterplot matrices, `graph box` for boxplots, `graph bar` for bar charts, `histogram` for histograms, `kdensity` for kernel density plots and `qnorm` for Q-Q plots.

For `graph box` and `graph bar`, we may wish to plot different variables, referred to as *yvars* in Stata, for different subgroups or categories, of individuals, specified using the `over()` option. For example,

```
replace x = x + 1
graph bar y x, over(group)
```

results in the bar chart in Figure 1.9. See `yvar options` and `group`

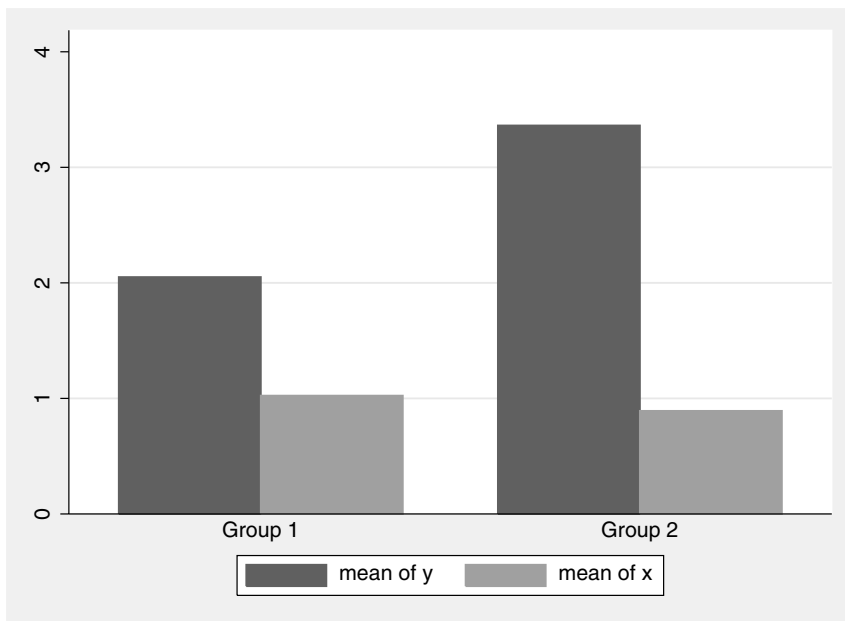


Figure 1.9: Bar chart.

`options` in [G] `graph bar` for ways to change the labeling and presentation of the bars.

The general appearance of graphs is defined in `schemes`. In this book we use scheme `sj` (Stata Journal) by issuing the command

```
set scheme sj
```

at the beginning of each Stata session. See [G] **schemes** or **help schemes** for a complete list and description of schemes available.

We find the GUI interface particularly useful for learning about these and other graphics commands and their options.

1.9 Stata as a calculator

Stata can be used as a simple calculator using the command **display** followed by an expression, e.g.,

```
display sqrt(5*(11-3^2))
```

```
3.1622777
```

There are also a number of statistical commands that can be used without reference to any variables. These commands end in **i**, where **i** stands for *immediate command*. For example, we can calculate the sample size required for an independent samples *t*-test to achieve 80% power to detect a difference at the 1% level (2-sided) if the population means are 1 and 2 and the within population standard deviation is 1 using **sampsi** as follows:

```
sampsi 1 2, sd(1) power(.8) alpha(0.01)
```

(see Display 1.1). Similarly, **ttesti** can be used to carry out a *t*-test

Estimated sample size for two-sample comparison of means

Test Ho: $m_1 = m_2$, where m_1 is the mean in population 1
and m_2 is the mean in population 2

Assumptions:

```
alpha =    0.0100  (two-sided)
power =    0.8000
  m1 =         1
  m2 =         2
  sd1 =         1
  sd2 =         1
n2/n1 =     1.00
```

Estimated required sample sizes:

```
n1 =        24
n2 =        24
```

Display 1.1

if the means, standard deviations, and sample sizes are given.

As briefly shown in [Section 1.5.4](#), results can be saved in local macros using the syntax

```
local a = exp
```

The result may then be used again by enclosing the local macro name in single quotes ``'` (using two different keys on the keyboard). For example,

```
local a = 5
display sqrt(`a`)
```

2.236068

Matrices can also be defined and matrix algebra carried out interactively. The following `matrix` commands define a matrix `a`, display it, and give its trace and its eigenvalues:

```
matrix a = (1,2\ 2,4)
matrix list a
```

```
symmetric a[2,2]
   c1  c2
r1   1
r2   2   4
```

```
dis trace(a)
```

5

```
matrix symeigen x v = a
matrix list v
```



```
v[1,2]
   e1  e2
r1   5   0
```

1.10 Brief introduction to programming

So far we have described commands as if they would be run interactively. However, in practice, it is always useful to be able to repeat the entire analysis using a single command. This is important, for example, when a data entry error is detected after most of the analysis has already been carried out. In Stata, a set of commands stored as a do-file, called for example, **analysis.do**, can be executed using the command

```
do analysis
```

We strongly recommend that readers create do-files for any work in Stata, e.g., for the exercises of this book.

One way of generating a do-file is to carry out the analysis interactively and save the commands, for example, by clicking into  in the Review window and selecting **Save Review Contents**. Stata's **Do-file Editor** can also be used to create or edit a do-file. One way of trying out commands interactively and building up a do-file is to run commands in the Commands window and copy them into the **Do-file Editor** after checking that they work. Another possibility is to type commands into the **Do-file Editor** and try them out individually by highlighting the commands and clicking into  or selecting **Tools** → **Do Selection**. Alternatively, any text editor may be used to create a do-file. The following is a useful template for a do-file:

```
/* comment describing what the file does */
version 8.0
capture log close
log using filename, replace
set more off

command 1
command 2
etc.

log close
exit
```

We will explain each line in turn.

1. The ‘brackets’ **/*** and ***/** cause Stata to ignore everything between them. Another way of *commenting out lines* of text is to start the lines with a simple *****.

2. The command `version 8.0` causes Stata to interpret all commands as if we were running Stata version 8.0 even if, in the future, we have actually installed a later version in which some of these commands do not work any more.
3. The `capture` prefix causes the do-file to continue running even if the command results in an error. The `capture log close` command therefore closes the current log file if one is open or returns an error message. (Another useful prefix is `quietly` which suppresses all output, except error messages.)
4. The command `log using filename, replace` opens a log file, replacing any file of the same name if it already exists.
5. The command `set more off` causes all the output to scroll past automatically instead of waiting for the user to scroll through it manually. This is useful if the user intends to look at the log file for the output.
6. After the analysis is complete, the log file is closed using `log close`.
7. The last statement, `exit`, is not necessary at the end of a do-file but may be used to make Stata stop running the do-file wherever it is placed.

Variables, global macros, local macros, and matrices can be used for storing and referring to data and these are used extensively in programs. For example, we may wish to subtract the mean of `x` from `x`. Interactively, we would use

```
summarize x
```

to find out what the mean is and then subtract that value from `x`. However, we should not type the value of the mean into a do-file because the result would no longer be valid if the data change. Instead, we can access the mean computed by `summarize` using `r(mean)`:

```
quietly summarize x, meanonly
gen xnew = x-r(mean)
```

(If all that is required from `summarize` is the mean or the sum, it is more efficient to use the `meanonly` option.) Most Stata commands are *r class*, meaning that they store results that may be accessed using `r()` with the appropriate name inside the brackets. Estimation commands store the results in `e()`. To find out under what names constants are stored, see the ‘Stored Results’ section for the command of interest in the *Stata Reference Manuals*.

If a local macro is defined without using the `=` sign, anything can appear on the right-hand side and typing the local macro name in single

quotes has the same effect as typing whatever appeared on the right-hand side in the definition of the macro. For example, if we have a variable `y`, we can use the commands

```
local a y
disp "`a'[1] = " `a'[1]
```

```
y[1] = 4.6169958
```

Local macros are only ‘visible’ within the do-file or program in which they are defined. Global macros may be defined using

```
global a = 1
```

and accessed by prefixing them with a dollar sign, for example,

```
gen b = $a
```

Sometimes it is useful to have a general set of commands (or a program) that may be applied in different situations. It is then essential that variable names and parameters specific to the application can be passed to the program. If the commands are stored in a do-file, the ‘arguments’ with which the do-file will be used are referred to as ``1’`, ``2’` etc. inside the do-file. For example, a do-file *filename.do* containing the command

```
list `1' `2'
```

may be run using

```
do filename x1 x2
```

to cause `x1` and `x2` to be listed. Alternatively, we can define a program which can be called without using the `do` command in much the same way as Stata’s own commands. This is done by enclosing the set of commands by

```
program proname
end
```

After running the program definition, we can run the program by typing the program name and arguments.

Most programs require things to be done repeatedly by looping through some list of objects. This can be achieved using `foreach` and

forvalues. For example, we define a program called `mylist` that lists the first three observations of each variable in a variable list:

```
program mylist
version 8.1
syntax varlist
foreach var in `varlist' { /* outer loop: variables */
    display "`var'"
    forvalues i=1/3 { /* inner loop: observations */
        display `var'[`i']
    }
    display " "
}
end
```

We can run the program using the command

```
mylist x y z
```

Here the `syntax` command defines the syntax to be

```
mylist varlist
```

(no options allowed), issues an error message if *varlist* is not valid, for example if one of the variables does not exist, and places the variable names into the local macro `'varlist'` (see `help syntax` and `[P] syntax`). The outer `foreach` loop repeats everything within the outer braces for each variable in `'varlist'`. Within this loop, the 'current' variable is placed in the local macro `var`. For each variable, the inner `forvalues` loop repeats the `display` command for `i` equal to 1, 2, and 3.

A program may be defined by typing it into the Commands window. This is almost never done in practice. However, a more useful method is to define the program within a do-file where it can easily be edited. Note that once the program has been loaded into memory (by running the `program` command), it has to be cleared from memory using `program drop` before it can be redefined. It is therefore useful to have the command

```
capture program drop mylist
```

in the do-file before the `program` command, where `capture` ensures that the do-file continues running even if `mylist` does not yet exist.

A program may also be saved in a separate file (containing only the program definition) of the same name as the program itself and having

the extension `.ado`. If the ado-file (automatic do-file) is in a directory in which Stata looks for ado-files, for example the current directory, it can be executed simply by typing the name of the file. There is no need to load the program first (by running the program definition). To find out where Stata looks for ado-files, type

```
adopath
```

This lists various directories including `\ado\personal\`, the directory where personal ado-files may be stored. Many of Stata's own commands are actually ado-files stored in the `ado` subdirectory of the directory where the Stata executable (e.g., `wstata.exe`) is located.

The [P] *Programming Reference Manual* gives detailed information on the programming commands mentioned here and many more. See also [P] **dialogs** for information on programming your own dialogs. Not discussed in the manuals for Stata release 8.0 is the Stata Plugin Interface (SPI) which allows compiled C-programs to be called from a Stata program. Information on how this is done can be found at <http://www.stata.com/support/plugins/>.

Chapter 13 of this book gives some examples of maximizing your own likelihood, and this is discussed in detail in Gould *et al.* (2003).

1.11 Keeping Stata up to date

Stata Corporation continually updates the current version of Stata. If the computer is connected to the Internet, Stata can be updated by issuing the command

```
update all
```

Ado-files are then downloaded and stored in the correct directory. If the executable has changed since the last update, a new executable (e.g., `wstata.bin`) is also downloaded. This file should be used to overwrite the old executable (e.g., `wstata.exe`) after saving the latter under a new name (e.g., `wstata.old`). A quick and easy way of achieving this is to issue the command `update swap` within Stata. The command `help whatsnew` lists all the changes since the release of the present version of Stata.

In addition to Stata's official updates to the package, users are continuously creating and updating their own commands and making them available to the Stata community. Articles on user-written programs are published in a peer-reviewed journal called *The Stata Journal* (SJ)

which replaced the *Stata Technical Bulletin* (STB) at the end of 2001. These and other user-written programs can be downloaded by clicking into **Help** → **SJ & User-written Programs**, and selecting one of a number of sites including sites for the SJ and STB. A large repository for user-written Stata programs is the *Statistical Software Components* (SSC) archive at <http://ideas.repec.org/s/boc/bocode.html> maintained by Kit Baum (the archive is part of IDEAS which uses the RePEc database). These programs can be downloaded using the `ssc` command. To find out about commands for a particular problem (user-written or part of Stata), use the `findit` command. For example, running

```
findit meta
```

brings up the **Stata Viewer** with a long list of entries including one on STB-42:

```
STB-42  sbe16.1 . . . . . New syntax and output for the meta-analysis command
        (help meta if installed) . . . . . S. Sharp and J. Sterne
        3/98    pp.6--8; STB Reprints Vol 7, pp.106--108
```

which reveals that STB-42 has a directory in it called `sbe16.1` containing files for ‘New syntax and output for the meta-analysis command’ and that help on the new command may be found using `help meta`, but only after the program has been installed. The authors are S. Sharp and J. Sterne. The command can be installed by clicking into the corresponding hyperlink in the **Stata Viewer** (or going through **Help** → **SJ & User-written Programs**, selecting <http://www.stata.com> (STB and more), clicking on `stb`, then `stb42`, `sbe16_1`) and selecting (**click here to install**). The program can also be installed using the commands

```
net stb 42 sbe16_1
```

(see `help net`). Note that `findit` first lists programs that have been published in the SJ and STB, followed by programs from other sites such as SSC. This order often does not reflect the chronological order of versions of a given program since SSC usually has the most up-to-date version. For example,

```
findit multilevel
```

lists versions of a user-written program `gllamm` that were published in 2002 in the SJ and in 2000 in the STB, followed by the current version on SSC. The latter can be obtained by clicking on the corresponding hyperlink in the **Stata Viewer** or using the command

```
ssc install gllamm
```

See `help ssc`.

1.12 Exercises

1. Use a text editor (e.g., Notepad, PFE, or the Stata Do-file Editor) to generate the dataset `test.dat` given below, where the columns are separated by tabs (make sure to save it as a text only or ASCII file).

v1	v2	v3
1	3	5
2	16	3
5	12	2

2. Read the data into Stata using `insheet` (see `help insheet`).
3. Click into the Data Editor and type in the variable `sex` with values 1, 2, and 1.
4. Define value labels for `sex` (1=male, 2=female).
5. Use `gen` to generate `id`, a subject index (from 1 to 3).
6. Use `rename` to rename the variables `v1` to `v3` to `time1` to `time3`. Also try doing this in a single command using `forvalues`.
7. Use `reshape` to convert the dataset to long shape.
8. Generate a variable `d` that is equal to the squared difference between the variable `time` at each occasion and the average of `time` for each subject.
9. Drop the observation corresponding to the third occasion for `id=2`.

Chapter 2

Data Description and Simple Inference: Female Psychiatric Patients

2.1 Description of data

The data to be used in this chapter consist of observations on 8 variables for 118 female psychiatric patients and are available in Hand *et al.* (1994). The variables are as follows:

- **age**: age in years
- **IQ**: intelligence score
- **anxiety**: anxiety (1=none, 2=mild, 3=moderate, 4=severe)
- **depress**: depression (1=none, 2=mild, 3=moderate, 4=severe)
- **sleep**: can you sleep normally? (1=yes, 2=no)
- **sex**: have you lost interest in sex? (1=no, 2=yes)
- **life**: have you thought recently about ending your life? (1=no, 2=yes)
- **weight**: increase in weight over last six months (in lbs)

The data are given in [Table 2.1](#); missing values are coded as `–99`. There are a variety of questions that might be addressed by these data; for example, how are the variables related? Do women who have recently contemplated suicide differ in any respects from those who have not? Also of interest are the correlations between anxiety and depression

and between weight change, age, and IQ. It should be noted, however, that any associations found from cross-sectional observational data like these are at best suggestive of causal relationships.

Table 2.1 Data in fem.dat

id	age	IQ	anx	depress	sleep	sex	life	weight
1	39	94	2	2	2	2	2	4.9
2	41	89	2	2	2	2	2	2.2
3	42	83	3	3	3	2	2	4.0
4	30	99	2	2	2	2	2	-2.6
5	35	94	2	1	1	2	1	-0.3
6	44	90	-99	1	2	1	1	0.9
7	31	94	2	2	-99	2	2	-1.5
8	39	87	3	2	2	2	1	3.5
9	35	-99	3	2	2	2	2	-1.2
10	33	92	2	2	2	2	2	0.8
11	38	92	2	1	1	1	1	-1.9
12	31	94	2	2	2	-99	1	5.5
13	40	91	3	2	2	2	1	2.7
14	44	86	2	2	2	2	2	4.4
15	43	90	3	2	2	2	2	3.2
16	32	-99	1	1	1	2	1	-1.5
17	32	91	1	2	2	-99	1	-1.9
18	43	82	4	3	2	2	2	8.3
19	46	86	3	2	2	2	2	3.6
20	30	88	2	2	2	2	1	1.4
21	34	97	3	3	-99	2	2	-99.0
22	37	96	3	2	2	2	1	-99.0
23	35	95	2	1	2	2	1	-1.0
24	45	87	2	2	2	2	2	6.5
25	35	103	2	2	2	2	1	-2.1
26	31	-99	2	2	2	2	1	-0.4
27	32	91	2	2	2	2	1	-1.9
28	44	87	2	2	2	2	2	3.7
29	40	91	3	3	2	2	2	4.5
30	42	89	3	3	2	2	2	4.2
31	36	92	3	-99	2	2	2	-99.0
32	42	84	3	3	2	2	2	1.7
33	46	94	2	-99	2	2	2	4.8
34	41	92	2	1	2	2	1	1.7
35	30	96	-99	2	2	2	2	-3.0
36	39	96	2	2	2	1	1	0.8
37	40	86	2	3	2	2	2	1.5
38	42	92	3	2	2	2	1	1.3
39	35	102	2	2	2	2	2	3.0
40	31	82	2	2	2	2	1	1.0
41	33	92	3	3	2	2	2	1.5
42	43	90	-99	-99	2	2	2	3.4
43	37	92	2	1	1	1	1	-99.0

Table 2.1 Data in fem.dat (continued)

44	32	88	4	2	2	2	1	-99.0
45	34	98	2	2	2	2	-99	0.6
46	34	93	3	2	2	2	2	0.6
47	42	90	2	1	1	2	1	3.3
48	41	91	2	1	1	1	1	4.8
49	31	-99	3	1	2	2	1	-2.2
50	32	92	3	2	2	2	2	1.0
51	29	92	2	2	2	1	2	-1.2
52	41	91	2	2	2	2	2	4.0
53	39	91	2	2	2	2	2	5.9
54	41	86	2	1	1	2	1	0.2
55	34	95	2	1	1	2	1	3.5
56	39	91	1	1	2	1	1	2.9
57	35	96	3	2	2	1	1	-0.6
58	31	100	2	2	2	2	2	-0.6
59	32	99	4	3	2	2	2	-2.5
60	41	89	2	1	2	1	1	3.2
61	41	89	3	2	2	2	2	2.1
62	44	98	3	2	2	2	2	3.8
63	35	98	2	2	2	2	1	-2.4
64	41	103	2	2	2	2	2	-0.8
65	41	91	3	1	2	2	1	5.8
66	42	91	4	3	-99	-99	2	2.5
67	33	94	2	2	2	2	1	-1.8
68	41	91	2	1	2	2	1	4.3
69	43	85	2	2	2	1	1	-99.0
70	37	92	1	1	2	2	1	1.0
71	36	96	3	3	2	2	2	3.5
72	44	90	2	-99	2	2	2	3.3
73	42	87	2	2	2	1	2	-0.7
74	31	95	2	3	2	2	2	-1.6
75	29	95	3	3	2	2	2	-0.2
76	32	87	1	1	2	2	1	-3.7
77	35	95	2	2	2	2	2	3.8
78	42	88	1	1	1	2	1	-1.0
79	32	94	2	2	2	2	1	4.7
80	39	-99	3	2	2	2	2	-4.9
81	34	-99	3	-99	2	2	1	-99.0
82	34	87	3	3	2	2	1	2.2
83	42	92	1	1	2	1	1	5.0
84	43	86	2	3	2	2	2	0.4
85	31	93	-99	2	2	2	2	-4.2
86	31	92	2	2	2	2	1	-1.1
87	36	106	2	2	2	1	2	-1.0
88	37	93	2	2	2	2	2	4.2
89	43	95	2	2	2	2	1	2.4
90	32	95	3	2	2	2	2	4.9
91	32	92	-99	-99	-99	2	2	3.0

Table 2.1 Data in fem.dat (continued)

92	32	98	2	2	2	2	2	-0.3
93	43	92	2	2	2	2	2	1.2
94	41	88	2	2	2	2	1	2.6
95	43	85	1	1	2	2	1	1.9
96	39	92	2	2	2	2	1	3.5
97	41	84	2	2	2	2	2	-0.6
98	41	92	2	1	2	2	1	1.4
99	32	91	2	2	2	2	2	5.7
100	44	86	3	2	2	2	2	4.6
101	42	92	3	2	2	2	1	-99.0
102	39	89	2	2	2	2	1	2.0
103	45	-99	2	2	2	2	2	0.6
104	39	96	3	-99	2	2	2	-99.0
105	31	97	2	-99	-99	-99	2	2.8
106	34	92	3	2	2	2	2	-2.1
107	41	92	2	2	2	2	2	-2.5
108	33	98	3	2	2	2	2	2.5
109	34	91	2	1	1	2	1	5.7
110	42	91	3	3	2	2	2	2.4
111	40	89	3	1	1	1	1	1.5
112	35	94	3	3	2	2	2	1.7
113	41	90	3	2	2	2	2	2.5
114	32	96	2	1	1	2	1	-99.0
115	39	87	2	2	2	1	2	-99.0
116	41	86	3	2	1	1	2	-1.0
117	33	89	1	1	1	1	1	6.5
118	42	-99	3	2	2	2	2	4.9

2.2 Group comparison and correlations

The data in Table 2.1 contain a number of *interval scale* or continuous variables (weight change, age, and IQ), *ordinal* variables (anxiety and depression), and *dichotomous* variables (sex and sleep) that we wish to compare between two groups of women: those who have thought about ending their lives and those who have not.

For interval scale variables, the most common statistical test is the *t*-test which assumes that the observations in the two groups are independent and are sampled from two populations each having a normal distribution and equal variances. A nonparametric alternative (which does not rely on the latter two assumptions) is the *Mann-Whitney U*-test.

For ordinal variables, either the Mann-Whitney *U*-test or a *chi-squared test* may be appropriate depending on the number of levels of the ordinal variable. The latter test can also be used to compare

dichotomous variables between the groups.

Continuous variables can be correlated using the *Pearson correlation*. If we are interested in the question whether the correlations differ significantly from zero, then a hypothesis test is available that assumes bivariate normality. A significance test not making this distributional assumption is also available; it is based on the correlation of the ranked variables, the *Spearman rank correlation*. Finally, if variables have only few categories, *Kendall's tau-b* provides a useful measure of correlation (see, e.g., Sprent and Smeeton, 2001). More details of these tests and correlation coefficients can be found in Altman (1991) and Agresti (2002).

2.3 Analysis using Stata

Assuming the data have been saved from a spreadsheet or statistical package (for example SAS or SPSS) as a tab-delimited ASCII file, `fem.dat`, they can be read using the instruction

```
insheet using fem.dat, clear
```

There are missing values which have been coded as `-99`. We replace these with Stata's missing value code `.` using

```
mvdecode _all, mv(-99)
```

The variable `sleep` has been entered incorrectly as `'3'` for subject 3. Such data entry errors can be detected using the command

```
codebook
```

which displays information on all variables; the output for `sleep` is shown below:

<code>sleep</code>	<code>SLEEP</code>
--------------------	--------------------

type:	numeric (byte)	
range:	[1,3]	units: 1
unique values:	3	missing : 5/118
tabulation:	Freq.	Value
	14	1
	98	2
	1	3
	5	.

Alternatively, we can detect errors using the `assert` command. For `sleep`, we would type

```
assert sleep==1|sleep==2|sleep==.
```

```
1 contradiction in 118 observations
assertion is false
```

Since we do not know what the correct code for **sleep** should have been, we can replace the incorrect value of 3 by ‘missing’

```
replace sleep=. if sleep==3
```

In order to have consistent coding for ‘yes’ and ‘no’, we recode the variable **sleep**

```
recode sleep 1=2 2=1
```

and to avoid confusion in the future, we label the values as follows:

```
label define yn 1 no 2 yes
label values sex yn
label values life yn
label values sleep yn
```

The last three commands could also have been carried out in a **foreach** loop:

```
foreach x in sex life sleep {
  label values `x' yn
}
```

First, we could compare the suicidal women with the non-suicidal by simply tabulating suitable summary statistics in each group. For example, for **IQ** the required instruction is

```
table life, contents(mean iq sd iq)
```

LIFE	mean(iq)	sd(iq)
no	91.2708	3.757204
yes	92.0984	5.0223

A more formal approach to comparing the two groups on say weight gain over the last six months might involve an independent samples *t*-test. First, however, we need to check whether the assumptions needed for the *t*-test appear to be satisfied for weight gain. One way this can be done is by plotting the variable **weight** as a boxplot for each group after defining appropriate labels:

```
label variable weight "weight change in last six months"
graph box weight, by(life) box(1, bfcolor(none)) /*
  */ box(2, bfcolor(none)) yline(0) medtype(line)
```

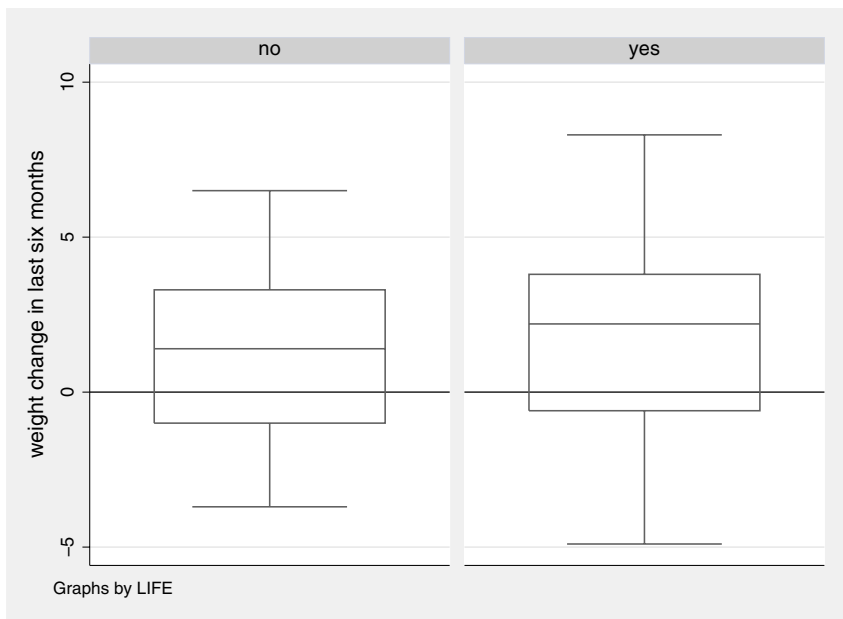


Figure 2.1: Boxplot of weight by group.

giving the graph shown in Figure 2.1. The `ylines(0)` option has placed a horizontal line at 0. (Note that in the instructions above, the ‘brackets’ for comments, `/*` and `*/` were used to make Stata ignore the line breaks in the middle of the `graph box` command in a do-file, but these ‘brackets’ should not be used in the Stata Command window.) The groups do not seem to differ much in their median weight change and the assumptions for the *t*-test seem reasonable because the distributions are symmetric with similar spread (box-heights represent inter-quartile ranges).

We can also check the assumption of normality more formally by plotting a normal quantile plot of suitably defined residuals. Here the difference between the observed weight changes and the group-specific mean weight changes can be used. If the normality assumption is satisfied, the quantiles of the residuals should be linearly related to the quantiles of the normal distribution. The residuals can be computed and plotted using

```
egen res=mean(weight), by(life)
replace res=weight-res
label variable res "residuals of t-test for weight"
qnorm res, title("Normal Q-Q plot")
```

The points in the Q-Q plot in Figure 2.2 appear to be sufficiently close to the straight line to justify the normality assumption.

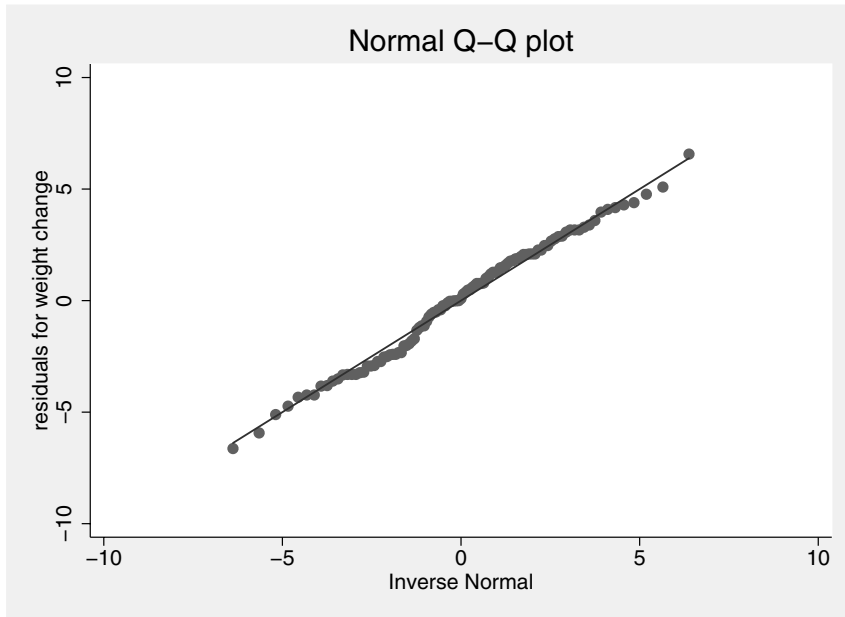


Figure 2.2: Normal Q-Q plot of residuals of weight change.

We could also test whether the variances differ significantly using

```
sdtest weight, by(life)
```

giving the output shown in Display 2.1. There is no evidence that the variances differ ($p = 0.57$). Note that the test for equal variances is only appropriate if the variable may be assumed to be normally distributed in each population.

Having found no strong evidence that the assumptions of the t -test are not valid for weight gain, we can proceed to apply a t -test:

```
ttest weight, by(life)
```

Display 2.2 shows that the difference in means is estimated as -0.32 with a 95% confidence interval from -1.39 to 0.74 . The two-tailed p -value is 0.55 , so there is no evidence that the populations differ in their mean weight change.

Now suppose we wish to compare the prevalence of depression between suicidal and non-suicidal women. The two categorical variables

Variance ratio test

Group	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf. Interval]	
no	45	1.408889	.3889616	2.609234	.6249883	2.19279
yes	61	1.731148	.3617847	2.825629	1.00747	2.454825
combined	106	1.59434	.2649478	2.727805	1.068997	2.119682

Ho: sd(no) = sd(yes)

F(44,60) observed = F_obs = 0.853
 F(44,60) lower tail = F_L = F_obs = 0.853
 F(44,60) upper tail = F_U = 1/F_obs = 1.173

Ha: sd(no) < sd(yes) Ha: sd(no) != sd(yes) Ha: sd(no) > sd(yes)
 P < F_obs = 0.2919 P < F_L + P > F_U = 0.5724 P > F_obs = 0.7081

Display 2.1

Two-sample t test with equal variances

Group	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf. Interval]	
no	45	1.408889	.3889616	2.609234	.6249883	2.19279
yes	61	1.731148	.3617847	2.825629	1.00747	2.454825
combined	106	1.59434	.2649478	2.727805	1.068997	2.119682
diff		-.3222587	.5376805		-1.388499	.743982

Degrees of freedom: 104

Ho: mean(no) - mean(yes) = diff = 0

Ha: diff < 0 Ha: diff != 0 Ha: diff > 0
 t = -0.5993 t = -0.5993 t = -0.5993
 P < t = 0.2751 P > |t| = 0.5502 P > t = 0.7249

Display 2.2

can be cross-tabulated and the appropriate chi-squared statistic calculated using a single command:

```
tabulate life depress, row chi2
```

The output is shown in Display 2.3. Here the `row` option was used

Key				
<i>frequency</i>				
<i>row percentage</i>				

LIFE	DEPRESS			Total
	1	2	3	
no	26	24	1	51
	50.98	47.06	1.96	100.00
yes	0	42	16	58
	0.00	72.41	27.59	100.00
Total	26	66	17	109
	23.85	60.55	15.60	100.00

Pearson chi2(2) = 43.8758 Pr = 0.000

Display 2.3

to display row-percentages, making it easier to compare the groups of women. For example, 51% of non-suicidal women are not depressed at all, compared with 0% of suicidal women. The value of the chi-squared statistic implies that there is a highly significant association between depression and suicidal thoughts ($p < 0.001$). Note that this test does not take account of the ordinal nature of depression and is therefore likely to be less sensitive than, for example, ordinal regression (see Chapter 6). Since some cells in the cross classification have only small counts, we might want to use Fisher's exact test (see Everitt, 1992) rather than the chi-squared test. The necessary command (without reproducing the table) is as follows:

```
tabulate life depress, exact nofreq
```

```
Fisher's exact = 0.000
```

Again we find evidence of a strong relationship between depression and suicidal thoughts.

We now test the null hypothesis that the proportion of women who have lost interest in sex does not differ between the populations of suicidal and non-suicidal women. We can obtain the relevant table and both the chi-squared test and Fisher's exact tests using

```
tabulate life sex, row chi2 exact
```

with results shown in Display 2.4. Therefore, those who have thought

Key			
	<i>frequency</i>		
	<i>row percentage</i>		

LIFE	SEX		Total
	no	yes	
no	12	38	50
	24.00	76.00	100.00
yes	5	58	63
	7.94	92.06	100.00
Total	17	96	113
	15.04	84.96	100.00

Pearson chi2(1) =	5.6279	Pr = 0.018
Fisher's exact =		0.032
1-sided Fisher's exact =		0.017

Display 2.4

about ending their lives are more likely to have lost interest in sex than those who have not (92% compared with 76%).

The correlations between the three variables **weight**, **IQ**, and **age** can be found using the **correlate** command

```
corr weight iq age
```

(see Display 2.5). This correlation matrix has been evaluated for those 100 women who had complete data on all three variables. An alternative approach is to use the command **pwcorr** to include, for each correlation, all observations that have complete data for the corresponding pair of variables, resulting in different sample sizes for different correlations. These *pairwise* correlations can be obtained together with the sample sizes and *p*-values using

```
pwcorr weight iq age, obs sig
```

(obs=100)

	weight	iq	age
weight	1.0000		
iq	-0.2920	1.0000	
age	0.4131	-0.4363	1.0000

Display 2.5

(see Display 2.6).

	weight	iq	age
weight	1.0000		
	107		
iq	-0.2920	1.0000	
	0.0032		
	100	110	
age	0.4156	-0.4345	1.0000
	0.0000	0.0000	
	107	110	118

Display 2.6

The corresponding scatterplot matrix is obtained using `graph matrix` as follows:

```
graph matrix weight iq age, half jitter(1) /*  
    */ msymbol(circle_hollow) msize(small) /*  
    */ diagonal("Weight change" "IQ" "Age")
```

where `jitter(1)` randomly moves the points by a very small amount to stop them overlapping completely due to the discrete nature of age and IQ. The resulting graph is shown in [Figure 2.3](#). We see that older, psychiatrically ill females tend to put on more weight than younger ones as do less intelligent women. However, older women in this sample also tended to be less intelligent so that age and intelligence are confounded.

It is of some interest to assess whether age and weight change have the same relationship in suicidal as in non-suicidal women. We shall do this informally by constructing a single scatterplot of weight change

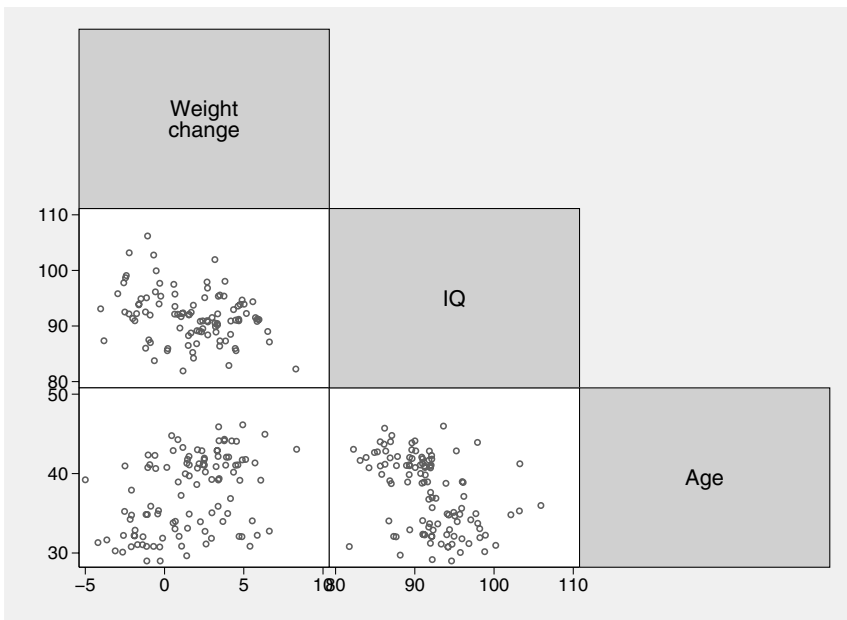


Figure 2.3: Scatterplot matrix for weight, IQ, and age.

against age in which the women in the two groups are represented by different symbols. This is easily done by simply specifying two (overlaid) scatterplots as follows:

```
twoway (scatter weight age if life==1, msymbol(circle) /*
      */ mcolor(black) jitter(1)) /*
      */ (scatter weight age if life==2, msymbol(x) /*
      */ mcolor(black) jitter(1)), legend(order(1 "no" 2 "yes"))
```

The resulting graph in [Figure 2.4](#) shows that within both groups, higher age is associated with larger weight increases, and the groups do not form distinct ‘clusters’.

Finally, an appropriate correlation between depression and anxiety is Kendall’s tau-b which can be obtained using

```
ktau depress anxiety
```

```
Number of obs =      107
Kendall's tau-a =      0.2827
Kendall's tau-b =      0.4951
Kendall's score =     1603
SE of score =      288.279 (corrected for ties)
Test of Ho: depress and anxiety are independent
Prob > |z| =          0.0000 (continuity corrected)
```

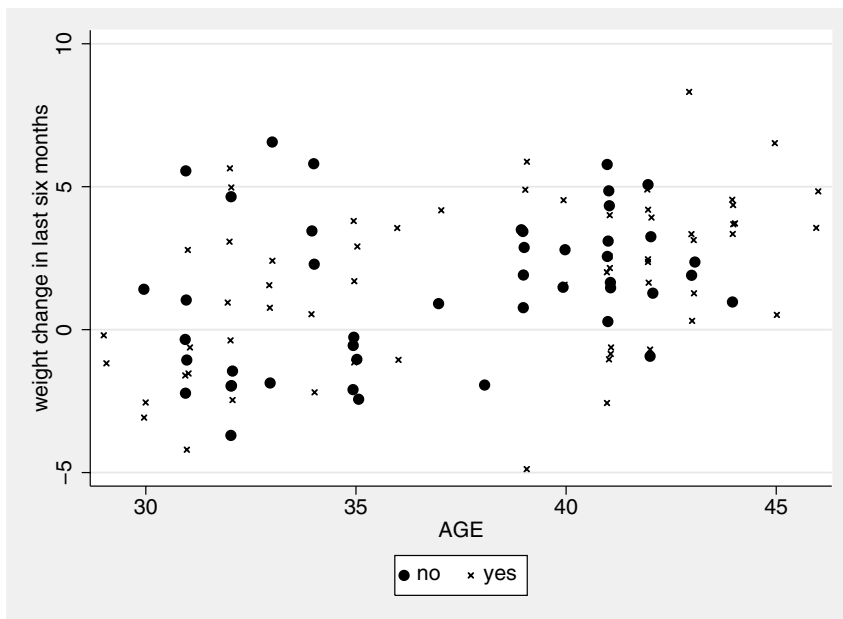


Figure 2.4: Scatterplot of weight against age.

giving a value of 0.50 with an approximate p -value of $p < 0.001$. Depression and anxiety are clearly related in these psychiatrically unwell women.

2.4 Exercises

1. Tabulate the mean weight change by level of depression.
2. By looping through the variables `age`, `iq`, and `weight` using `foreach`, tabulate the means and standard deviations for each of these variables by `life`.
3. Use `search nonparametric` or `search mann` or `search whitney` to find help on how to run the Mann-Whitney U -test.
4. Compare the weight changes between the two groups using the Mann-Whitney U -test.
5. Form a scatterplot for `IQ` and `age` using different symbols for the two groups (`life=1` and `life=2`). Explore the use of the option `jitter(#)` for different integers `#` to stop symbols overlapping.
6. Having tried out all these commands interactively, create a do-file containing these commands and run the do-file. In the `graph`

commands, use the option `saving(filename,replace)` to save the graphs in the current directory and view the graphs later using the command `graph use filename`.

See also [Exercises in Chapter 6](#).

Chapter 3

Multiple Regression: Determinants of Pollution in U.S. Cities

3.1 Description of data

Data on air pollution in 41 U.S. cities were collected by Sokal and Rohlf (1981) from several U.S. government publications and are reproduced here in [Table 3.1](#). (The data are also in Hand *et al.*, 1994.) There is a single dependent variable, `so2`, the annual mean concentration of sulphur dioxide, in micrograms per cubic meter. These data are means for the three years 1969 to 1971 for each city. The values of six explanatory variables, two of which concern human ecology and four climate, are also recorded; details are as follows:

- `temp`: average annual temperature in °F
- `manuf`: number of manufacturing enterprises employing 20 or more workers
- `pop`: population size (1970 census) in thousands
- `wind`: average annual wind speed in miles per hour
- `precip`: average annual precipitation in inches
- `days`: average number of days with precipitation per year

The main question of interest about these data is how the pollution level as measured by sulphur dioxide concentration is determined by the six explanatory variables. The central method of analysis will be *multiple regression*.

Table 3.1 Data in usair.dat

Town	SO2	temp	manuf	pop	wind	precip	days
Phoenix	10	70.3	213	582	6.0	7.05	36
Little Rock	13	61.0	91	132	8.2	48.52	100
San Francisco	12	56.7	453	716	8.7	20.66	67
Denver	17	51.9	454	515	9.0	12.95	86
Hartford	56	49.1	412	158	9.0	43.37	127
Wilmington	36	54.0	80	80	9.0	40.25	114
Washington	29	57.3	434	757	9.3	38.89	111
Jackson	14	68.4	136	529	8.8	54.47	116
Miami	10	75.5	207	335	9.0	59.80	128
Atlanta	24	61.5	368	497	9.1	48.34	115
Chicago	110	50.6	3344	3369	10.4	34.44	122
Indianapolis	28	52.3	361	746	9.7	38.74	121
Des Moines	17	49.0	104	201	11.2	30.85	103
Wichita	8	56.6	125	277	12.7	30.58	82
Louisville	30	55.6	291	593	8.3	43.11	123
New Orleans	9	68.3	204	361	8.4	56.77	113
Baltimore	47	55.0	625	905	9.6	41.31	111
Detroit	35	49.9	1064	1513	10.1	30.96	129
Minneapolis	29	43.5	699	744	10.6	25.94	137
Kansas	14	54.5	381	507	10.0	37.00	99
St. Louis	56	55.9	775	622	9.5	35.89	105
Omaha	14	51.5	181	347	10.9	30.18	98
Albuquerque	11	56.8	46	244	8.9	7.77	58
Albany	46	47.6	44	116	8.8	33.36	135
Buffalo	11	47.1	391	463	12.4	36.11	166
Cincinnati	23	54.0	462	453	7.1	39.04	132
Cleveland	65	49.7	1007	751	10.9	34.99	155
Columbia	26	51.5	266	540	8.6	37.01	134
Philadelphia	69	54.6	1692	1950	9.6	39.93	115
Pittsburgh	61	50.4	347	520	9.4	36.22	147
Providence	94	50.0	343	179	10.6	42.75	125
Memphis	10	61.6	337	624	9.2	49.10	105
Nashville	18	59.4	275	448	7.9	46.00	119
Dallas	9	66.2	641	844	10.9	35.94	78
Houston	10	68.9	721	1233	10.8	48.19	103
Salt Lake City	28	51.0	137	176	8.7	15.17	89
Norfolk	31	59.3	96	308	10.6	44.68	116
Richmond	26	57.8	197	299	7.6	42.59	115
Seattle	29	51.1	379	531	9.4	38.79	164
Charleston	31	55.2	35	71	6.5	40.75	148
Milwaukee	16	45.7	569	717	11.8	29.07	123

3.2 The multiple regression model

The multiple regression model has the general form

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \epsilon_i \quad (3.1)$$

where y_i is a continuous response (or dependent) variable for the i th member of the sample, $x_{1i}, x_{2i}, \dots, x_{pi}$ are a set of explanatory (or independent) variables or covariates, $\beta_1, \beta_2, \dots, \beta_p$ are regression coefficients, and ϵ_i is a residual or error term with zero mean that is uncorrelated with the explanatory variables. It follows that the expected value of the response for given values of the covariates $\mathbf{x}_i^T = (1, x_{1i}, x_{2i}, \dots, x_{pi})$ is

$$E(y_i | \mathbf{x}_i) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi}.$$

This is also the value we would predict for a new individual with covariate values \mathbf{x}_i if we knew the regression coefficients.

Each regression coefficient represents the mean change in the response variable when the corresponding explanatory variable increases by one unit and all other explanatory variables remain constant. The coefficients therefore represent the effects of each explanatory variable, controlling for all other explanatory variables in the model, giving rise to the term ‘partial’ regression coefficients. The residual is the difference between the observed value of the response and the value predicted by the explanatory variables.

The regression coefficients $\beta_0, \beta_1, \dots, \beta_p$ are generally estimated by *least squares*; in other words the estimates $\hat{\beta}_k$ minimize the sum of the squared differences between observed and predicted responses, or the sum of squared estimated residuals,

$$\sum_{i=1}^n [y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_p x_{pi})]^2. \quad (3.2)$$

Significance tests for the regression coefficients can be derived by assuming that the error terms are independently normally distributed with zero mean and constant variance σ^2 .

For n observations of the response and explanatory variables, the regression model may be written concisely as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad (3.3)$$

where \mathbf{y} is the $n \times 1$ vector of responses, \mathbf{X} is an $n \times (p + 1)$ matrix of known constants, the first column containing a series of ones corre-

sponding to the term β_0 in (3.1), and the remaining columns values of the explanatory variables. The elements of the vector $\boldsymbol{\beta}$ are the regression coefficients β_0, \dots, β_p , and those of the vector $\boldsymbol{\epsilon}$, the error terms $\epsilon_1, \dots, \epsilon_n$. The least squares estimates of the regression coefficients can then be written as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

and the variances and covariances of these estimates can be found from

$$\mathbf{S}_{\hat{\boldsymbol{\beta}}} = s^2(\mathbf{X}'\mathbf{X})^{-1},$$

where s^2 is an estimate of the residual variance σ^2 given by the sum of squared estimated residuals in equation (3.2) divided by $n - p - 1$. The goodness of fit of the model is often expressed in terms of the percentage of variance of the response variable explained by the explanatory variables, given by R^2 , the square of the *multiple correlation coefficient*. The multiple correlation coefficient is just the correlation between the observed response variable and the response predicted by the model. For full details of multiple regression see, for example, Rawlings *et al.* (1998).

3.3 Analysis using Stata

Assuming the data are available as an ASCII file `usair.dat` in the current directory and that the file contains city names (abbreviated versions of those in [Table 3.1](#)), they may be read in for analysis using the following instruction:

```
infile str10 town so2 temp manu pop /*
      */ wind precip days using usair.dat
```

Here we had to declare the ‘type’ for the string variable `town` as `str10` which stands for ‘string variable with 10 characters’.

Before undertaking a formal regression analysis of these data, it will be helpful to examine them graphically using a scatterplot matrix. Such a display is useful in assessing the general relationships between the variables, in identifying possible outliers, and in highlighting potential collinearity problems amongst the explanatory variables. The basic plot can be obtained using

```
graph matrix so2 temp manu pop wind precip days
```

The resulting diagram is shown in [Figure 3.1](#). Several of the scatterplots show evidence of outliers, and the relationship between `manu`

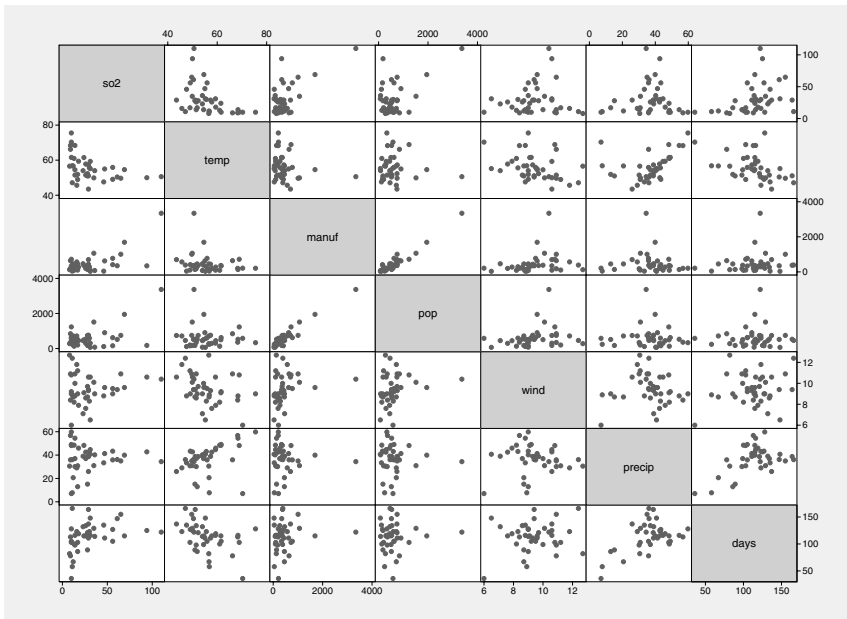


Figure 3.1: Scatterplot matrix.

and **pop** is very strong suggesting that using both as explanatory variables in a regression analysis may lead to problems (see later). The relationships of particular interest, namely those between **so2** and the explanatory variables (the relevant scatterplots are those in the first row of Figure 3.1), indicate some possible nonlinearity. A more informative, although slightly more ‘messy’ diagram can be obtained if the plotted points are labeled with the associated town name. We first create a variable containing the first three characters of the strings in **town** using the function **substr()**

```
gen str3 twn = substr(town,1,3)
```

We can then create a scatterplot matrix with these three-character town labels using

```
graph matrix so2-days, msymbol(none) mlabel(twn) /*
    */ mlabposition(0)
```

The **mlabel()** option labels the points with the names in the **tw**n variable. By default, a symbol would also be plotted and this can be suppressed using **msymbol(none)**; **mlabposition(0)** centers the labels where the symbol would normally go. The resulting diagram appears in [Figure 3.2](#). Clearly, Chicago and to a lesser extent Philadelphia might be considered outliers. Chicago has such a high degree of pollution

compared with the other cities that it should perhaps be considered as a special case and excluded from further analysis. We can remove Chicago using

```
drop if town=="Chicago"
```

The command `regress` may be used to fit a basic multiple regression model. The necessary Stata instruction to regress sulphur dioxide concentration on the six explanatory variables is

```
regress so2 temp manuf pop wind precip days
```

or, alternatively,

```
regress so2 temp-days
```

(see Display 3.1).

Source	SS	df	MS			
Model	8203.60523	6	1367.26754	Number of obs = 40		
Residual	7282.29477	33	220.675599	F(6, 33) = 6.20		
Total	15485.9	39	397.074359	Prob > F = 0.0002		
				R-squared = 0.5297		
				Adj R-squared = 0.4442		
				Root MSE = 14.855		

so2	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
temp	-1.268452	.6305259	-2.01	0.052	-2.551266	.0143631
manuf	.0654927	.0181777	3.60	0.001	.0285098	.1024756
pop	-.039431	.0155342	-2.54	0.016	-.0710357	-.0078264
wind	-3.198267	1.859713	-1.72	0.095	-6.981881	.5853468
precip	.5136846	.3687273	1.39	0.173	-.2364966	1.263866
days	-.0532051	.1653576	-0.32	0.750	-.3896277	.2832175
_cons	111.8709	48.07439	2.33	0.026	14.06278	209.679

Display 3.1

The main features of interest in the output in Display 3.1 are the analysis of variance table and the parameter estimates. In the former, the ratio of the model mean square to the residual mean square gives an F -test for the hypothesis that *all* the regression coefficients in the fitted model are zero (except the constant β_0). The resulting F -statistic with 6 and 33 degrees of freedom takes the value 6.20 and is shown on the right-hand side; the associated p -value is very small. Consequently, the hypothesis is rejected. The square of the multiple correlation coefficient (R^2) is 0.53 showing that 53% of the variance of sulphur dioxide concentration is accounted for by the six explanatory variables of interest.

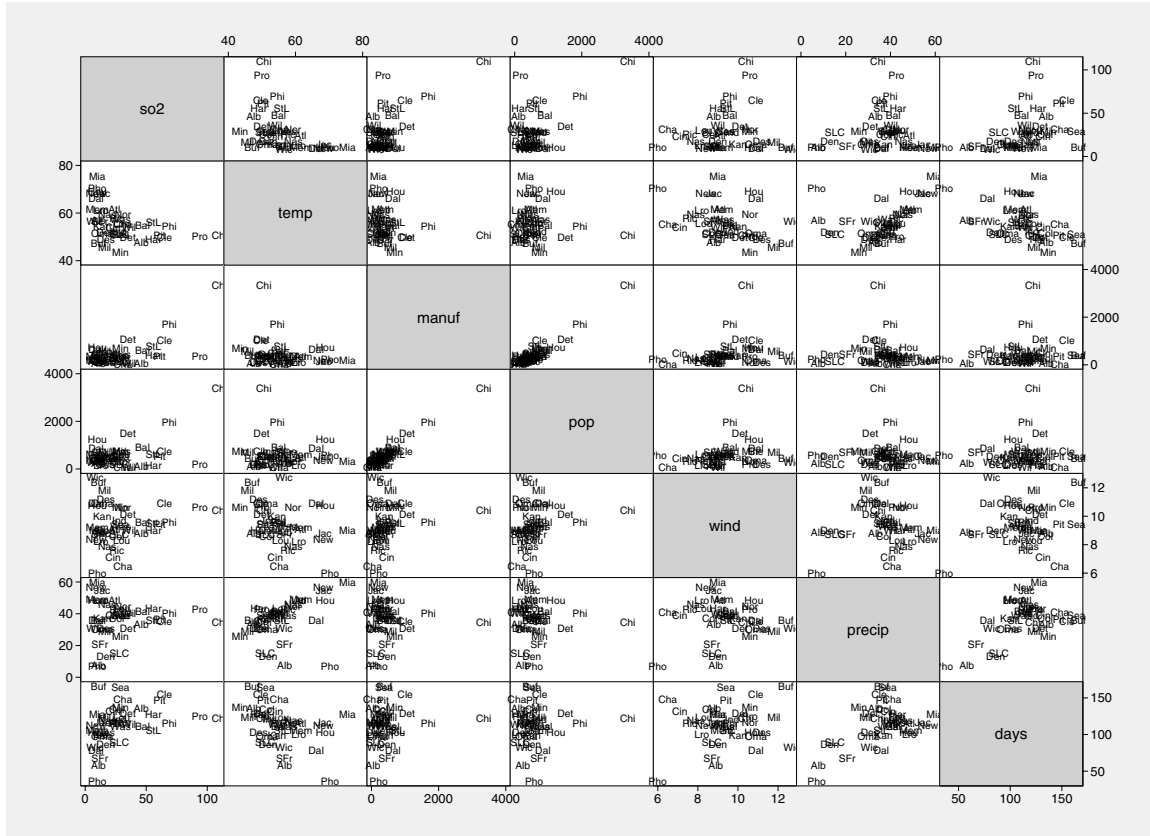


Figure 3.2: Scatterplot matrix with town labels.

The adjusted R^2 statistic is an estimate of the population R^2 taking account of the fact that the parameters were estimated from the data. The statistic is calculated as

$$\text{adj } R^2 = 1 - \frac{(n-1)(1-R^2)}{n-p} \quad (3.4)$$

where n is the number of observations used in fitting the model. The root MSE is simply the square root of the residual mean square in the analysis of variance table, which itself is an estimate of the parameter σ^2 . The estimated regression coefficients give the estimated change in the response variable produced by a unit change in the corresponding explanatory variable with the remaining explanatory variables held constant.

One concern generated by the initial graphical material on this data was the strong relationship between the two explanatory variables **manuf** and **pop**. The correlation of these two variables is obtained by using

```
correlate manuf pop
```

(obs=40)

	manuf	pop
manuf	1.0000	
pop	0.8906	1.0000

The strong linear dependence might be a source of collinearity problems and can be investigated further by calculating what are known as *variance inflation factors* for each of the explanatory variables. These are given by

$$\text{VIF}(x_k) = \frac{1}{1 - R_k^2} \quad (3.5)$$

where $\text{VIF}(x_k)$ is the variance inflation factor for explanatory variable x_k , and R_k^2 is the square of the multiple correlation coefficient obtained from regressing x_k on the remaining explanatory variables.

The variance inflation factors can be obtained using the **vif** command after **regress**:

```
vif
```

(see Display 3.2).

Chatterjee *et al.* (1999) give the following ‘rules-of-thumb’ for evaluating these factors:

Variable	VIF	1/VIF
manuf	6.28	0.159275
pop	6.13	0.163165
temp	3.72	0.269156
days	3.47	0.287862
precip	3.41	0.293125
wind	1.26	0.790619
Mean VIF	4.05	

Display 3.2

- Values larger than 10 give evidence of collinearity.
- A mean of the VIF factors considerably larger than one suggests collinearity.

Here there are no values greater than 10 (as an exercise we suggest readers also calculate the VIFs when the observations for Chicago are included), but the mean value of 4.05 gives some cause for concern. A simple (although not necessarily the best) way to proceed is to drop one of `manuf` or `pop`. Another possibility is to replace `manuf` by a new variable equal to `manuf` divided by `pop`, representing the number of large manufacturing enterprises per thousand inhabitants (see Exercise 2). However, we shall simply exclude `manuf` and repeat the regression analysis using the five remaining explanatory variables:

```
regress so2 temp pop wind precip days
```

The output is shown in Display 3.3.

Now recompute the variance inflation factors:

```
vif
```

The variance inflation factors in Display 3.4 are now satisfactory.

The very general hypothesis concerning all regression coefficients mentioned previously is not usually of great interest in most applications of multiple regression because it is most unlikely that all the chosen explanatory variables will be unrelated to the response variable. The more interesting question is whether a subset of the regression coefficients are zero, implying that not all the explanatory variables are of use in predicting the response variable. A preliminary assessment of the likely importance of each explanatory variable can be made using the table of estimated regression coefficients and associated statistics. Using a conventional 5% criterion, the only ‘significant’ coefficient is that for the variable `temp`. Unfortunately, this very simple approach is not in general suitable, since in most cases the explanatory variables

Source	SS	df	MS			
Model	5339.03465	5	1067.80693			
Residual	10146.8654	34	298.437216			
Total	15485.9	39	397.074359			

				Number of obs =	40
				F(5, 34) =	3.58
				Prob > F	= 0.0105
				R-squared	= 0.3448
				Adj R-squared	= 0.2484
				Root MSE	= 17.275

so2	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
temp	-1.867665	.7072827	-2.64	0.012	-3.305037	-.430294
pop	.0113969	.0075627	1.51	0.141	-.0039723	.0267661
wind	-3.126429	2.16257	-1.45	0.157	-7.5213	1.268443
precip	.6021108	.4278489	1.41	0.168	-.2673827	1.471604
days	-.020149	.1920012	-0.10	0.917	-.4103424	.3700445
_cons	135.8565	55.36797	2.45	0.019	23.33529	248.3778

Display 3.3

Variable	VIF	1/VIF
days	3.46	0.288750
temp	3.46	0.289282
precip	3.40	0.294429
wind	1.26	0.790710
pop	1.07	0.931015
Mean VIF	2.53	

Display 3.4

are correlated, and the t -tests will not be independent of each other. Consequently, removing a particular variable from the regression will alter both the estimated regression coefficients of the remaining variables and their standard errors. A more involved approach to identifying important subsets of explanatory variables is therefore required. A number of procedures are available.

1. *Confirmatory approach*: Considering a model with a small set of explanatory variables chosen according to substantive theory and testing particular *a priori* hypotheses. Modifying the model by removing some variables, considering interactions, etc. to achieve a better fit to the data.
2. *Exploratory approach*: Automatic selection methods, which are of the following types:
 - a. *Forward selection*: This method starts with a model containing none of the explanatory variables and then considers variables one by one for inclusion. At each step, the variable added is the one that results in the biggest increase in the model or regression sum of squares. An F -type statistic is used to judge when further additions would not represent a significant improvement in the model.
 - b. *Backward elimination*: Here variables are considered for removal from an initial model containing all the explanatory variables. At each stage, the variable chosen for exclusion is the one leading to the smallest reduction in the regression sum of squares. Again, an F -type statistic is used to judge when further exclusions would represent a significant deterioration in the model.
 - c. *Stepwise regression*: This method is essentially a combination of the previous two. The forward selection procedure is used to add variables to an existing model and, after each addition, a backward elimination step is introduced to assess whether variables entered earlier might now be removed because they no longer contribute significantly to the model.

It is clear that the automatic selection methods are based on a large number of significance tests, one for each variable considered for inclusion or exclusion in each step. It is well-known that the probability of a false positive result or Type I error increases with the number of tests. The chosen model should therefore be interpreted with extreme caution, particularly if there were a large number of candidate variables. Another problem with the three automatic procedures is that they often do not lead to the same model; see also Harrell (2001) for a discussion of model selection strategies. Although we would generally not recommend automatic procedures, we will use them here for illustration.

But first we will take the confirmatory approach to investigate if climate (**temp**, **wind**, **precip**, **days**) or human ecology (**pop**) or both are important predictors of air pollution. We treat these groups of variables as single terms, allowing either all variables in a group to be included or none. This can be done using the following instruction:

`sw regress so2 (temp wind precip days) (pop), pe(0.05)`
(see Display 3.5). Note the grouping as required. The `pe` term specifies

begin with empty model						
p = 0.0119 <	0.0500	adding	temp	wind	precip	days
Source	SS	df	MS			
Model	4661.27545	4	1165.31886	Number of obs = 40		
Residual	10824.6246	35	309.274987	F(4, 35) = 3.77		
Total	15485.9	39	397.074359	Prob > F = 0.0119		
				R-squared = 0.3010		
				Adj R-squared = 0.2211		
				Root MSE = 17.586		

so2	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
temp	-1.689848	.7099204	-2.38	0.023	-3.131063	-.2486329
wind	-2.309449	2.13119	-1.08	0.286	-6.635996	2.017097
precip	.5241595	.4323535	1.21	0.234	-.3535647	1.401884
days	.0119373	.1942509	0.06	0.951	-.382413	.4062876
_cons	123.5942	55.75236	2.22	0.033	10.41091	236.7775

Display 3.5

the significance level of the F -test for addition to the model; terms with a p -value less than the figure specified will be included. Here, only the climate variables are shown since they are jointly significant ($p = 0.0119$) at the significance level for inclusion.

As an illustration of the automatic selection procedures, the following Stata instruction applies the backward elimination method, with explanatory variables whose F -values for removal have associated p -values greater than 0.2 being removed:

`sw regress so2 temp pop wind precip days, pr(0.2)`
(see Display 3.6). With the chosen significance level, only the variable **days** is excluded.

The next stage in the analysis should be an examination of the *residuals* from the chosen model; that is, the differences between the observed and fitted values of sulphur dioxide concentration. Such a procedure is vital for assessing model assumptions, identifying any un-

begin with full model					
p = 0.9170 >= 0.2000 removing days					
Source	SS	df	MS		
Model	5335.74801	4	1333.937	Number of obs = 40	
Residual	10150.152	35	290.004343	F(4, 35) = 4.60	
Total	15485.9	39	397.074359	Prob > F = 0.0043	
				R-squared = 0.3446	
				Adj R-squared = 0.2696	
				Root MSE = 17.03	

so2	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
temp	-1.810123	.4404001	-4.11	0.000	-2.704183	-.9160635
pop	.0113089	.0074091	1.53	0.136	-.0037323	.0263501
wind	-3.085284	2.096471	-1.47	0.150	-7.341347	1.170778
precip	.5660172	.2508601	2.26	0.030	.0567441	1.07529
_cons	131.3386	34.32034	3.83	0.001	61.66458	201.0126

Display 3.6

usual features in the data indicating outliers, and suggesting possibly simplifying transformations. The most useful ways of examining the residuals are graphical, and the most commonly used plots are as follows:

- A plot of the residuals against each explanatory variable in the model. The presence of a curvilinear relationship, for example, would suggest that a higher-order term, perhaps a quadratic in the explanatory variable, should be added to the model.
- A plot of the residuals against predicted values of the response variable. If the variance of the response appears to increase or decrease with predicted value, a transformation of the response may be in order.
- A normal probability plot of the residuals—after all systematic variation has been removed from the data, the residuals should look like a sample from the normal distribution. A plot of the ordered residuals against the expected order statistics from a normal distribution provides a graphical check on this assumption.

The first two plots can be obtained after using the **regress** procedure with the **rvpplot** and **rvfplot** instructions. For example, for the model chosen by the backward selection procedure, a plot of residuals against predicted values with the first three letters of the town name used to label the points is obtained using the instruction

```
rvfplot, mlabel(twn)
```

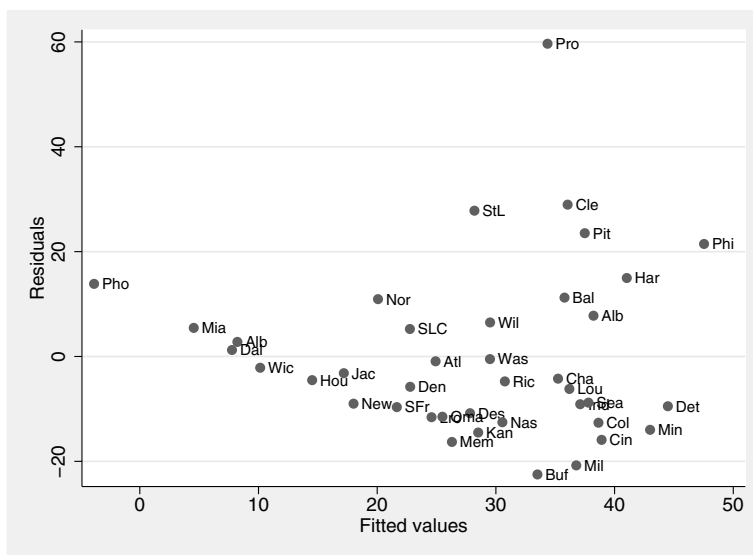


Figure 3.3: Residuals against predicted response.

The resulting plot is shown in Figure 3.3, and indicates a possible problem, namely the apparently increasing variance of the residuals as the fitted values increase (see also [Chapter 7](#)). Perhaps some thought needs to be given to the possible transformations of the response variable (see exercises).

Next, graphs of the residuals plotted against each of the four explanatory variables can be obtained using the following `foreach` loop:

```
foreach x in pop temp wind precip {
    rvppplot `x', mlabel(twn)
    more
}
```

Here `more` causes Stata to pause after each graph has been plotted until the user presses any key. The resulting graphs are shown in [Figures 3.4 to 3.7](#). In each graph the point corresponding to the town *Providence* is somewhat distant from the bulk of the points, and the graph for `wind` has perhaps a ‘hint’ of a curvilinear structure. Note that the appearance of these graphs could be improved using the `mlabvposition(varname)` option to specify the ‘clock positions’ (e.g. 12 is straight above) of the labels relative to the points.

The simple residuals plotted by `rvfplot` and `rvppplot` have a distribution that is scale dependent because the variance of each is a function of both σ^2 and the diagonal values of the so-called ‘hat’ matrix, \mathbf{H} , given by

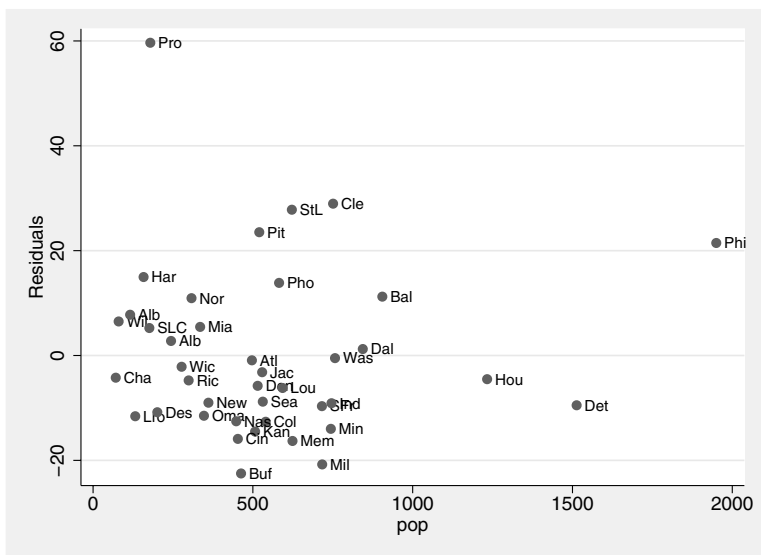


Figure 3.4: Residuals against population.

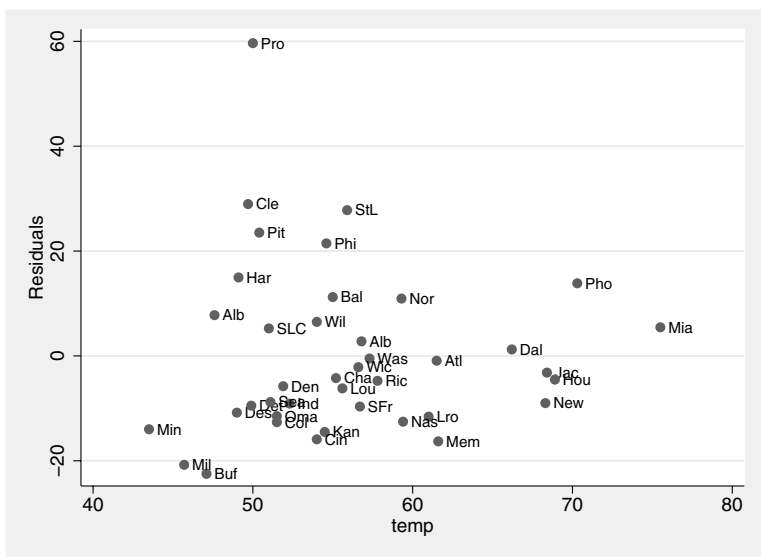


Figure 3.5: Residuals against temperature.

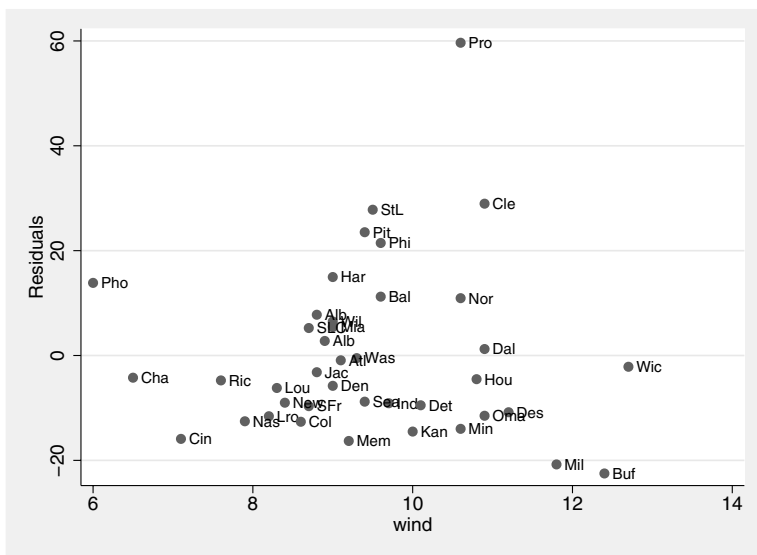


Figure 3.6: Residuals against wind speed.

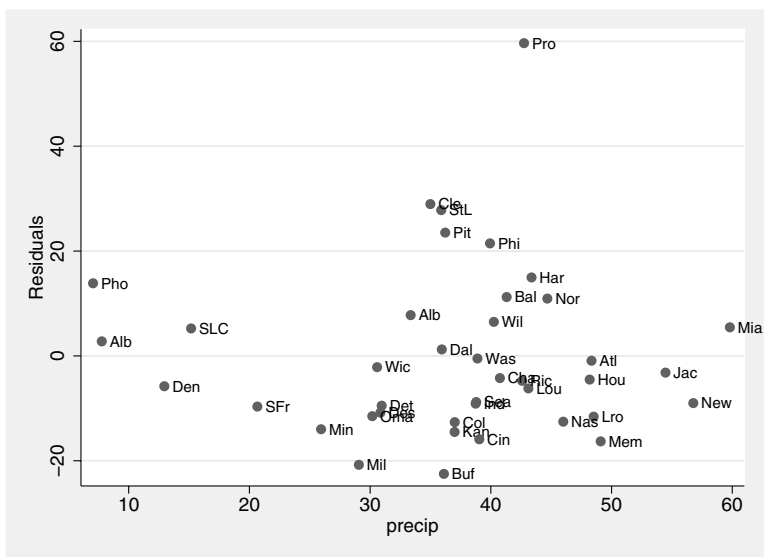


Figure 3.7: Residuals against precipitation.

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \quad (3.6)$$

(see Cook and Weisberg (1982) for a full explanation of the hat matrix). Consequently, it is often more useful to work with a standardized version

$$r_i = \frac{y_i - \hat{y}_i}{s\sqrt{1 - h_{ii}}}, \quad (3.7)$$

where s^2 is the estimate of σ^2 , \hat{y}_i is the predicted value of the response, and h_{ii} is the i th diagonal element of \mathbf{H} .

These standardized residuals can be obtained using the `predict` command. For example, to obtain a normal probability plot of the standardized residuals and to plot them against the fitted values requires the following instructions:

```
predict fit
predict sdres, rstandard
pnorm sdres
twoway scatter sdres fit, mlabel(twn)
```

The first instruction stores the fitted values in the variable `fit`, the second stores the standardized residuals in the variable `sdres`, the third produces a normal probability plot (Figure 3.8), and the last instruction produces the graph of standardized residuals against fitted values, which is shown in Figure 3.9.

The normal probability plot indicates that the distribution of the residuals departs somewhat from normality. The pattern in the plot shown in Figure 3.9 is identical to that in Figure 3.3 but here values outside $(-2, 2)$ indicate possible outliers, in this case the point corresponding to the town Providence. Analogous plots to those in Figures 3.4 to 3.7 could be obtained in the same way.

A rich variety of other diagnostics for investigating fitted regression models has been developed and many of these are available with the `regress` procedure. Illustrated here is the use of two of these, namely the *partial residual plot* (Mallows, 1973) and *Cook's distance* (Cook, 1977, 1979). The former are useful in identifying whether, for example, quadratic or higher order terms are needed for any of the explanatory variables; the latter measures the change to the estimates of the regression coefficients that results from deleting each observation and can be used to indicate those observations that may be having an undue influence on the estimates.

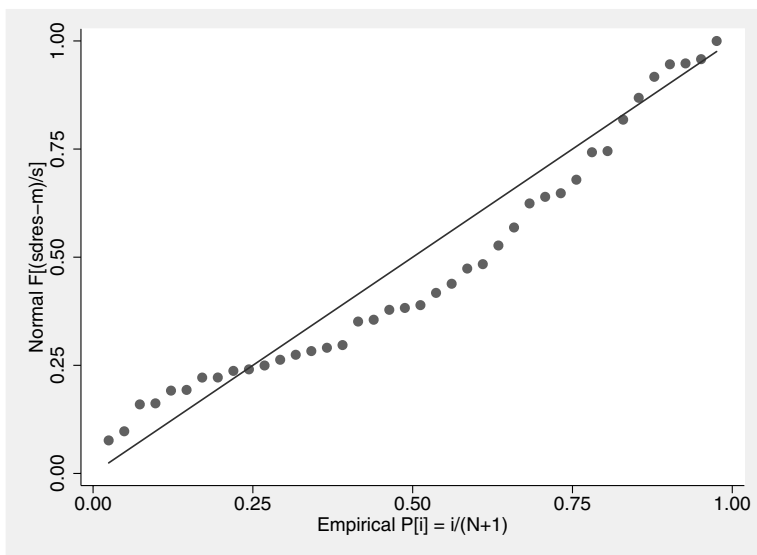


Figure 3.8: Normal probability plot of standardized residuals.

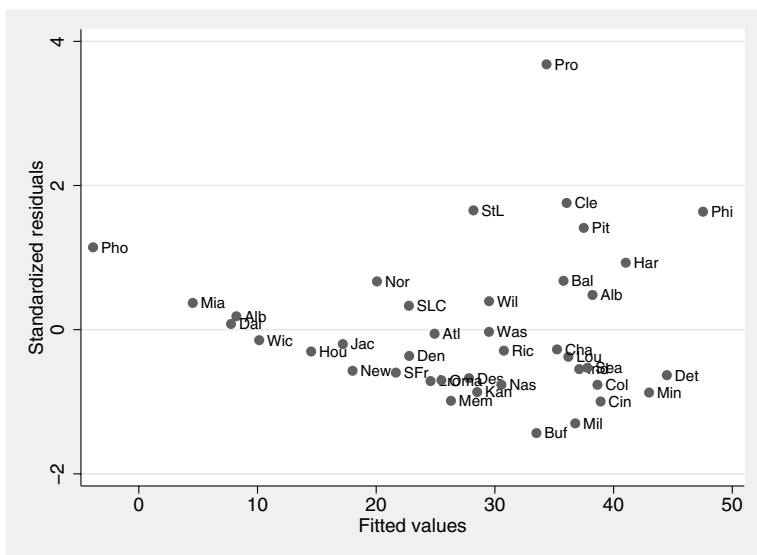


Figure 3.9: Standardized residuals against predicted values.

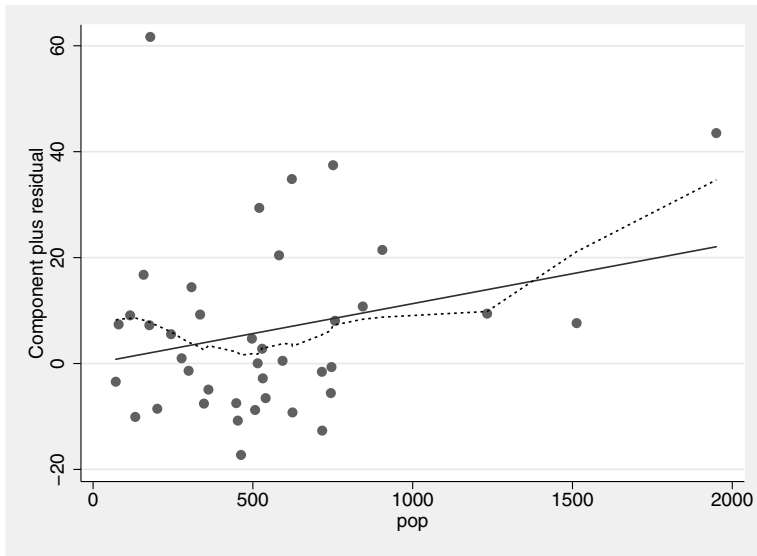


Figure 3.10: Partial residual plot for population.

The partial residual plots are obtained using the `cprplot` command. For the four explanatory variables in the selected model for the pollution data, the required plots are obtained as follows:

```
foreach x in pop temp wind precip {
  cprplot `x`, lowess
  more
}
```

The `lowess` option produces a locally weighted regression curve or *lowess*. The resulting graphs are shown in Figures 3.10 to 3.13. The graphs have to be examined for nonlinearities and for assessing whether the regression line, which has slope equal to the estimated effect of the corresponding explanatory variable in the chosen model, fits the data adequately. The added lowess curve is generally helpful for both. None of the four graphs gives any obvious indication of nonlinearity.

The Cook's distances are found by again using the `predict` instruction; the following calculates these statistics for the chosen model for the pollution data and lists the observations where the statistic is greater than $4/40$ ($4/n$), which is usually the value regarded as indicating possible problems.

```
predict cook, cooksdata
list town so2 cook if cook>4/40
```

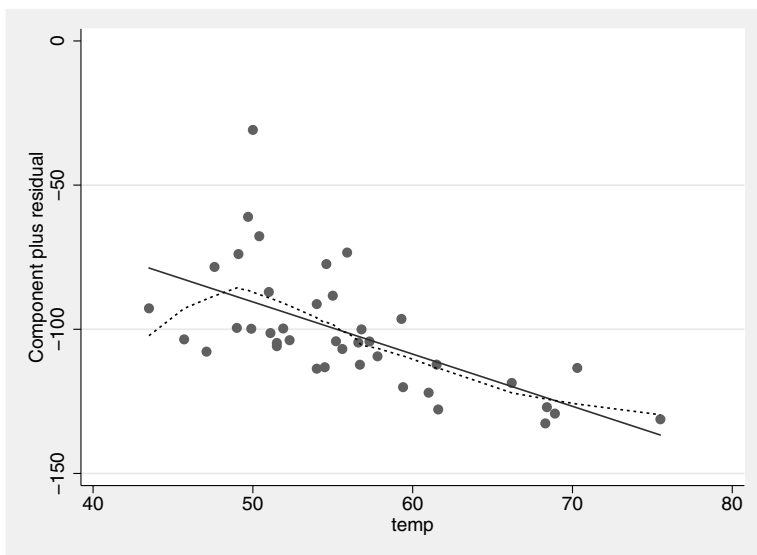


Figure 3.11: Partial residual plot for temperature.

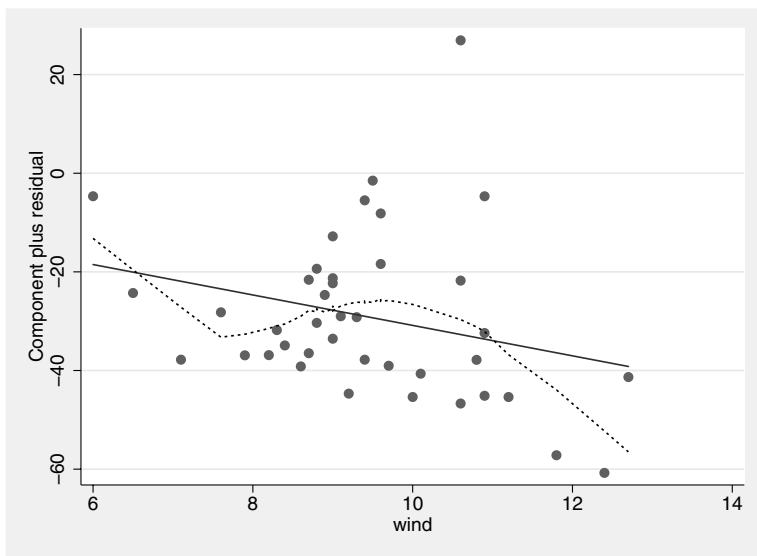


Figure 3.12: Partial residual plot for wind speed.

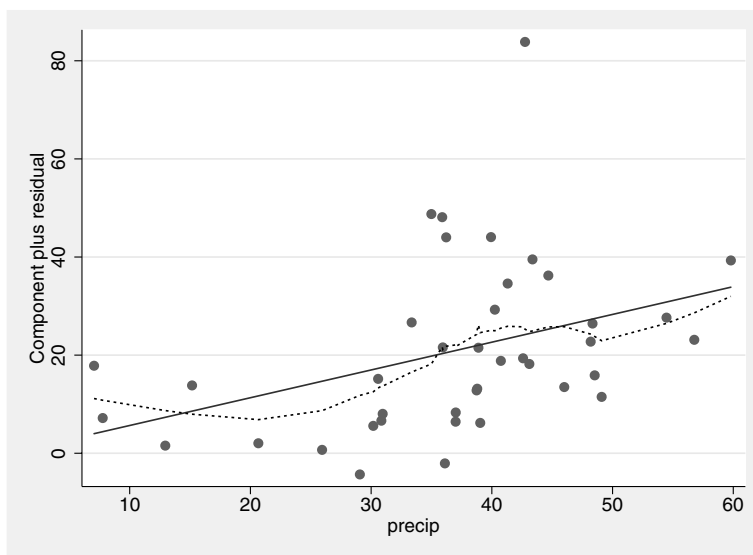


Figure 3.13: Partial residual plot for precipitation.

	town	so2	cook
1.	Phoenix	10	.2543286
28.	Philad	69	.3686437
30.	Provid	94	.2839324

The first instruction stores the Cook's distance statistics in the variable `cook`, and the second lists details of those observations for which the statistic is above the suggested cut-off point.

There are three influential observations. Several of the diagnostic procedures used previously also suggest these observations as possibly giving rise to problems, and some consideration should be given to repeating the analyses with these three observations removed in addition to the initial removal of *Chicago*.

3.4 Exercises

1. Repeat the analyses described in this chapter after removing the three possible outlying observations identified by Cook's distances.
2. The solution to the high correlation between the variables `manuf` and `pop` adopted in the chapter was simply to remove the for-

mer. Investigate other possibilities such as defining a new variable `manuf/pop` in addition to `pop` to be used in the regression analysis.

3. Consider the possibility of taking a transformation of sulphur dioxide pollution before undertaking any regression analyses. For example, try a log transformation.
4. Explore the use of the many other diagnostic procedures available with the `regress` procedure.

See also Exercises in [Chapter 14](#).

Chapter 4

Analysis of Variance I: Treating Hypertension

4.1 Description of data

Maxwell and Delaney (1990) describe a study in which the effects of three possible treatments for hypertension were investigated. The details of the treatments are as follows:

Treatment	Description	Levels
drug	medication	drug X, drug Y, drug Z
biofeed	biofeedback	present, absent
diet	special diet	present, absent

All 12 combinations of the three treatments were included in a $3 \times 2 \times 2$ design. Seventy-two subjects suffering from hypertension were recruited, and six were allocated randomly to each combination of treatments. Blood pressure measurements were made on each subject leading to the data shown in [Table 4.1](#). Questions of interest concern differences in mean blood pressure for the different levels of the three treatments and the effects of interactions between the treatments on blood pressure.

4.2 Analysis of variance model

A suitable model for these data is

Table 4.1 Data in bp.raw

Biofeedback			No Biofeedback		
drug X	drug Y	drug Z	drug X	drug Y	drug Z
Diet absent					
170	186	180	173	189	202
175	194	187	194	194	228
165	201	199	197	217	190
180	215	170	190	206	206
160	219	204	176	199	224
158	209	194	198	195	204
Diet present					
161	164	162	164	171	205
173	166	184	190	173	199
157	159	183	169	196	170
152	182	156	164	199	160
181	187	180	176	180	179
190	174	173	175	203	179

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + \epsilon_{ijkl} \quad (4.1)$$

where y_{ijkl} represents the blood pressure of the l th subject for the i th drug, the j th level of biofeedback, and the k th level of diet, μ is the overall mean, α_i , β_j , and γ_k are the main effects for drugs, biofeedback, and diets, $(\alpha\beta)_{ij}$, $(\alpha\gamma)_{ik}$, and $(\beta\gamma)_{jk}$ are the first-order interaction terms, $(\alpha\beta\gamma)_{ijk}$ is a second-order interaction term, and ϵ_{ijkl} are the residual or error terms assumed to be normally distributed with zero mean and variance σ^2 .

To identify the model, some constraints have to be imposed on the parameters. The standard constraints are:

$$\sum_i \alpha_i = \sum_j \beta_j = \sum_k \gamma_k = 0,$$

$$\begin{aligned} \sum_i (\alpha\beta)_{ij} &= \sum_j (\alpha\beta)_{ij} = \sum_i (\alpha\gamma)_{ik} = \sum_k (\alpha\gamma)_{ik} \\ &= \sum_j (\beta\gamma)_{jk} = \sum_k (\beta\gamma)_{jk} = 0, \end{aligned}$$

and

$$\sum_i (\alpha\beta\gamma)_{ijk} = \sum_j (\alpha\beta\gamma)_{ijk} = \sum_k (\alpha\beta\gamma)_{ijk} = 0.$$

We can test the following null hypotheses:

$$H_0^{(1)} : \text{No drug effect} : \alpha_1 = \alpha_2 = \alpha_3 = 0$$

$$H_0^{(2)} : \text{No biofeedback effect} : \beta_1 = \beta_2 = 0$$

$$H_0^{(3)} : \text{No diet effect} : \gamma_1 = \gamma_2 = 0$$

$$H_0^{(4)} : \text{No drug by biofeedback interaction} : \\ (\alpha\beta)_{ij} = 0, \quad i = 1, 2, 3; \quad j = 1, 2$$

$$H_0^{(5)} : \text{No drug by diet interaction} : (\alpha\gamma)_{ik} = 0, \quad i = 1, 2, 3; \quad k = 1, 2$$

$$H_0^{(6)} : \text{No biofeedback by diet interaction} : \\ (\beta\gamma)_{jk} = 0 \quad j = 1, 2; \quad k = 1, 2$$

$$H_0^{(7)} : \text{No drug by biofeedback by diet interaction} : \\ (\alpha\beta\gamma)_{ijk} = 0 \quad i = 1, 2, 3; \quad j = 1, 2; \quad k = 1, 2$$

Since there are an equal number of observations in each cell of [Table 4.1](#), the total variation in the responses can be partitioned into non-overlapping parts (an orthogonal partition) representing main effects and interactions and residual variation suitable for constructing F -tests for each hypothesis described above. Details can be found in Everitt (2001).

4.3 Analysis using Stata

Assuming the data are in an ASCII file `bp.raw`, exactly as shown in Table 4.1, i.e., 12 rows, the first containing the observations 170 186 180 173 189 202, they can be read into Stata by producing a dictionary file `bp.dct` containing the following statements:

```
dictionary using bp.raw {
    _column(6) int bp11
    _column(14) int bp12
    _column(22) int bp13
    _column(30) int bp01
}
```



```

    _column(38) int bp02
    _column(46) int bp03
}

```

and using the following command

```
infile using bp
```

Note that it was not necessary to define a dictionary here since the same result could have been achieved using a simple `infile varlist` command (see exercises). Here the variable names end on two digits, the first standing for the levels of biofeedback (1: present, 0: absent), and the second for the levels of drug (1,2,3 for X,Y,Z). The final dataset should have a single variable, `bp`, that contains all the blood pressures, and three additional variables, `drug`, `biofeed`, and `diet`, representing the corresponding levels of drug, biofeedback, and diet.

First, create `diet` which should take on one value for the first six rows and another for the following rows. This is achieved using the commands

```

gen diet=0 if _n<=6
replace diet=1 if _n>6

```

or, more concisely, using

```
gen diet=_n>6
```

Now use the `reshape long` command to stack the columns on top of each other. If we specify `bp0` and `bp1` as the variable names in the `reshape` command, then `bp01`, `bp02`, and `bp03` are stacked into one column with variable name `bp0` (and similarly for `bp1`) and another variable is created that contains the suffixes 1, 2, and 3. We ask for this latter variable to be called `drug` using the option `j(drug)` as follows:

```

gen id=_n
reshape long bp0 bp1, i(id) j(drug)
list in 1/9

```

(see Display 4.1). Here, `id` was generated because we needed to specify the row indicator in the `i()` option.

We now need to run the `reshape long` command again to stack up the columns `bp0` and `bp1` and generate the variable `biofeed`. The instructions to achieve this and to label all the variables are given below.

```

replace id=_n
reshape long bp, i(id) j(biofeed)
replace id=_n

label drop _all

```

	id	drug	bp1	bp0	diet
1.	1	1	170	173	0
2.	1	2	186	189	0
3.	1	3	180	202	0
4.	2	1	175	194	0
5.	2	2	194	194	0
6.	2	3	187	228	0
7.	3	1	165	197	0
8.	3	2	201	217	0
9.	3	3	199	190	0

Display 4.1

```
label define d 0 "absent" 1 "present"
label values diet d
label values biofeed d
label define dr 1 "Drug X" 2 "Drug Y" 3 "Drug Z"
label values drug dr
```

To begin, it will be helpful to look at some summary statistics for each of the cells of the design. A simple way of obtaining the required summary measures is to use the `table` instruction.

```
table drug, contents(freq mean bp median bp sd bp)/*
    */ by(diet biofeed)
```

The standard deviations in Display 4.2 indicate that there are considerable differences in the within cell variability. This may have implications for the analysis of variance of these data: one of the assumptions made is that the observations within each cell have the same population variance. To begin, however, we will apply the model specified in Section 3.2 to the raw data using the `anova` instruction

```
anova bp drug diet biofeed diet*drug diet*biofeed /*
    */ drug*biofeed drug*diet*biofeed
```

The resulting ANOVA table is shown in Display 4.3.

The `Root MSE` is simply the square root of the residual mean square, with `R-squared` and `Adj R-squared` being as described in [Chapter 3](#). The F -statistic of each effect represents the mean sum of squares for that effect, divided by the residual mean sum of squares, given under the heading `MS`. There are highly significant main effects of `drug` ($F_{2,60} = 11.73$, $p < 0.001$), `diet` ($F_{1,60} = 33.20$, $p < 0.001$), and `biofeed` ($F_{1,60} = 13.07$, $p < 0.001$). The two-way interactions are not significant at the 5% level but the three-way interaction `drug by diet`

diet, biofeed and drug	Freq.	mean(bp)	med(bp)	sd(bp)
absent absent				
Drug X	6	188	192	10.86278
Drug Y	6	200	197	10.07968
Drug Z	6	209	205	14.3527
absent present				
Drug X	6	168	167.5	8.602325
Drug Y	6	204	205	12.68069
Drug Z	6	189	190.5	12.61745
present absent				
Drug X	6	173	172	9.797959
Drug Y	6	187	188	14.01428
Drug Z	6	182	179	17.1114
present present				
Drug X	6	169	167	14.81891
Drug Y	6	172	170	10.93618
Drug Z	6	173	176.5	11.6619

Display 4.2

	Number of obs = 72		R-squared = 0.5840		
	Root MSE = 12.5167		Adj R-squared = 0.5077		
Source	Partial SS	df	MS	F	Prob > F
Model	13194	11	1199.45455	7.66	0.0000
drug	3675	2	1837.5	11.73	0.0001
diet	5202	1	5202	33.20	0.0000
biofeed	2048	1	2048	13.07	0.0006
diet*drug	903	2	451.5	2.88	0.0638
diet*biofeed	32	1	32	0.20	0.6529
drug*biofeed	259	2	129.5	0.83	0.4425
drug*diet*biofeed	1075	2	537.5	3.43	0.0388
Residual	9400	60	156.666667		
Total	22594	71	318.225352		

Display 4.3

by **biofeed** is ($F_{2,60} = 3.43$, $p = 0.04$). The existence of a three-way interaction complicates the interpretation of the other terms in the model; it implies that the interaction between any two of the factors is different at the different levels of the third factor. Perhaps the best way of trying to understand the meaning of the three-way interaction is to plot a number of *interaction diagrams*; that is, plots of mean values for a factor at the different levels of the other factors.

This can be done by first creating a variable **predbp** containing the predicted means (which in this case coincide with the observed cell means because the model fitted is saturated, i.e., the number of parameters is equal to the number of cell means) using the command

```
predict predbp
```

Plots of **predbp** against **biofeed** for each level of **drug** with separate lines for **diet** can be obtained using the instruction

```
twoway (line predbp biofeed if diet==0)          /*
      /* (line predbp biofeed if diet==1) , by(drug) /*
      /* xlabel(0 "no biofeed." 1 "biofeed.")      /*
      /* ylabel(170 190 210) xtitle(" ")          /*
      /* legend(order(1 "no diet" 2 "diet"))
```

The resulting interaction diagrams are shown in [Figure 4.1](#). For drug Y, the presence of biofeedback increases the effect of diet (the vertical distance between the solid and dashed lines), whereas for drug Z the effect of diet is hardly altered by the presence of biofeedback and for drug X the effect is decreased.

Tables of the cell means plotted in the interaction diagrams, as well as the corresponding standard deviations, are produced for each drug using the following command:

```
table diet biofeed, contents(mean bp sd bp) by(drug)
```

giving the output shown in Display 4.4.

As mentioned previously, the observations in the 12 cells of the $3 \times 2 \times 2$ design have variances that differ considerably. Consequently, an analysis of variance of the data transformed in some way might be worth considering. For example, to analyze the log transformed observations, we can use the following instructions:

```
gen lbp=log(bp)
anova lbp drug diet biofeed diet*drug diet*biofeed /*
      /* drug*biofeed drug*diet*biofeed
```

The resulting analysis of variance table is shown in Display 4.5.

The results are similar to those for the untransformed blood pressures. The three-way interaction is only marginally significant. If no

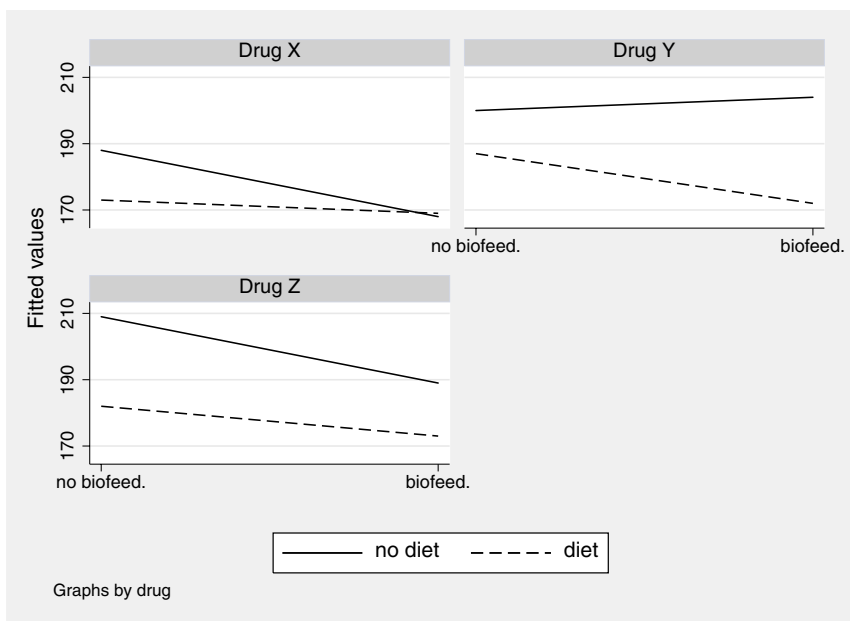


Figure 4.1: Interaction diagrams showing the interaction between diet and biofeedback for each level of drug.

drug and diet	biofeed	
	absent	present
Drug X		
	absent	188 168 10.86278 8.602325
	present	173 169 9.797959 14.81891
Drug Y		
	absent	200 204 10.07968 12.68069
	present	187 172 14.01428 10.93618
Drug Z		
	absent	209 189 14.3527 12.61745
	present	182 173 17.1114 11.6619

	Number of obs =	72	R-squared =	0.5776	
	Root MSE =	.068013	Adj R-squared =	0.5002	
Source	Partial SS	df	MS	F	Prob > F
Model	.379534762	11	.03450316	7.46	0.0000
diet	.149561559	1	.149561559	32.33	0.0000
drug	.107061236	2	.053530618	11.57	0.0001
biofeed	.061475507	1	.061475507	13.29	0.0006
diet*drug	.024011594	2	.012005797	2.60	0.0830
diet*biofeed	.000657678	1	.000657678	0.14	0.7075
drug*biofeed	.006467873	2	.003233936	0.70	0.5010
diet*drug*biofeed	.030299315	2	.015149657	3.28	0.0447
Residual	.277545987	60	.004625766		
Total	.657080749	71	.009254658		

Display 4.5

substantive explanation of this interaction is available, it might be better to interpret the results in terms of the very significant main effects. The relevant summary statistics for the log transformed blood pressures can be obtained using the following instructions:

```
table drug, contents(mean lbp sd lbp)
```

```
table diet, contents(mean lbp sd lbp)
```

```
table biofeed, contents(mean lbp sd lbp)
```

giving the tables in Displays 4.6 to 4.8.

drug	mean(lbp)	sd(lbp)
Drug X	5.159152	.075955
Drug Y	5.247087	.0903675
Drug Z	5.232984	.0998921

Display 4.6

Drug X appears to produce lower blood pressures as does the special diet and the presence of biofeedback. Readers are encouraged to try other transformations.

diet	mean(lbp)	sd(lbp)
absent	5.258651	.0915982
present	5.167498	.0781686

Display 4.7

biofeed	mean(lbp)	sd(lbp)
absent	5.242295	.0890136
present	5.183854	.0953618

Display 4.8

Note that it is easy to estimate the model with main effects only using regression with dummy variables. Since **drug** has three levels and therefore requires two dummy variables, we save some time by using the **xi:** prefix as follows:

```
xi: regress lbp i.drug i.diet i.biofeed
```

leading to the results shown in Display 4.9. The coefficients represent the mean differences between each level compared with the reference level (the omitted categories: drug X, diet absent, and biofeedback absent) when the other variables are held constant. The *p*-values are equal to those of ANOVA except that no overall *p*-value for **drug** is given. This can be obtained using

```
testparm _Idrug*
```

```
( 1)  _Idrug_2 = 0
( 2)  _Idrug_3 = 0
      F( 2,    67) =   10.58
      Prob > F =   0.0001
```

The *F*-statistic is different from that in the last **anova** command because no interactions were included in the model; hence the residual degrees of freedom and the residual sum of squares have both increased.

i.drug	_Idrug_1-3	(naturally coded; _Idrug_1 omitted)				
i.diet	_Idiet_0-1	(naturally coded; _Idiet_0 omitted)				
i.biofeed	_Ibiofeed_0-1	(naturally coded; _Ibiofeed_0 omitted)				
Source	SS	df	MS	Number of obs = 72		
Model	.318098302	4	.079524576	F(4, 67) = 15.72		
Residual	.338982447	67	.00505944	Prob > F = 0.0000		
Total	.657080749	71	.009254658	R-squared = 0.4841		
				Adj R-squared = 0.4533		
				Root MSE = .07113		
lbp	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
_Idrug_2	.0879354	.0205334	4.28	0.000	.0469506	.1289203
_Idrug_3	.0738315	.0205334	3.60	0.001	.0328467	.1148163
_Idiet_1	-.0911536	.0167654	-5.44	0.000	-.1246175	-.0576896
_Ibiofeed_1	-.0584406	.0167654	-3.49	0.001	-.0919046	-.0249767
_cons	5.233949	.0187443	279.23	0.000	5.196535	5.271363

Display 4.9

4.4 Exercises

1. Reproduce the result of the command `infile` using `bp` without using the dictionary, and follow the `reshape` instructions to generate the required dataset.
2. Produce three diagrams with boxplots of blood pressure (1) for each level of drug, (2) for each level of diet, and (3) for each level of biofeedback.
3. Investigate other possible transformations of the data and compare the resulting analyses of variance with those given in the text.
4. Suppose that in addition to the blood pressure of each of the individuals in the study, the investigator had also recorded their ages in file `age.dat` as shown in [Table 4.2](#). Reanalyze the data using age as a covariate (see `help merge` and `help anova`).

Table 4.2 Data in
age.dat

id	age	id	age
1	39	37	45
2	39	38	58
3	61	39	61
4	50	40	47
5	51	41	67
6	43	42	49
7	59	43	54
8	50	44	48
9	47	45	46
10	60	46	67
11	77	47	56
12	57	48	54
13	62	49	66
14	44	50	43
15	63	51	47
16	77	52	35
17	56	53	50
18	62	54	60
19	44	55	73
20	61	56	46
21	66	57	59
22	52	58	65
23	53	59	49
24	54	60	52
25	40	61	40
26	62	62	80
27	68	63	46
28	63	64	63
29	47	65	56
30	70	66	58
31	57	67	53
32	51	68	56
33	70	69	64
34	57	70	57
35	64	71	60
36	66	72	48

Chapter 5

Analysis of Variance II: Effectiveness of Slimming Clinics

5.1 Description of data

Slimming clinics aim to help people lose weight by offering encouragement and support about dieting through regular meetings. In a study of their effectiveness, a 2×2 factorial design was used to investigate whether giving clients a technical manual containing slimming advice based on psychological behaviorist theory would help them to control their diet, and how this might be affected by whether a client had already been trying to slim. The data collected are shown in [Table 5.1](#). (They are also given in Hand *et al.*, 1994.) The response variable was defined as follows:

$$\frac{\text{weight after three months of treatment} - \text{ideal weight}}{\text{initial weight} - \text{ideal weight}} \quad (5.1)$$

The number of observations in each cell of the design is not the same, so this is an example of an *unbalanced* 2×2 design.

5.2 Analysis of variance model

A suitable analysis of variance model for the data is

Table 5.1 Data in `slim.dat`

cond	status	resp	cond	status	resp
1	1	-14.67	1	1	-1.85
1	1	-8.55	1	1	-23.03
1	1	11.61	1	2	0.81
1	2	2.38	1	2	2.74
1	2	3.36	1	2	2.10
1	2	-0.83	1	2	-3.05
1	2	-5.98	1	2	-3.64
1	2	-7.38	1	2	-3.60
1	2	-0.94	2	1	-3.39
2	1	-4.00	2	1	-2.31
2	1	-3.60	2	1	-7.69
2	1	-13.92	2	1	-7.64
2	1	-7.59	2	1	-1.62
2	1	-12.21	2	1	-8.85
2	2	5.84	2	2	1.71
2	2	-4.10	2	2	-5.19
2	2	0.00	2	2	-2.80

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon_{ijk} \quad (5.2)$$

where y_{ijk} represents the weight change of the k th individual having status j and condition i , μ is the overall mean, α_i represents the effect of condition i , β_j the effect of status j , γ_{ij} the status \times condition interaction, and ϵ_{ijk} the errors – these are assumed to have a normal distribution with mean zero and variance σ^2 .

The unbalanced nature of the slimming data presents some difficulties for analysis not encountered in factorial designs having the same number of observations in each cell (see the previous chapter). The main problem is that when the data are unbalanced there is no unique way of finding a ‘sum of squares’ corresponding to each main effect and their interactions, because these effects are no longer independent of one another. If the data were balanced, the among cells sum of squares would partition orthogonally into three component sums of squares representing the two main effects and their interaction. Several methods have been proposed for dealing with this problem and each leads to a different partition of the overall sum of squares. The different methods for arriving at the sums of squares for unbalanced designs can be explained in terms of the comparisons of different sets of specific models. For a design with two factors A and B, Stata can calculate the following types of sums of squares:

5.2.1 Sequential sums of squares

Sequential sums of squares (also known as hierarchical sums of squares) represent the effect of adding a term to an existing model. So, for example, a set of sequential sums of squares such as

Source	SS
A	SS(A)
B	SS(B A)
AB	SS(AB A,B)

represent a comparison of the following models:

- SS(AB|A,B)—model including an interaction and main effects compared with one including only main effects.
- SS(B|A)—model including both main effects, but with no interaction, compared with one including only the main effects of factor A.
- SS(A)—model containing only the A main effect compared with one containing only the overall mean.

The use of these sums of squares in a series of tables in which the effects are considered in different orders (see later) will often provide the most satisfactory way of deciding which model is most appropriate for the observations. (These are SAS Type I sums of squares—see Everitt and Der, 2001.)

5.2.2 Unique sums of squares

By default, Stata produces unique sums of squares that represent the contribution of each term to a model including all the other terms. So, for a two-factor design, the sums of squares represent the following.

Source	SS
A	SS(A B,AB)
B	SS(B A,AB)
AB	SS(AB A,B)

(These are SAS Type III sums of squares.) Note that these sums of squares generally do not add up to the total sums of squares.

5.2.3 Regression

As we have shown in [Chapter 4](#), ANOVA models may also be estimated using regression by defining suitable dummy variables. Assume that A is represented by a single dummy variable. The regression coefficients for A represents the *partial* contribution of that variable, adjusted for

all other variables in the model, say B. This is equivalent to the contribution of A to a model already including B. A complication with regression models is that, in the presence of an interaction, the p -values of the terms depend on the exact coding of the dummy variables (see Aitkin, 1978). The unique sums of squares correspond to regression where dummy variables are coded in a particular way, for example a two-level factor can be coded as $-1, 1$.

There have been numerous discussions over which sums of squares are most appropriate for the analysis of unbalanced designs. The Stata manual appears to recommend its default for general use. Nelder (1977) and Aitkin (1978) however, are strongly critical of ‘correcting’ main effects for an interaction term involving the same factor; their criticisms are based on both theoretical and pragmatic arguments and seem compelling. A frequently used approach is therefore to test the highest order interaction adjusting for all lower order interactions and not vice versa. Both Nelder and Aitkin prefer the use of Type I sums of squares in association with different orders of effects as the procedure most likely to identify an appropriate model for a data set. For a detailed explanation of the various types of sums of squares, see Boniface (1995).

5.3 Analysis using Stata

The data can be read in from an ASCII file `slim.dat` in the usual way using

```
infile cond status resp using slim.dat
```

A table showing the unbalanced nature of the 2×2 design can be obtained from

```
tabulate cond status
```

cond	status		Total
	1	2	
1	5	12	17
2	11	6	17
Total	16	18	34

We now use the `anova` command with no options specified to obtain the unique (Type III) sums of squares:

```
anova resp cond status cond*status
```

(see Display 5.1).

<div> <div>Number of obs = 34</div> <div>R-squared = 0.2103</div> <div>Root MSE = 5.9968</div> <div>Adj R-squared = 0.1313</div> </div>					
Source	Partial SS	df	MS	F	Prob > F
Model	287.231861	3	95.7439537	2.66	0.0659
cond	2.19850409	1	2.19850409	0.06	0.8064
status	265.871053	1	265.871053	7.39	0.0108
cond*status	.130318264	1	.130318264	0.00	0.9524
Residual	1078.84812	30	35.961604		
Total	1366.07998	33	41.3963631		

Display 5.1

Our recommendation is that the sums of squares shown in this table are *not* used to draw inferences because the main effects have been adjusted for the interaction.

Instead we prefer an analysis that consists of obtaining two sets of sequential sums of squares, the first using the order `cond status cond*status` and the second the order `status cond cond*status`. The necessary instructions are

```
anova resp cond status cond*status, sequential
(see Display 5.2).
```

<div> <div>Number of obs = 34</div> <div>R-squared = 0.2103</div> <div>Root MSE = 5.9968</div> <div>Adj R-squared = 0.1313</div> </div>					
Source	Seq. SS	df	MS	F	Prob > F
Model	287.231861	3	95.7439537	2.66	0.0659
cond	21.1878098	1	21.1878098	0.59	0.4487
status	265.913733	1	265.913733	7.39	0.0108
cond*status	.130318264	1	.130318264	0.00	0.9524
Residual	1078.84812	30	35.961604		
Total	1366.07998	33	41.3963631		

Display 5.2

```
anova resp status cond cond*status, sequential
(see Display 5.3). The sums of squares corresponding to model and
```

	Number of obs = 34		R-squared = 0.2103		
	Root MSE = 5.9968		Adj R-squared = 0.1313		
Source	Seq. SS	df	MS	F	Prob > F
Model	287.231861	3	95.7439537	2.66	0.0659
status	284.971071	1	284.971071	7.92	0.0085
cond	2.13047169	1	2.13047169	0.06	0.8094
cond*status	.130318264	1	.130318264	0.00	0.9524
Residual	1078.84812	30	35.961604		
Total	1366.07998	33	41.3963631		

Display 5.3

residuals are, of course, the same in both tables, as is the sum of squares for the interaction term. What differ are the sums of squares in the **cond** and **status** rows in the two tables. The terms of most interest are the sum of squares of **status|cond** which is obtained from the table as 265.91, and the sum of squares of **cond|status** which is 2.13. These sums of squares are less than the sums of squares for **status** and **cond** alone (284.97 and 21.19, respectively), by an amount of 19.06, a portion of the sums of squares which cannot be uniquely attributed to either of the variables. The associated *F*-tests in the two tables make it clear that there is no interaction effect and that **status|cond** is significant but **cond|status** is not. The conclusion is that only **status**, i.e., whether the woman had been slimming for over one year, is important in determining weight change. Provision of the manual appears to have no discernible effect.

Results equivalent to the unique (Type III) sums of squares can be obtained using regression:

```
gen cond1=cond
recode cond1 1=-1 2=1
gen status1=status
recode status1 1=-1 2=1
gen statcond = cond1*status1
regress res cond1 status1 statcond
```

(see Display 5.4). The *p*-values agree with those based on unique sums of squares. However, these results differ from the regression used by Stata's **anova** with the option **regress**:

```
anova resp cond status cond*status, regress
```

(see Display 5.5) because this uses different dummy variables. The

Source	SS	df	MS	Number of obs = 34		
Model	287.231861	3	95.7439537	F(3, 30) = 2.66		
Residual	1078.84812	30	35.961604	Prob > F = 0.0659		
				R-squared = 0.2103		
				Adj R-squared = 0.1313		
Total	1366.07998	33	41.3963631	Root MSE = 5.9968		

resp	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
cond1	.2726251	1.102609	0.25	0.806	-1.979204	2.524454
status1	2.998042	1.102609	2.72	0.011	.746213	5.24987
statcond	-.066375	1.102609	-0.06	0.952	-2.318204	2.185454
_cons	-3.960958	1.102609	-3.59	0.001	-6.212787	-1.70913

Display 5.4

Source	SS	df	MS	Number of obs = 34		
Model	287.231861	3	95.7439537	F(3, 30) = 2.66		
Residual	1078.84812	30	35.961604	Prob > F = 0.0659		
				R-squared = 0.2103		
				Adj R-squared = 0.1313		
Total	1366.07998	33	41.3963631	Root MSE = 5.9968		

resp	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
_cons	-.7566666	2.448183	-0.31	0.759	-5.756524	4.24319
cond						
1	-.4125001	2.9984	-0.14	0.891	-6.536049	5.711049
2	(dropped)					
status						
1	-5.863333	3.043491	-1.93	0.064	-12.07897	.3523044
2	(dropped)					
cond*status						
1 1	-.2655002	4.410437	-0.06	0.952	-9.272815	8.741815
1 2	(dropped)					
2 1	(dropped)					
2 2	(dropped)					

Display 5.5

dummy variables are equal to 1 for the levels shown on the left of the reported coefficient and 0 otherwise, i.e., the dummy variable for `cond*status` is 1 when `status` and `cond` are both 1.

A table of mean values helpful in interpreting these results can be found using

```
table cond status, c(mean resp) row col f(%8.2f)
```

cond	status		
	1	2	Total
1	-7.30	-1.17	-2.97
2	-6.62	-0.76	-4.55
Total	-6.83	-1.03	-3.76

The means demonstrate that experienced slimmers achieve the greatest weight reduction.

5.4 Exercises

1. Investigate what happens to the sequential sums of squares if the `cond*status` interaction term is given before the main effects `cond` `status` in the `anova` command with the `sequential` option.
2. Use `regress` to reproduce the analysis of variance by coding both condition and status as (0,1) dummy variables and creating an interaction variable as the product of these dummy variables.
3. Use `regress` in conjunction with `xi:` to fit the same model without the need to generate any dummy variables.
4. Reproduce the results of `anova resp cond status cond*status, regress` using `regress` by making `xi:` omit the last category instead of the first (see `help xi`, under ‘Summary of controlling the omitted dummy’).

See also the Exercises in [Chapters 7](#) and [13](#).

Chapter 6

Logistic Regression: Treatment of Lung Cancer and Diagnosis of Heart Attacks

6.1 Description of data

Two datasets will be analyzed in this chapter. The first dataset shown in Table 6.1 originates from a clinical trial in which lung cancer patients were randomized to receive two different kinds of chemotherapy (sequential therapy and alternating therapy). The outcome was classified into one of four categories: progressive disease, no change, partial remission, or complete remission. The data were published in Holtbrugge and Schumacher (1991) and also appear in Hand *et al.* (1994). The central question is whether there is any evidence of a difference in the outcomes achieved by the two types of therapy.

Table 6.1 Lung cancer data in tumor.dat

Therapy	Sex	Progressive disease	No change	Partial remission	Complete remission
Sequential	Male	28	45	29	26
	Female	4	12	5	2
Alternative	Male	41	44	20	20
	Female	12	7	3	1

Table 6.2 Data in `sck.dat`

Maximum CK level	Infarct present	Infarct absent
0 – 39	2	88
40 – 79	13	26
80 – 119	30	8
120 – 159	30	5
160 – 199	21	0
200 – 239	19	1
240 – 279	18	1
280 – 319	13	1
320 – 359	19	0
360 – 399	15	0
400 – 439	7	0
440 – 479	8	0
480 –	35	0

The second dataset to be used in this chapter arises from a study investigating the use of serum creatine kinase (CK) levels for the diagnosis of myocardial infarction (heart attack). Patients admitted to a coronary care unit because they were suspected of having had a myocardial infarction within the last 48 hours had their CK levels measured on admission and the next two mornings. A clinician who was ‘blind’ to the CK results came to an independent ‘gold standard’ diagnosis using electrocardiograms, clinical records, and autopsy reports. The maximum CK levels for 360 patients are given in Table 6.2 together with the clinician’s diagnosis. The table was taken from Sackett *et al.* (1991) (with permission of the publisher, Little Brown & Company), where only the ranges of CK levels were given, not their precise values.

The main questions of interest for this second dataset are how well CK discriminates between those with and without myocardial infarction, and how diagnostic tests perform that are based on applying different thresholds to CK.

6.2 The logistic regression model

In this chapter we discuss the analysis of dichotomous (or binary) responses as well as ordinal responses. Dichotomous responses arise when the outcome is presence or absence of a characteristic or event, for example myocardial infarction in the second dataset.

What we would like to do is to investigate the effects of a number of explanatory variables on our binary or ordinal response variable. This appears to be the same aim as for multiple regression discussed in

Chapter 3, where the model for a response y_i and explanatory variables x_{1i} to x_{pi} can be written as

$$E(y_i|\mathbf{x}_i) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}. \quad (6.1)$$

Binary responses are typically coded 1 for the event of interest, such as infarct present, and 0 for the opposite event. In this case the expected value, π_i , is simply the probability that the event of interest occurs. This raises the first problem for applying the model above to a binary response variable, namely that

1. the predicted probability must satisfy $0 \leq \pi_i \leq 1$ whereas a linear predictor can yield any value from minus infinity to plus infinity

A second problem with the model is that

2. the observed values of y_i do not follow a normal distribution with mean π_i , but rather a Bernoulli (or Binomial($1, \pi_i$)) distribution.

Consequently a new approach is needed, and this is provided by logistic regression.

In logistic regression, the first problem is addressed by replacing the probability $\pi_i = E(y_i|\mathbf{x}_i)$ on the left-hand side of equation (6.1) by the *logit* of the probability, giving

$$\text{logit}(\pi_i) = \log(\pi_i/(1 - \pi_i)) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi}. \quad (6.2)$$

The logit of the probability is simply the log of the odds of the event of interest. Writing $\boldsymbol{\beta}$ and \mathbf{x}_i for the column vectors $(\beta_0, \beta_1, \dots, \beta_p)'$ and $(1, x_{1i}, \dots, x_{pi})'$, respectively, the predicted probability as a function of the linear predictor is

$$\begin{aligned} \pi_i &= \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}'\mathbf{x}_i)} \\ &= \frac{1}{1 + \exp(-\boldsymbol{\beta}'\mathbf{x}_i)}. \end{aligned} \quad (6.3)$$

When the logit takes on any real value, this probability always satisfies $0 \leq \pi_i \leq 1$.

The second problem relates to the estimation procedure. Whereas maximum likelihood estimation in conventional linear regression leads to least squares, this is not the case in logistic regression. In logistic regression the log likelihood is maximized numerically using an iterative algorithm. For full details of logistic regression, see for example Collett (2002), Agresti (1996), and Long and Freese (2003). (The last

reference provides a comprehensive discussion of regression models for categorical variables using Stata.) Logistic regression can be generalized to the situation where the response variable has more than two ordered response categories. In the latent response formulation, we think of the ordered categories as representing successive intervals of an underlying latent (unobserved) continuous response. If there are S response categories labeled a_1, \dots, a_S , the relationship between the observed and latent response can be formulated as a threshold model:

$$y_i = \begin{cases} a_1 & \text{if } y_i^* \leq \kappa_1 \\ a_2 & \text{if } \kappa_1 < y_i^* \leq \kappa_2 \\ \vdots & \vdots \\ a_S & \text{if } \kappa_{S-1} < y_i^*, \end{cases}$$

where κ_s , $s = 1, \dots, S - 1$ are threshold or cut-point parameters. The latent response is modeled as a linear regression

$$y_i^* = \beta' \mathbf{x}_i + \epsilon_i,$$

where ϵ_i has a logistic distribution,

$$\Pr(\epsilon_i < X) = \frac{\exp(X)}{1 + \exp(X)}.$$

Note that we could not simultaneously estimate the constant in the model for y_i^* and all threshold parameters since we could increase the constant and all thresholds by the same amount without changing the model. In Stata, the constant is therefore set to zero for identification. The latent response and threshold model imply a simple *cumulative model* for the observed responses. The cumulative probability γ_{is} that the response y_i takes on a value up to and including a_s becomes

$$\begin{aligned} \gamma_{is} &= \Pr(y_i^* \leq \kappa_s) = \Pr(y_i^* - \beta' \mathbf{x}_i \leq \kappa_s - \beta' \mathbf{x}_i) \\ &= \frac{\exp(\kappa_s - \beta' \mathbf{x}_i)}{1 + \exp(\kappa_s - \beta' \mathbf{x}_i)}, \quad s = 1, \dots, S - 1. \end{aligned} \quad (6.4)$$

Note that $\gamma_{iS} = 1$. This cumulative ordinal regression model is also called the *proportional odds model* because the log odds that $y_i > a_s$ are

$$\log \left(\frac{1 - \gamma_{is}}{\gamma_{is}} \right) = \beta' \mathbf{x}_i - \kappa_s \quad (6.5)$$

so that the log odds ratio for two units i and j is $\beta'(\mathbf{x}_i - \mathbf{x}_j)$ which is independent of s . Therefore, $\exp(\beta_k)$ represents the odds ratio that $y > a_s$ for any s when x_k increases by one unit if all other covariates remain the same.

In binary logistic regression for dichotomous responses, $a_1 = 0$, $a_2 = 1$, $\kappa_1 = 0$, and $\exp(\beta_k)$ is the odds ratio that $y = 1$ when x_k increases by one unit and all other covariates remain the same. Note that a different identifying restriction is used than for ordinal responses: the threshold is set to zero instead of the constant in the model for y_i^* .

Note that the probit and ordinal probit models correspond to logistic and ordinal logistic regression models with the cumulative distribution function in (6.4) replaced by the standard normal cumulative distribution function. More information on models for ordinal data can be found in Agresti (1996) and Long and Freese (2003).

6.3 Analysis using Stata

6.3.1 Chemotherapy treatment of lung cancer

Assume the ASCII file `tumor.dat` contains the four by four matrix of frequencies shown in [Table 6.1](#). First read the data and generate variables for therapy and sex using the `egen` function `seq()`:

```
infile fr1 fr2 fr3 fr4 using tumor.dat
egen therapy=seq(), from(0) to(1) block(2)
egen sex=seq(), from(1) to(2) by(therapy)
label define t 0 seq 1 alt
label values therapy t
label define s 1 male 2 female
label values sex s
```

`block(2)` causes the number in the sequence (from 0 to 1) to be repeated in blocks of two, whereas `by(therapy)` causes the sequence to start from the lower limit every time the value of `therapy` changes.

We next reshape the data to long, placing the four levels of the outcome into a variable `outc`, and expand the dataset by replicating each observation `freq` times so that we have one observation per subject:

```
reshape long fr, i(therapy sex) j(outc)
expand fr
```

We can check that the data conversion is correct by tabulating these data as in [Table 6.1](#):

```
table sex outc, contents(freq) by(therapy)
giving the table in Display 6.1.
```

therapy and sex	outc			
	1	2	3	4
seq				
male	28	45	29	26
female	4	12	5	2
alt				
male	41	44	20	20
female	12	7	3	1

Display 6.1

To be able to carry out ordinary logistic regression, we need to dichotomize the outcome, for example, by considering partial and complete remission to be an improvement and the other categories to be no improvement. The new outcome variable may be generated as follows:

```
gen improve=outc
recode improve 1/2=0 3/4=1
```

or using

```
gen improve = outc>2
```

The command **logit** for logistic regression behaves the same way as **regress** and all other estimation commands. For example, automatic selection procedures can be carried out using **sw** and post-estimation commands such as **testparm** and **predict** are available. First, include **therapy** as the only explanatory variable:

```
logit improve therapy
```

(see Display 6.2). The algorithm takes three iterations to converge. The coefficient of **therapy** represents the difference in the log odds (of an improvement) between the alternating and sequential therapies. The negative value indicates that sequential therapy is superior to alternating therapy. The *p*-value of the coefficient is 0.041 in the table. This was derived from the Wald-statistic, **z**, which is equal to the coefficient divided by its asymptotic standard error (**Std. Err.**) as derived from the Hessian matrix of the log likelihood function, evaluated at the maximum likelihood solution. This *p*-value is less reliable than the *p*-value based on the likelihood ratio between the model including only the constant and the current model, which is given at the top of the output (**chi2(1)=4.21**). Here, minus twice the log of the likelihood ratio is equal to 4.21 which has an approximate χ^2 -distribution if the simple

```

Iteration 0:  log likelihood = -194.40888
Iteration 1:  log likelihood = -192.30753
Iteration 2:  log likelihood = -192.30471

Logit estimates
Number of obs   =      299
LR chi2(1)      =       4.21
Prob > chi2     =      0.0402
Pseudo R2      =      0.0108

Log likelihood = -192.30471

```

improve	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
therapy	-.4986993	.2443508	-2.04	0.041	-.977618	-.0197805
_cons	-.361502	.1654236	-2.19	0.029	-.6857263	-.0372777

Display 6.2

model is correct with one degree of freedom (because there is one additional parameter) giving a p -value of 0.040, very similar to that based on the Wald test. The coefficient of **therapy** represents the difference in log odds between the therapies and is not easy to interpret apart from the sign. Exponentiating the coefficient gives the odds ratio and exponentiating the 95% confidence limits gives the confidence interval for the odds ratio. Fortunately, the command **logistic** may be used to obtain the required odds ratio and its confidence interval directly (alternatively, we could use the **or** option in the **logit** command):

```
logistic improve therapy
```

(see Display 6.3). The standard error now represents the approximate

```

Logistic regression
Number of obs   =      299
LR chi2(1)      =       4.21
Prob > chi2     =      0.0402
Pseudo R2      =      0.0108

Log likelihood = -192.30471

```

improve	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
therapy	.6073201	.1483991	-2.04	0.041	.3762061	.9804138

Display 6.3

standard error of the odds ratio (calculated using the delta method, see e.g., Agresti, 2002). Since the sampling distribution of the odds ratio is not well approximated by a normal distribution, the Wald statistic

and confidence interval are derived using the log odds and its standard error. To test whether the inclusion of `sex` in the model significantly increases the likelihood, the current likelihood (and all the estimates) can be saved using

```
estimates store model1
Including sex
```

```
logistic improve therapy sex
```

gives the output shown in Display 6.4. The p -value of `sex` based on

Logistic regression		Number of obs	=	299
		LR chi2(2)	=	7.55
		Prob > chi2	=	0.0229
Log likelihood = -190.63171		Pseudo R2	=	0.0194

improve	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
therapy	.6051969	.1486907	-2.04	0.041	.3739084	.9795537
sex	.5197993	.1930918	-1.76	0.078	.2509785	1.076551

Display 6.4

the Wald-statistic is 0.078, and a p -value for the likelihood ratio test is obtained using

```
lrtest model1 .
```

likelihood-ratio test	LR chi2(1)	=	3.35
(Assumption: model1 nested in .)	Prob > chi2	=	0.0674

which is not very different from the value of 0.078. In the `lrtest` command ‘.’ refers to the current model and `model1` is the model excluding `sex` which was previously stored using `estimates store`. We could have specified the models in the reverse order as Stata assumes that the model with the lower log likelihood is nested within the other model. Note that it is essential that both models compared in the likelihood ratio test be based on the same sample. If `sex` had missing values, the number of observations contributing to the model including `sex` could be lower than for the nested model excluding `sex`. In this case, we would have to restrict estimation of the nested model to the ‘estimation sample’ of the full model. If the full model has been estimated first, this can be achieved using `logistic improve therapy if e(sample)`.

Retaining the variable **sex** in the model, the predicted probabilities can be obtained using

```
predict prob
```

and the four different predicted probabilities may be compared with the observed proportions as follows:

```
table sex, contents(mean prob mean improve freq) /*
    */ by(therapy)
```

(see Display 6.5). The agreement is good, so there appears to be no

therapy and sex	mean(prob)	mean(improve)	Freq.
seq			
male	.4332747	.4296875	128
female	.2843846	.3043478	23
alt			
male	.3163268	.32	125
female	.1938763	.173913	23

Display 6.5

strong interaction between sex and type of therapy. (We could test for an interaction between sex and therapy by using **xi: logistic improve i.therapy*i.sex.**) Residuals are not very informative for these data because there are only four different predicted probabilities.

We now fit the proportional odds model using the full ordinal response variable **outc**:

```
ologit outc therapy sex, table
```

The results are shown in Display 6.6. Both **therapy** and **sex** are more significant than before, as might be expected because information was lost in dichotomizing the outcome. The coefficients represent the log odds ratios of being in complete remission versus being at best in partial remission; or equivalently, the log odds ratio of being at least in partial remission rather than having progressive disease or no change. The option **table** has produced the last part of the output to remind us how the proportional odds model is defined. We could calculate the probability that a male (**sex**=1) who is receiving sequential therapy (**therapy**=0) will be in complete remission (**outc**=4) using $1 - \gamma_3$ (see equation (6.4)):

```

Iteration 0:  log likelihood = -399.98398
Iteration 1:  log likelihood = -394.53988
Iteration 2:  log likelihood = -394.52832

Ordered logit estimates                                Number of obs   =      299
                                                        LR chi2(2)      =      10.91
                                                        Prob > chi2     =      0.0043
Log likelihood = -394.52832                            Pseudo R2       =      0.0136

```

outc	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
therapy	-.580685	.2121432	-2.74	0.006	-.9964781	-.164892
sex	-.5413938	.2871764	-1.89	0.059	-1.104249	.0214616
_cut1	-1.859437	.3828641	(Ancillary parameters)			
_cut2	-.2921603	.3672626				
_cut3	.758662	.3741486				

outc	Probability	Observed
1	Pr(xb+u<_cut1)	0.2843
2	Pr(_cut1<xb+u<_cut2)	0.3612
3	Pr(_cut2<xb+u<_cut3)	0.1906
4	Pr(_cut3<xb+u)	0.1639

Display 6.6

```

display 1-1/(1+exp(-0.5413938-0.758662))

.21415563

```

However, a much quicker way of computing the predicted probabilities for all four responses and all combinations of explanatory variables is to use the command **predict**:

```

predict p1 p2 p3 p4

```

and to tabulate the results as follows:

```

table sex, contents(mean p1 mean p2 mean p3 mean p4) /*
*/ by(therapy)

```

giving the table in Display 6.7.

6.3.2 *Diagnosis of heart attacks*

The data in **sck.dat** are read in using

```

infile ck pres abs using sck.dat

```

therapy and sex	mean(p1)	mean(p2)	mean(p3)	mean(p4)
seq				
male	.2111441	.3508438	.2238566	.2141556
female	.3150425	.3729235	.175154	.1368799
alt				
male	.3235821	.3727556	.1713585	.1323038
female	.4511651	.346427	.1209076	.0815003

Display 6.7

Each observation represents all subjects with maximum creatine kinase values in the same range. The total number of subjects is `pres+abs`, calculated using

```
gen tot=pres+abs
```

and the number of subjects with the disease is `pres`. The probability of `pres` ‘successes’ in `tot` trials is binomial with ‘denominator’ `tot` and probability π_i , `Binomial(tot, π_i)`. The programs `logit` and `logistic` are for data where each observation represents a single Bernoulli trial, with binomial ‘denominator’ equal to 1, `Binomial(1, π_i)`. Another command, `blogit`, can be used to analyze the ‘grouped’ data with ‘denominators’ `tot` as considered here:

```
blogit pres tot ck
```

(see Display 6.8). There is a very significant association between CK

Logit estimates	Number of obs	=	360
	LR chi2(1)	=	283.15
	Prob > chi2	=	0.0000
Log likelihood = -93.886407	Pseudo R2	=	0.6013

_outcome	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
ck	.0351044	.0040812	8.60	0.000	.0271053	.0431035
_cons	-2.326272	.2993611	-7.77	0.000	-2.913009	-1.739535

Display 6.8

and the probability of infarct. We now need to investigate whether

it is reasonable to assume that the log odds depends linearly on CK. Therefore, we plot the observed proportions and predicted probabilities as follows:

```
gen prop = pres/tot
predict pred, p
label variable prop "observed"
label variable pred "predicted"
twoway (line pred ck) (scatter prop ck), /*
    */ ytitle("Probability")
```

The `predict` command gives predicted counts by default and therefore the `p` option was used to obtain predicted probabilities instead. In the resulting graph in Figure 6.1, the curve fits the data reasonably well, the largest discrepancy being at CK=280. However, the curve for

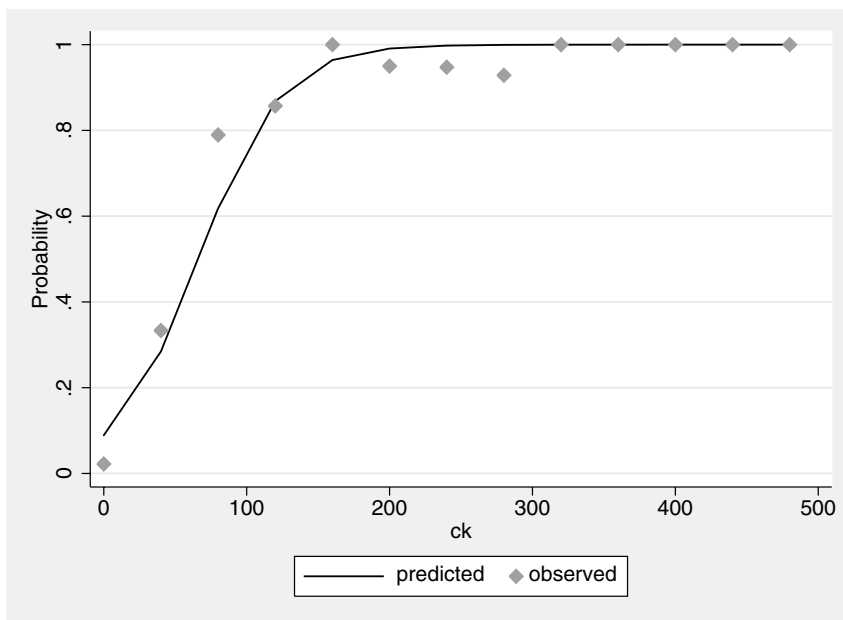


Figure 6.1: Probability of infarct as a function of creatine Kinase levels.

the predicted probabilities is not smooth. Using `mspline` instead of `line` produces a smooth curve, but a more faithful smooth curve can be obtained using the `graph twoway function` command as follows:

```
twoway (function y=1/(1+exp(-_b[_cons]-_b[ck]*x)), /*
    */ range(0 480)) (scatter prop ck), ytitle("Probability")
```

Here we are using the regression coefficients `_b[_cons]` and `_b[ck]` to calculate the predicted probability as a function of some hypothetical variable `x` varying in the range from 0 to 480. This improved graph is shown in Figure 6.2. Note that we could also use

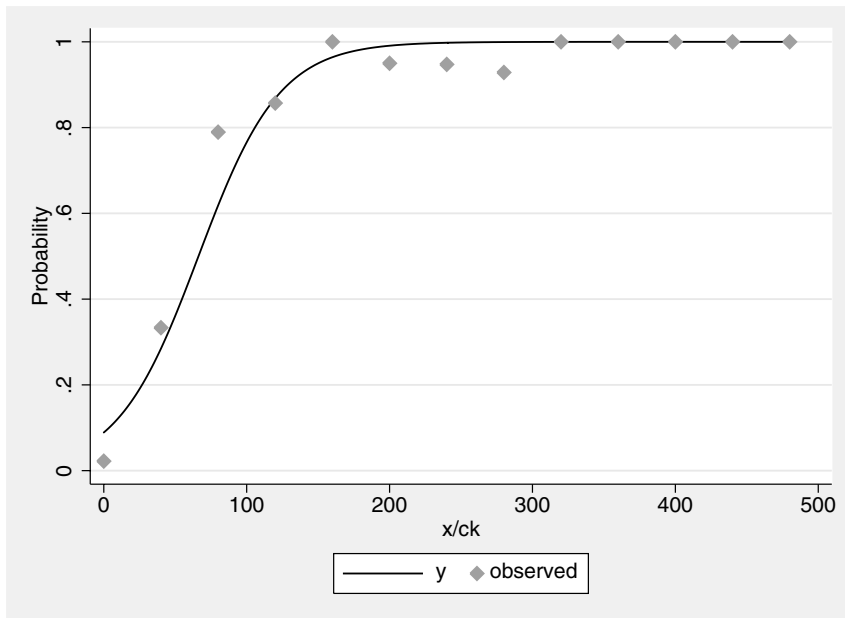


Figure 6.2: Smoother version of [Figure 6.1](#).

the `invlogit()` function to calculate the inverse logit, i.e. `function y = invlogit(_b[_cons]+_b[ck]*x)`.

We will now plot some residuals and then consider the performance of CK as a diagnostic tool, defining the test as positive if CK exceeds a certain threshold. In particular, we will consider the sensitivity (probability of a positive test result if the disease is present) and specificity (probability of a negative test result if the disease is absent) for different thresholds. There are some useful post-estimation commands available for these purposes for use after `logistic` that are not available after `blogit`. We therefore transform the data into the form required for `logistic`, i.e., one observation per Bernoulli trial with outcome `infct` equal to 0 or 1 so that the number of ones per CK level equals `pres`:

```
expand tot
bysort ck: gen infct=_n<=pres
```

We can reproduce the results of `blogit` using `logit`:

```
logit infct ck, nolog
```

(see Display 6.9) where the `nolog` option was used to stop the iteration

Logit estimates	Number of obs	=	360
	LR chi2(1)	=	283.15
	Prob > chi2	=	0.0000
Log likelihood = -93.886407	Pseudo R2	=	0.6013

infct	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
ck	.0351044	.0040812	8.60	0.000	.0271053	.0431035
_cons	-2.326272	.2993611	-7.77	0.000	-2.913009	-1.739535

Display 6.9

history from being given.

One useful type of residual is the standardized Pearson residual for each ‘covariate pattern’, i.e., for each combination of values in the covariates (here for each value of CK). These residuals may be obtained and plotted as follows:

```
predict resi, rstandard
twoway scatter resi ck, mlabel(ck)
```

The graph is shown in [Figure 6.3](#). There are several large outliers. The largest outlier at CK=280 is due to one subject out of 14 not having had an infarct although the predicted probability of an infarct is almost 1.

We now determine the accuracy of the diagnostic test. A *classification table* of the predicted diagnosis (using a cut-off of the predicted probability of 0.5) versus the true diagnosis may be obtained using

```
lstat
```

giving the table shown in Display 6.10. Both the sensitivity and the specificity are relatively high. These characteristics are generally assumed to generalize to other populations whereas the positive and negative predictive values (probabilities of the disease being present/absent if the test is positive/negative) depend on the prevalence (or prior probability) of the condition (see for example Sackett *et al.*, 1991).

The use of other probability cut-offs could be investigated using the option `cutoff(#)` in the above command or using the commands `lroc`

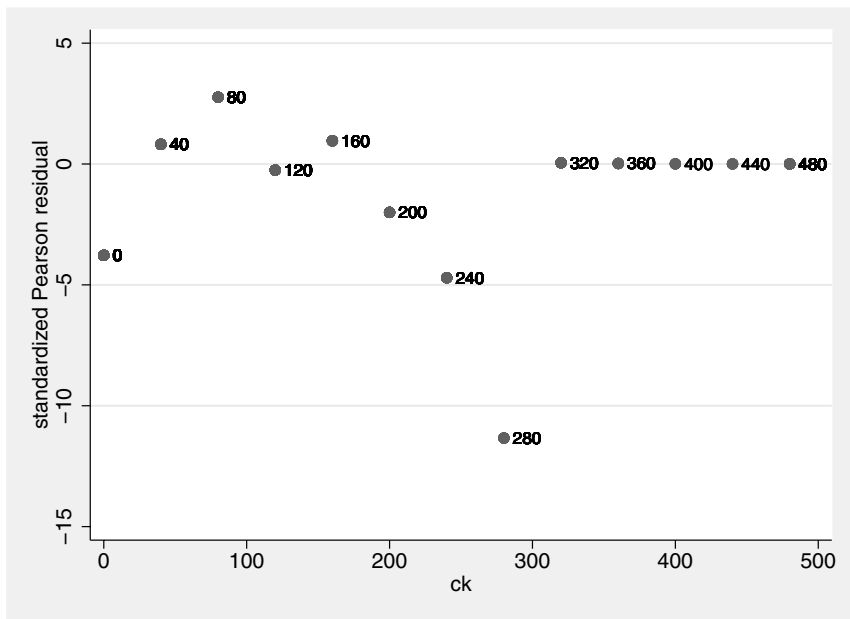


Figure 6.3: Standardized Pearson residuals vs. creatine kinase level.

Logistic model for infct

Classified	True		Total
	D	~D	
+	215	16	231
-	15	114	129
Total	230	130	360

Classified + if predicted $\Pr(D) \geq .5$
True D defined as infct != 0

Sensitivity	$\Pr(+ D)$	93.48%
Specificity	$\Pr(- \sim D)$	87.69%
Positive predictive value	$\Pr(D +)$	93.07%
Negative predictive value	$\Pr(\sim D -)$	88.37%
False + rate for true ~D	$\Pr(+ \sim D)$	12.31%
False - rate for true D	$\Pr(- D)$	6.52%
False + rate for classified +	$\Pr(\sim D +)$	6.93%
False - rate for classified -	$\Pr(D -)$	11.63%
Correctly classified		91.39%

Display 6.10

to plot a ROC-curve (specificity vs. sensitivity for different cut-offs) or `lsens` to plot sensitivity and specificity against cut-off (see exercises).

The above classification table may be misleading because we are testing the model on the same sample that was used to derive it. An alternative approach is to compute predicted probabilities for each observation from a model fitted to the remaining observations. This method, called ‘leave one out’ method or *jackknifing* (see Lachenbruch and Mickey, 1986), can be carried out relatively easily for our data because we only have a small number of covariate and response patterns. Instead of looping through all observations, excluding each observation in the logistic regression command and computing that observation’s predicted probability, we can loop through a subset of observations representing all combinations of covariates and responses found in the data.

First, label each unique covariate pattern consecutively in a variable `num` using `predict` with the `number` option:

```
predict num, number
```

Now generate `first`, equal to one for the first observation in each group of unique covariate and response patterns and zero otherwise:

```
bysort num infct: gen first=_n==1
```

(We could also have used `egen first = tag(num infct)`.) Now define `grp`, equal to the cumulative sum of `first`, obtained using the function `sum()`. This variable numbers the groups of unique covariate and response patterns consecutively:

```
gen grp=sum(first)
```

(An alternative way of generating `grp` without having to first create `first` would be to use the command `egen grp=group(num infct)`.) Now determine the number of unique combinations of CK levels and infarct status:

```
summ grp
```

Variable	Obs	Mean	Std. Dev.	Min	Max
grp	360	8.658333	6.625051	1	20

As there are 20 groups, we need to run `logistic` 20 times (for each value of `grp`), excluding one observation from `grp` to derive the model for predicting the probability for all observations in `grp`.

First generate a variable, `nxt`, that consecutively labels the 20 observations to be excluded in turn:

```
gen nxt = first*grp
```

Now build up a variable **prp** of predicted probabilities as follows:

```
gen prp=0
forvalues n= 1/20 {
    logistic infct ck if nxt~=`n'
    predict p
    replace prp=p if grp==`n'
    drop p
}
```

The purpose of these four commands inside the loop is to

1. derive the model excluding one observation from **grp**,
2. obtain the predicted probabilities **p** (**predict** produces results for the whole sample, not just the estimation sample),
3. set **prp** to the predicted probability for all observations in **grp**, and
4. drop **p** so that it can be defined again in the next iteration.

The classification table for the jackknifed probabilities can be obtained using

```
gen class= prp>=0.5
tab class infct
```

class	infct		Total
	0	1	
0	114	15	129
1	16	215	231
Total	130	230	360

giving the same result as before, although this is not generally the case.

6.4 Exercises

1. Read in the data without using the **expand** command, and reproduce the result of ordinal logistic regressions by using the appropriate weights.
2. Carry out significance tests for an association between **depress** and **life** for the data described in [Chapter 2](#) using
 - a. ordinal logistic regression with **depress** as dependent variable
 - b. logistic regression with **life** as dependent variable.
3. Use **sw** together with **logistic** to find a model for predicting **life** using the data from Chapter 2 with different sets of candidate variables (see [Chapter 3](#)).

4. Produce graphs similar to those in [Figures 6.1](#) and [6.2](#) using a probit model instead of a logit (see help for `bprobit`).
5. Explore the use of `lstat`, `cutoff(#)`, `lroc`, and `lsens` for the diagnosis data.

Chapter 7

Generalized Linear Models: Australian School Children

7.1 Description of data

This chapter reanalyzes a number of datasets discussed in previous chapters and in addition, describes the analysis of a new dataset given in Aitkin (1978). These data come from a sociological study of Australian aboriginal and white children. The sample included children from four age groups (final year in primary school and first three years in secondary school) who were classified as slow or average learners. The number of days absent from school during the school year was recorded for each child. The data are given in [Table 7.1](#). The variables are as follows:

- **eth**: ethnic group (A=aboriginal, N=white)
- **sex**: sex (M=male, F=female)
- **age**: class in school (F0, F1, F2, F3)
- **lrn**: average or slow learner (SL=slow learner, AL=average learner)
- **days**: number of days absent from school in one year

One aim of the analysis is to investigate ethnic differences in the mean number of days absent from school while controlling for the other potential predictors **sex**, **age**, and **lrn**.

7.2 Generalized linear models

Previous chapters have described linear (Chapter 3) and logistic regression (Chapter 6). In this chapter, we will describe a more general class of models, called *generalized linear models*, of which linear regression and logistic regression are special cases.

Both linear and logistic regression involve a linear combination of the explanatory variables, called the linear predictor, of the form

$$\begin{aligned}\eta_i &= \beta_0 + \beta x_{1i} + \beta x_{2i} + \cdots + \beta x_{pi} \\ &= \boldsymbol{\beta}^T \mathbf{x}_i.\end{aligned}\tag{7.1}$$

In both types of regression, the linear predictor determines the expectation μ_i of the response variable. In linear regression, where the response is continuous, μ_i is directly equated with the linear predictor. This is not advisable when the response is dichotomous because in this case the expectation is a probability which must satisfy $0 \leq \mu_i \leq 1$. In logistic regression, the linear predictor is therefore equated with a function of μ_i , the logit, $\eta_i = \log(\mu_i/(1 - \mu_i))$. In generalized linear models, the linear predictor may be equated with any of a number of different functions $g(\mu_i)$ of μ_i , called *link functions*; that is,

$$\eta_i = g(\mu_i).\tag{7.2}$$

In linear regression, the probability distribution of the response variable is assumed to be normal with mean μ_i . In logistic regression a binomial distribution is assumed with probability parameter μ_i . Both the normal and binomial distributions come from the same family of distributions, called the exponential family,

$$f(y_i; \theta_i, \phi) = \exp\{(y_i \theta_i - b(\theta_i))/a(\phi) + c(y_i, \phi)\}.\tag{7.3}$$

For example, for the normal distribution,

$$\begin{aligned}f(y_i; \theta_i, \phi) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-(y_i - \mu_i)^2/2\sigma^2\} \\ &= \exp\{(y_i \mu_i - \mu_i^2/2)/\sigma^2 - \frac{1}{2}(y_i^2/\sigma^2 + \log(2\pi\sigma^2))\}\end{aligned}\tag{7.4}$$

so that $\theta_i = \mu_i$, $b(\theta_i) = \theta_i^2/2$, $\phi = \sigma^2$, and $a(\phi) = \phi$.

The parameter θ_i , a function of μ_i , is called the *canonical link*. The canonical link is frequently chosen as the link function (and is the

Table 7.1 Data in quine.dta presented in four columns to save space. (Taken from Aitkin (1978) with permission of the Royal Statistical Society.)

eth	sex	age	lrn	days	eth	sex	age	lrn	days	eth	sex	age	lrn	days	eth	sex	age	lrn	days
A	M	F0	SL	2	A	M	F0	SL	11	A	M	F0	SL	14	A	M	F0	AL	5
A	M	F0	AL	5	A	M	F0	AL	13	A	M	F0	AL	20	A	M	F0	AL	22
A	M	F1	SL	6	A	M	F1	SL	6	A	M	F1	SL	15	A	M	F1	AL	7
A	M	F1	AL	14	A	M	F2	SL	6	A	M	F2	SL	32	A	M	F2	SL	53
A	M	F2	SL	57	A	M	F2	AL	14	A	M	F2	AL	16	A	M	F2	AL	16
A	M	F2	AL	17	A	M	F2	AL	40	A	M	F2	AL	43	A	M	F2	AL	46
A	M	F3	AL	8	A	M	F3	AL	23	A	M	F3	AL	23	A	M	F3	AL	28
A	M	F3	AL	34	A	M	F3	AL	36	A	M	F3	AL	38	A	F	F0	SL	3
A	F	F0	AL	5	A	F	F0	AL	11	A	F	F0	AL	24	A	F	F0	AL	45
A	F	F1	SL	5	A	F	F1	SL	6	A	F	F1	SL	6	A	F	F1	SL	9
A	F	F1	SL	13	A	F	F1	SL	23	A	F	F1	SL	25	A	F	F1	SL	32
A	F	F1	SL	53	A	F	F1	SL	54	A	F	F1	AL	5	A	F	F1	AL	5
A	F	F1	AL	11	A	F	F1	AL	17	A	F	F1	AL	19	A	F	F2	SL	8
A	F	F2	SL	13	A	F	F2	SL	14	A	F	F2	SL	20	A	F	F2	SL	47
A	F	F2	SL	48	A	F	F2	SL	60	A	F	F2	SL	81	A	F	F2	AL	2
A	F	F3	AL	0	A	F	F3	AL	2	A	F	F3	AL	3	A	F	F3	AL	5
A	F	F3	AL	10	A	F	F3	AL	14	A	F	F3	AL	21	A	F	F3	AL	36
A	F	F3	AL	40	N	M	F0	SL	6	N	M	F0	SL	17	N	M	F0	SL	67
N	M	F0	AL	0	N	M	F0	AL	0	N	M	F0	AL	2	N	M	F0	AL	7
N	M	F0	AL	11	N	M	F0	AL	12	N	M	F1	SL	0	N	M	F1	SL	0
N	M	F1	SL	5	N	M	F1	SL	5	N	M	F1	SL	5	N	M	F1	SL	11
N	M	F1	SL	17	N	M	F1	AL	3	N	M	F1	AL	4	N	M	F2	SL	22
N	M	F2	SL	30	N	M	F2	SL	36	N	M	F2	AL	8	N	M	F2	AL	0
N	M	F2	AL	1	N	M	F2	AL	5	N	M	F2	AL	7	N	M	F2	AL	16
N	M	F2	AL	27	N	M	F3	AL	0	N	M	F3	AL	30	N	M	F3	AL	10
N	M	F3	AL	14	N	M	F3	AL	27	N	M	F3	AL	41	N	M	F3	AL	69
N	F	F0	SL	25	N	F	F0	AL	10	N	F	F0	AL	11	N	F	F0	AL	20
N	F	F0	AL	33	N	F	F1	SL	5	N	F	F1	SL	7	N	F	F1	SL	0
N	F	F1	SL	1	N	F	F1	SL	5	N	F	F1	SL	5	N	F	F1	SL	5
N	F	F1	SL	5	N	F	F1	SL	7	N	F	F1	SL	11	N	F	F1	SL	15
N	F	F1	AL	5	N	F	F1	AL	14	N	F	F1	AL	6	N	F	F1	AL	6
N	F	F1	AL	7	N	F	F1	AL	28	N	F	F2	SL	0	N	F	F2	SL	5
N	F	F2	SL	14	N	F	F2	SL	2	N	F	F2	SL	2	N	F	F2	SL	3
N	F	F2	SL	8	N	F	F2	SL	10	N	F	F2	SL	12	N	F	F2	AL	1
N	F	F3	AL	1	N	F	F3	AL	9	N	F	F3	AL	22	N	F	F3	AL	3
N	F	F3	AL	3	N	F	F3	AL	5	N	F	F3	AL	15	N	F	F3	AL	18
N	F	F3	AL	22	N	F	F3	AL	37										

default link in the Stata command for fitting generalized linear models, `glm`), although the canonical link is not necessarily more appropriate than any other link. Table 7.2 lists some of the most common distributions and their canonical link functions used in generalized linear models.

Table 7.2 Probability distributions and their canonical link functions

Distribution	Variance function	Dispersion parameter	Link function	$g(\mu) = \theta(\mu)$
Normal	1	σ^2	identity	μ
Binomial	$\mu(1 - \mu)$	1	logit	$\log(\mu/(1 - \mu))$
Poisson	μ	1	log	$\ln(\mu)$
Gamma	μ^2	ν^{-1}	reciprocal	$1/\mu$

The conditional mean and variance of Y_i are given by

$$E(Y_i|\mathbf{x}_i) = b'(\theta_i) = \mu_i \quad (7.5)$$

and

$$\text{var}(Y_i|\mathbf{x}_i) = b''(\theta_i)a(\phi) = V(\mu_i)a(\phi) \quad (7.6)$$

where $b'(\theta_i)$ and $b''(\theta_i)$ denote the first and second derivatives of $b(\cdot)$ evaluated at θ_i , and the variance function $V(\mu_i)$ is obtained by expressing $b''(\theta_i)$ as a function of μ_i . It can be seen from (7.4) that the variance for the normal distribution is simply σ^2 regardless of the value of the mean μ_i , i.e., the variance function is 1.

The data on Australian school children will be analyzed by assuming a Poisson distribution for the number of days absent from school. The Poisson distribution is the appropriate distribution of the number of events observed over a period of time, if these events occur independently in continuous time at a constant instantaneous probability rate (or incidence rate); see for example Clayton and Hills (1993). The Poisson distribution is given by

$$f(y_i; \mu_i) = \mu_i^{y_i} e^{-\mu_i} / y_i!, \quad y_i = 0, 1, 2, \dots \quad (7.7)$$

Taking the logarithm and summing over observations, the log likelihood is

$$l(\boldsymbol{\mu}; \mathbf{y}) = \sum_i \{(y_i \ln \mu_i - \mu_i) - \ln(y_i!)\} \quad (7.8)$$

so that $\theta_i = \ln \mu_i$, $b(\theta_i) = \exp(\theta_i)$, $\phi = 1$, $a(\phi) = 1$, and $\text{var}(Y_i|\mathbf{x}_i) = \exp(\theta_i) = \mu_i$. Therefore, the variance of the Poisson distribution is not constant, but equal to the mean. Unlike the normal distribution, the Poisson distribution has no separate parameter for the variance and the same is true of the binomial distribution. Table 7.2 shows the variance functions and dispersion parameters for some commonly used probability distributions.

7.2.1 Model selection and measure of fit

Lack of fit may be expressed by the deviance, which is minus twice the difference between the maximized log likelihood of the model and the maximum likelihood achievable, i.e., the maximized likelihood of the *full* or saturated model. For the normal distribution, the deviance is simply the residual sum of squares. Another measure of lack of fit is the generalized Pearson X^2 ,

$$X^2 = \sum_i (y_i - \hat{\mu}_i)^2 / V(\hat{\mu}_i), \quad (7.9)$$

which, for the Poisson distribution, is just the familiar Pearson chi-squared statistic for two-way cross-tabulations (since $V(\hat{\mu}_i) = \hat{\mu}_i$). Both the deviance and Pearson X^2 have χ^2 distributions when the sample size tends to infinity. When the dispersion parameter ϕ is fixed (not estimated), an analysis of deviance may be used for comparing nested models. To test the null hypothesis that the restrictions leading to the nested model are true, the difference in deviance between two models is compared with the χ^2 distribution with degrees of freedom equal to the difference in model degrees of freedom.

The Pearson and deviance residuals are defined as the (signed) square roots of the contributions of the individual observations to the Pearson X^2 and deviance respectively. These residuals may be used to assess the appropriateness of the link and variance functions.

A relatively common phenomenon with count data is *overdispersion*, i.e., the variance is greater than that of the assumed distribution (binomial with denominator greater than 1 or Poisson). This overdispersion may be due to extra variability in the parameter μ_i which has not been completely explained by the covariates. One way of addressing the problem is to allow μ_i to vary randomly according to some (prior) distribution and to assume that conditional on μ_i , the response variable follows the binomial (or Poisson) distribution. Such models are called *random effects models*; see also Chapter 9.

A more pragmatic way of accommodating overdispersion in the model is to assume that the variance is proportional to the variance

function, but to estimate the dispersion or scale parameter ϕ rather than assuming the value 1 appropriate for the distributions. For the Poisson distribution, the variance is modeled as

$$\text{var}(Y|\mathbf{x}_i) = \phi\mu_i \quad (7.10)$$

where ϕ is estimated from the Deviance or Pearson X^2 . (This is analogous to the estimation of the residual variance in linear regression models from the residual sums of squares.) This parameter is then used to scale the estimated standard errors of the regression coefficients. This approach of assuming a variance function that does not correspond to any probability distribution is an example of the *quasi-likelihood* approach. If the variance is not proportional to the variance function, robust standard errors can be used as described in the next section. See McCullagh and Nelder (1989) and Hardin and Hilbe (2001) for more details on generalized linear models.

7.2.2 Robust standard errors of parameter estimates

A very useful feature of Stata is that robust standard errors of estimated parameters can be obtained for most estimation commands. In maximum likelihood estimation, the standard errors of the estimated parameters are derived from the Hessian (matrix of second derivatives with respect to the parameters) of the log likelihood. However, these standard errors are correct only if the likelihood is the true likelihood of the data. If this assumption is not correct, for instance due to omission of covariates, misspecification of the link function or probability distribution function, we can still use robust estimates of the standard errors known as the Huber, White, or sandwich variance estimates (for details, see Binder, 1983).

In the description of the robust variance estimator in the *Stata User's Guide* (Section 23.14), it is pointed out that the use of robust standard errors implies a less ambitious interpretation of the parameter estimates and their standard errors than a model-based approach. Instead of assuming that the model is 'true' and attempting to estimate 'true' parameters, we just consider the properties of the estimator (whatever it may mean) under repeated sampling and define the standard error as its sampling standard deviation (see also the FAQ by Sribney, 1998).

Another approach to estimating the standard errors without making any distributional assumptions is *bootstrapping* (Efron and Tibshirani, 1993). If we could obtain repeated samples from the population (from which our data were sampled), we could obtain an empirical sampling distribution of the parameter estimates. In Monte Carlo sim-

ulation, the required samples are drawn from the assumed distribution. In bootstrapping, the sample is resampled ‘to approximate what would happen if the population were sampled’ (Manly, 1997). Bootstrapping works as follows. Take a random sample of n observations (n is the sample size), with replacement, and estimate the regression coefficients. Repeat this a number of times to obtain a sample of estimates. From the resulting sample of parameter estimates, obtain the empirical variance-covariance matrix of the parameter estimates. Confidence intervals may be constructed using the estimated variance or directly from the appropriate centiles of the empirical distribution of parameter estimates. See Manly (1997) and Efron and Tibshirani (1993) for more information on the bootstrap.

7.3 Analysis using Stata

The `glm` command can be used to fit generalized linear models. The syntax is analogous to `logistic` and `regress` except that the options `family()` and `link()` are used to specify the probability distribution of the response and the link function, respectively. We first analyze data from the previous chapter to show how linear regression, ANOVA and logistic regression are performed using `glm` and then move on to the data on Australian school children.

7.3.1 *Linear regression*

First, we show how linear regression can be carried out using `glm`. In [Chapter 3](#), the U.S. air-pollution data were read in using the instructions

```
infile str10 town so2 temp manuf pop wind precip days /*
      */ using usair.dat
drop if town=="Chicago"
```

and now we regress `so2` on a number of variables using

```
glm so2 temp pop wind precip, fam(gauss) link(id)
```

(see Display 7.1). The results are identical to those of the regression analysis. The scale parameter given on the right-hand side above the regression table represents the residual variance given under **Residual MS** in the analysis of variance table of the regression analysis in [Chapter 3](#). We can estimate robust standard errors using the `robust` option

```
glm so2 temp pop wind precip, fam(gauss) link(id) robust
```

(see Display 7.2) giving slightly different standard errors, suggesting

Iteration 0: log likelihood = -167.48483

Generalized linear models	No. of obs	=	40
Optimization : ML: Newton-Raphson	Residual df	=	35
	Scale parameter	=	290.0043
Deviance = 10150.15199	(1/df) Deviance	=	290.0043
Pearson = 10150.15199	(1/df) Pearson	=	290.0043
Variance function: V(u) = 1	[Gaussian]		
Link function : g(u) = u	[Identity]		
Standard errors : OIM			
Log likelihood = -167.4848314	AIC	=	8.624242
BIC = 10021.04121			

so2	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
temp	-1.810123	.4404001	-4.11	0.000	-2.673292	-.9469549
pop	.0113089	.0074091	1.53	0.127	-.0032126	.0258304
wind	-3.085284	2.096471	-1.47	0.141	-7.194292	1.023723
precip	.5660172	.2508601	2.26	0.024	.0743404	1.057694
_cons	131.3386	34.32034	3.83	0.000	64.07195	198.6052

Display 7.1

Iteration 0: log pseudo-likelihood = -167.48483

Generalized linear models	No. of obs	=	40
Optimization : ML: Newton-Raphson	Residual df	=	35
	Scale parameter	=	290.0043
Deviance = 10150.15199	(1/df) Deviance	=	290.0043
Pearson = 10150.15199	(1/df) Pearson	=	290.0043
Variance function: V(u) = 1	[Gaussian]		
Link function : g(u) = u	[Identity]		
Standard errors : Sandwich			
Log pseudo-likelihood = -167.4848314	AIC	=	8.624242
BIC = 10021.04121			

so2	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
temp	-1.810123	.3280436	-5.52	0.000	-2.453077	-1.16717
pop	.0113089	.0079634	1.42	0.156	-.0042992	.0269169
wind	-3.085284	1.698542	-1.82	0.069	-6.414366	.2437966
precip	.5660172	.1818484	3.11	0.002	.2096009	.9224335
_cons	131.3386	18.21993	7.21	0.000	95.62816	167.049

Display 7.2

that some assumptions may not be entirely satisfied.

7.3.2 ANOVA

We now show how an analysis of variance model can be fitted using `glm`, using the slimming clinic example of [Chapter 5](#). The data are read using

```
infile cond status resp using slim.dat
```

and the full, saturated model may be obtained using

```
xi: glm resp i.cond*i.status, fam(gauss) link(id)
```

(see Display 7.3). This result is identical to that obtained using the

```

i.cond          _Icond_1-2          (naturally coded; _Icond_1 omitted)
i.status         _Istatus_1-2        (naturally coded; _Istatus_1 omitted)
i.cond*i.status  _IconXsta_#-#       (coded as above)
Iteration 0:    log likelihood = -107.01782
Generalized linear models                               No. of obs      =       34
Optimization    : ML: Newton-Raphson                   Residual df    =       30
                                                        Scale parameter =   35.9616
Deviance        = 1078.848121                          (1/df) Deviance =   35.9616
Pearson         = 1078.848121                          (1/df) Pearson  =   35.9616
Variance function: V(u) = 1                            [Gaussian]
Link function   : g(u) = u                             [Identity]
Standard errors : OIM
Log likelihood  = -107.0178176                          AIC              =   6.53046
BIC            =  973.0573056

```

resp	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_Icond_2	.6780002	3.234433	0.21	0.834	-5.661372	7.017373
_Istatus_2	6.128834	3.19204	1.92	0.055	-.1274504	12.38512
IconXsta~2	-.2655002	4.410437	-0.06	0.952	-8.909799	8.378798
_cons	-7.298	2.68185	-2.72	0.007	-12.55433	-2.04167

Display 7.3

command

```
xi: regress resp i.cond*i.status
```

(see exercises in Chapter 5).

We can obtain the F -statistics for the interaction term by saving the deviance of the above model (residual sum of squares) in a local macro and refitting the model with the interaction removed:

```
local dev0=e(deviance)
```

```
xi: glm resp i.cond i.status, fam(gauss) link(id)
```

(see Display 7.4). The increase in deviance caused by the removal of

```
i.cond      _Icond_1-2      (naturally coded; _Icond_1 omitted)
i.status     _Istatus_1-2    (naturally coded; _Istatus_1 omitted)
Iteration 0:  log likelihood = -107.01987
```

Generalized linear models		No. of obs	=	34
Optimization	: ML: Newton-Raphson	Residual df	=	31
Deviance	= 1078.97844	Scale parameter	=	34.80576
Pearson	= 1078.97844	(1/df) Deviance	=	34.80576
		(1/df) Pearson	=	34.80576
Variance function:	V(u) = 1	[Gaussian]		
Link function	: g(u) = u	[Identity]		
Standard errors	: OIM			
Log likelihood	= -107.0198709	AIC	=	6.471757
BIC	= 969.6612634			

resp	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_Icond_2	.5352102	2.163277	0.25	0.805	-3.704734	4.775154
_Istatus_2	5.989762	2.167029	2.76	0.006	1.742463	10.23706
_cons	-7.199832	2.094584	-3.44	0.001	-11.30514	-3.094524

Display 7.4

the interaction term represents the sum of squares of the interaction term after eliminating the main effects:

```
local dev1=e(deviance)
local ddev=`dev1'-`dev0'
display `ddev'
```

```
.13031826
```

and the F -statistic is simply the mean sum of squares of the interaction term after eliminating the main effects divided by the residual mean square of the full model. The numerator and denominator degrees of freedom are 1 and 30 respectively, so that F and the associated p -value may be obtained as follows:

```
local f=(`ddev'/1)/(`dev0'/30)
display `f'
```

.00362382

```
display fprob(1,30,`f`)
```

.95239706

The general method for testing the difference in fit of two nested generalized linear models, using the difference in deviance, is not appropriate here because the scale parameter $\phi = \sigma^2$ was estimated.

7.3.3 Logistic regression

The logistic regression analysis of [Chapter 6](#) may also be repeated using `glm`. We first read the data as before, without replicating records.

```
infile fr1 fr2 fr3 fr4 using tumor.dat, clear
gen therapy=int((_n-1)/2)
sort therapy
by therapy: gen sex=_n
reshape long fr, i(therapy sex) j(outc)
gen improve=outc
recode improve 1/2=0 3/4=1
list
```

	therapy	sex	outc	fr	improve
1.	0	1	1	28	0
2.	0	1	2	45	0
3.	0	1	3	29	1
4.	0	1	4	26	1
5.	0	2	1	4	0
6.	0	2	2	12	0
7.	0	2	3	5	1
8.	0	2	4	2	1
9.	1	1	1	41	0
10.	1	1	2	44	0
11.	1	1	3	20	1
12.	1	1	4	20	1
13.	1	2	1	12	0
14.	1	2	2	7	0
15.	1	2	3	3	1
16.	1	2	4	1	1

The `glm` command may be used with weights just like other estimation commands, so that we can analyze the data using `fweight`.

```
glm improve therapy sex [fweight=fr], fam(binomial) /*
  */ link(logit)
```

(see Display 7.5).

```
Iteration 0:  log likelihood = -190.9051
Iteration 1:  log likelihood = -190.63185
Iteration 2:  log likelihood = -190.63171
Iteration 3:  log likelihood = -190.63171

Generalized linear models                No. of obs      =       299
Optimization      : ML: Newton-Raphson  Residual df     =       296
                                                Scale parameter =         1
Deviance          = 381.2634298          (1/df) Deviance = 1.288052
Pearson           = 298.7046083          (1/df) Pearson  = 1.009137
Variance function: V(u) = u*(1-u)        [Bernoulli]
Link function     : g(u) = ln(u/(1-u))   [Logit]
Standard errors   : OIM

Log likelihood    = -190.6317149          AIC              = 1.295195
BIC              = -1306.067868
```

improve	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
therapy	-.5022014	.2456898	-2.04	0.041	-.9837445	-.0206582
sex	-.6543125	.3714739	-1.76	0.078	-1.382388	.0737629
_cons	.3858095	.4514172	0.85	0.393	-.4989519	1.270571

Display 7.5

The likelihood ratio test for `sex` can be obtained as follows:

```
local dev0=e(deviance)
quietly glm improve therapy [fweight=fr], fam(binomial) /*
  */ link(logit)
local dev1=e(deviance)
dis `dev1'-'dev0'

3.3459816

dis chiprob(1,`dev1'-'dev0')

.0673693
```

which gives the same result as in [Chapter 6](#). (We could also have used `estimates store` and `lrtest` as in [Section 6.3.1](#).)

7.3.4 Australian school children

We now move on to analyze the data in [Table 7.1](#) to investigate differences between aboriginal and white children in the mean number of days absent from school after controlling for other covariates. The data are available as a Stata file `quine.dta` and may therefore be read simply by using the command

```
use quine, clear
```

The variables are of type string and can be converted to numeric using the `encode` command as follows:

```
encode eth, gen(ethnic)
drop eth
encode sex, gen(gender)
drop sex
encode age, gen(class)
drop age
encode lrn, gen(slow)
drop lrn
```

The number of children in each of the combinations of categories of `gender`, `class`, and `slow` can be found using

```
table slow class ethnic, contents(freq) by(gender)
```

(see Display 7.6). This reveals that there were no ‘slow learners’ in

gender and slow		ethnic and class							
		A				N			
		F0	F1	F2	F3	F0	F1	F2	F3
F									
	AL	4	5	1	9	4	6	1	10
	SL	1	10	8		1	11	9	
M									
	AL	5	2	7	7	6	2	7	7
	SL	3	3	4		3	7	3	

Display 7.6

class F3. A table of the means and standard deviations is obtained using

```
table slow class ethnic, contents(mean days sd days) /*
```



```
*/ by(gender) format(%4.1f)
```

(see Display 7.7), where the `format()` option causes only a single deci-

gender and slow		ethnic and class							
		A				N			
		F0	F1	F2	F3	F0	F1	F2	F3
F	AL	21.3	11.4	2.0	14.6	18.5	11.0	1.0	13.5
		17.7	6.5		14.9	10.7	8.9		11.5
	SL	3.0	22.6	36.4		25.0	6.0	6.2	
			18.7	26.5			4.2	5.0	
M	AL	13.0	10.5	27.4	27.1	5.3	3.5	9.1	27.3
		8.0	4.9	14.7	10.4	5.4	0.7	9.5	22.9
	SL	9.0	9.0	37.0		30.0	6.1	29.3	
		6.2	5.2	23.4		32.5	6.1	7.0	

Display 7.7

mal place to be given. This table suggests that the variance associated with the Poisson distribution is not appropriate here as squaring the standard deviations (to get the variances) results in values that are greater than the means, i.e., there is overdispersion. In this case, the overdispersion probably arises from substantial variability in children's underlying tendency to miss days of school that cannot be fully explained by the variables we have included in the model.

Ignoring the problem of overdispersion for the moment, a generalized linear model with a Poisson family and log link can be fitted using

```
glm days slow class ethnic gender, fam(pois) link(log)
```

(see Display 7.8). The algorithm takes four iterations to converge to the maximum likelihood (or minimum deviance) solution. In the absence of overdispersion, the scale parameters based on the Pearson X^2 or the deviance should be close to 1. The values of 14.1 and 12.5, respectively, therefore indicate that there is overdispersion. The confidence intervals are therefore likely to be too narrow. McCullagh and Nelder (1989) use the Pearson X^2 divided by the degrees of freedom to estimate the scale parameter for the quasi-likelihood method for Poisson models. This

```

Iteration 0:  log likelihood = -1192.3347
Iteration 1:  log likelihood = -1178.6003
Iteration 2:  log likelihood = -1178.5612
Iteration 3:  log likelihood = -1178.5612

Generalized linear models
Optimization   : ML: Newton-Raphson
No. of obs     =      146
Residual df    =      141
Scale parameter =       1
(1/df) Deviance = 12.54358
(1/df) Pearson  = 14.11449
Variance function: V(u) = u
Link function   : g(u) = ln(u)
Standard errors : OIM
Log likelihood  = -1178.561184
BIC             = 1065.956757
AIC             = 16.21317

```

days	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
slow	.2661578	.0445715	5.97	0.000	.1787992	.3535164
class	.2094662	.0218245	9.60	0.000	.166691	.2522414
ethnic	-.5511688	.0418391	-13.17	0.000	-.633172	-.4691656
gender	.2256243	.0415927	5.42	0.000	.1441041	.3071445
_cons	2.336676	.1427925	16.36	0.000	2.056808	2.616545

Display 7.8

may be achieved using the option `scale(x2)`:

```

glm days slow class ethnic gender, fam(pois) link(log) /*
    */ scale(x2) nolog

```

(see Display 7.9). Here, the option `nolog` was used to stop the iteration log from being printed. Allowing for overdispersion has no effect on the regression coefficients, but a large effect on the p -values and confidence intervals so that `gender` and `slow` are now no longer significant at the 5% level. These terms will be removed from the model. The coefficients can be interpreted as the difference in the logs of the predicted mean counts between groups. For example, the log of the predicted mean number of days absent from school for white children is -0.55 lower than that for aboriginals.

$$\ln(\hat{\mu}_2) = \ln(\hat{\mu}_1) - 0.55 \quad (7.11)$$

Exponentiating the coefficients yields count ratios (or rate ratios). Stata exponentiates all coefficients and confidence intervals when the option `eform` is used:

```

glm days class ethnic, fam(pois) link(log) scale(x2) /*
    */ eform nolog

```

Generalized linear models	No. of obs	=	146
Optimization : ML: Newton-Raphson	Residual df	=	141
	Scale parameter	=	1
Deviance	(1/df) Deviance	=	12.54358
Pearson	(1/df) Pearson	=	14.11449
Variance function: V(u) = u	[Poisson]		
Link function : g(u) = ln(u)	[Log]		
Standard errors : OIM			
Log likelihood	AIC	=	16.21317
BIC			

days	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
slow	.2661578	.1674519	1.59	0.112	-.0620419	.5943575
class	.2094662	.0819929	2.55	0.011	.0487631	.3701693
ethnic	-.5511688	.1571865	-3.51	0.000	-.8592486	-.243089
gender	.2256243	.1562606	1.44	0.149	-.0806409	.5318896
_cons	2.336676	.5364608	4.36	0.000	1.285233	3.38812

(Standard errors scaled using square root of Pearson X2-based dispersion)

Display 7.9

(see Display 7.10). Therefore, white children are absent from school about 58% as often as aboriginal children (95% confidence interval from 42% to 79%) after controlling for `class`. We have treated `class` as a continuous covariate. This implies that the rate ratio for two categories is a constant multiple of the difference in scores assigned to these categories; for example the rate ratio comparing classes F1 and F0 is the same as that comparing F2 and F1. To see whether this appears to be appropriate, we can form the square of `class` and include this in the model:

```
gen class2=class^2
glm days class class2 ethnic, fam(pois) link(log) /*
    */ scale(x2) eform nolog
```

(see Display 7.11). This term is not significant so we can return to the simpler model. (Note that the interaction between `class` and `ethnic` is also not significant, see exercises.)

We now look at the residuals for this model. The post-estimation command `predict` that was used for `regress` and `logistic` can be used here as well. To obtain standardized Pearson residuals, use the `pearson` option with `predict` and divide the residuals by the square root of the estimated dispersion parameter stored in `e(dispersp_ps)`:

```
qui glm days class ethnic, fam(pois) link(log) scale(x2)
```

Generalized linear models	No. of obs	=	146
Optimization : ML: Newton-Raphson	Residual df	=	143
	Scale parameter	=	1
Deviance = 1823.481292	(1/df) Deviance	=	12.75162
Pearson = 2091.29704	(1/df) Pearson	=	14.62445
Variance function: V(u) = u	[Poisson]		
Link function : g(u) = ln(u)	[Log]		
Standard errors : OIM			
Log likelihood = -1205.979185	AIC	=	16.56136
BIC = 1110.825545			

days	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
class	1.177895	.0895256	2.15	0.031	1.014872	1.367105
ethnic	.5782531	.0924981	-3.42	0.001	.4226284	.7911836

(Standard errors scaled using square root of Pearson X2-based dispersion)

Display 7.10

Generalized linear models	No. of obs	=	146
Optimization : ML: Newton-Raphson	Residual df	=	142
	Scale parameter	=	1
Deviance = 1822.560172	(1/df) Deviance	=	12.83493
Pearson = 2081.259434	(1/df) Pearson	=	14.65676
Variance function: V(u) = u	[Poisson]		
Link function : g(u) = ln(u)	[Log]		
Standard errors : OIM			
Log likelihood = -1205.518625	AIC	=	16.56875
BIC = 1114.888032			

days	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
class	1.059399	.4543011	0.13	0.893	.4571295	2.455158
class2	1.020512	.0825501	0.25	0.802	.8708906	1.195839
ethnic	.5784944	.092643	-3.42	0.001	.4226525	.7917989

(Standard errors scaled using square root of Pearson X2-based dispersion)

Display 7.11

```
predict resp, pearson
gen stres=resp/sqrt(e(dispersp_ps))
```

The residuals are plotted against the linear predictor using

```
predict xb, xb
twoway scatter stres xb, ytitle("Standardized Residuals")
```

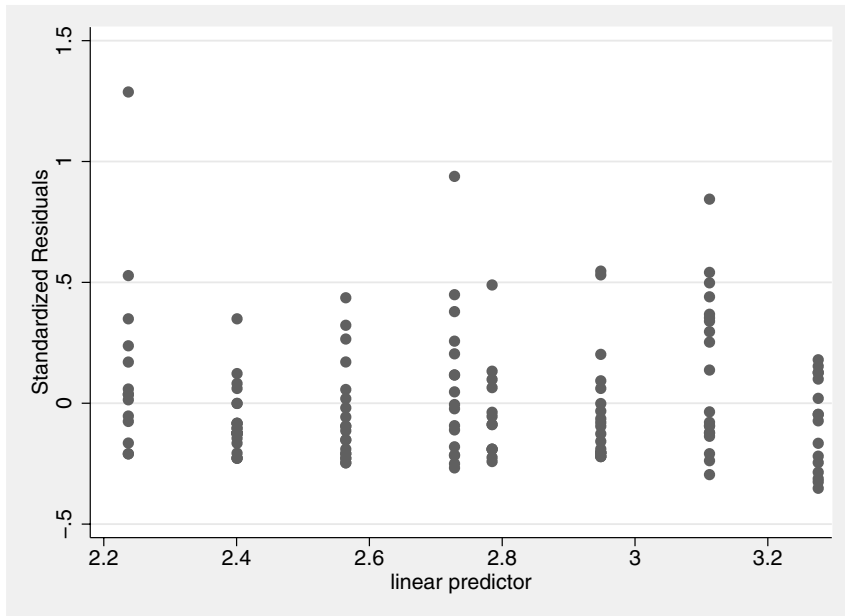


Figure 7.1: Standardized residuals against linear predictor.

with the result shown in Figure 7.1.

There is one large outlier. In order to find out which observation this is, we list a number of variables for cases with large standardized Pearson residuals:

```
predict mu, mu
list stres days mu ethnic class if stres>2|stres<=-2
```

(see Display 7.12). Case 72, a white primary school child, has a very large residual.

We now also check the assumptions of the model by estimating robust standard errors.

```
glm days class ethnic, fam(pois) link(log) robust nolog
```

(see Display 7.13) giving almost exactly the same p -values as the quasi-

	stres	days	mu	ethnic	class
45.	2.028696	53	19.07713	A	F1
46.	2.0885	54	19.07713	A	F1
58.	2.067949	60	22.47085	A	F2
59.	3.225102	81	22.47085	A	F2
72.	4.919289	67	9.365361	N	F0
104.	3.585005	69	15.30538	N	F3
109.	2.017287	33	9.365361	N	F0

Display 7.12

Generalized linear models	No. of obs	=	146
Optimization : ML: Newton-Raphson	Residual df	=	143
	Scale parameter	=	1
Deviance = 1823.481292	(1/df) Deviance	=	12.75162
Pearson = 2091.29704	(1/df) Pearson	=	14.62445
Variance function: V(u) = u	[Poisson]		
Link function : g(u) = ln(u)	[Log]		
Standard errors : Sandwich			
Log pseudo-likelihood = -1205.979185	AIC	=	16.56136
BIC = 1110.825545			

days	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
class	.1637288	.0766153	2.14	0.033	.0135655	.313892
ethnic	-.5477436	.1585381	-3.45	0.001	-.8584725	-.2370147
_cons	3.168776	.3065466	10.34	0.000	2.567956	3.769597

Display 7.13

likelihood solution,

`glm days class ethnic, fam(pois) link(log) scale(x2) nolog`
(see Display 7.14). Alternatively, we can use the bootstrap command

Generalized linear models

No. of obs = 146

Optimization : ML: Newton-Raphson

Residual df = 143

Scale parameter = 1

Deviance = 1823.481292

(1/df) Deviance = 12.75162

Pearson = 2091.29704

(1/df) Pearson = 14.62445

Variance function: V(u) = u

[Poisson]

Link function : g(u) = ln(u)

[Log]

Standard errors : OIM

Log likelihood = -1205.979185

AIC = 16.56136

BIC = 1110.825545

days	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
class	.1637288	.0760048	2.15	0.031	.0147622	.3126954
ethnic	-.5477436	.1599613	-3.42	0.001	-.861262	-.2342252
_cons	3.168776	.3170159	10.00	0.000	2.547437	3.790116

(Standard errors scaled using square root of Pearson X2-based dispersion)

Display 7.14

`bs`, followed by the estimation command in quotes, followed, in quotes, by expressions for the estimates for which bootstrap standard errors are required. To be on the safe side, we will ask for 500 bootstrap samples using the option `reps(500)`. First, we set the seed of the pseudorandom number generator using the `set seed` command so that we can run the sequence of commands again in the future and obtain the same results.

```
set seed 12345678
bs "poisson days class ethnic" "_b[class] _b[ethnic]", /*
  */ reps(500)
```

(see Display 7.15). This compares very well with the robust Poisson and quasi-likelihood results. Note that it is also possible to use bootstrapping within the `glm` command using the `bstrap` and `brep(#)` options.

We could also model overdispersion by assuming a *random effects* model where each child has an unobserved, random proneness to be absent from school. This proneness (called *frailty* in a medical context) multiplies the rate predicted by the covariates so that some children have higher or lower rates of absence from school than other children

command:	glm days class ethnic , fam(pois) link(log)					
statistics:	_bs_1	=	_b[class]			
	_bs_2	=	_b[ethnic]			
Bootstrap statistics				Number of obs	=	146
				Replications	=	500

Variable	Reps	Observed	Bias	Std. Err.	[95% Conf. Interval]		
_bs_1	500	.1637288	.0026568	.0708082	.0246099	.3028477	(N)
					.0203793	.3041895	(P)
					.014529	.2918807	(BC)
_bs_2	500	-.5477436	-.0126206	.1462108	-.8350083	-.260479	(N)
					-.8556656	-.2822307	(P)
					-.8363549	-.2529301	(BC)

Note: N = normal
P = percentile
BC = bias-corrected

Display 7.15

with the same covariates. The observed counts are assumed to have a Poisson distribution conditional on the random effects. If the frailties are assumed to have a log gamma distribution, then the (marginal) distribution of the counts has a negative binomial distribution. The negative binomial model may be fitted using `nbreg` as follows:

```
nbreg days class ethnic, nolog
```

(see Display 7.16). Alternatively, the same model can be estimated using `glm` with `family(nbinomial)`.

All four methods of analyzing the data lead to the same conclusions. The Poisson model is a special case of the negative binomial model with $\alpha = 0$. The likelihood ratio test for α is therefore a test of the negative binomial against the Poisson distribution. The very small p -value ‘against Poisson’ indicates that there is significant overdispersion. (Note that the test is too conservative since the null hypothesis is on the boundary of the parameter space, see e.g., Snijders and Bosker, 1999.)

7.4 Exercises

1. Calculate the F -statistic and difference in deviance for adding `status` to a model already containing `cond` for the data in `slim.dat`.
2. Fit the model using `status` as the only explanatory variable, using robust standard errors. How does this compare with a t -test with

Negative binomial regression	Number of obs	=	146
	LR chi2(2)	=	15.77
	Prob > chi2	=	0.0004
Log likelihood = -551.24625	Pseudo R2	=	0.0141

days	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
class	.1505165	.0732832	2.05	0.040	.0068841	.2941489
ethnic	-.5414185	.1578378	-3.43	0.001	-.8507748	-.2320622
_cons	3.19392	.3217681	9.93	0.000	2.563266	3.824574
/lnalpha	-.1759664	.1243878			-.4197619	.0678292
alpha	.8386462	.1043173			.6572032	1.070182

Likelihood-ratio test of alpha=0: chibar2(01) = 1309.47 Prob>=chibar2 = 0.000

Display 7.16

unequal variances?

3. Carry out a significance test for the interaction between `class` and `ethnic` for the data in `quine.dta`.
4. Excluding the potential outlier (case 72), fit the model with explanatory variables `ethnic` and `class`.
5. Dichotomize days absent from school by classifying 14 days or more as frequently absent. Analyze this new response using both logistic and probit links and the binomial family.
6. Repeat the analyses with the `robust` option, and compare the robust standard errors with the standard errors obtained using bootstrapping.

See also the Exercises in [Chapter 11](#).

Chapter 8

Summary Measure Analysis of Longitudinal Data: The Treatment of Post-Natal Depression

8.1 Description of data

The data set to be analyzed in this chapter originates from a clinical trial of the use of estrogen patches in the treatment of postnatal depression; full details are given in Gregoire *et al.* (1996). In total, 61 women with major depression, which began within 3 months of child-birth and persisted for up to 18 months postnatally, were allocated randomly to the active treatment or a placebo (a dummy patch); 34 received the former and the remaining 27 received the latter. The women were assessed pretreatment and monthly for six months after treatment on the Edinburgh postnatal depression scale (EPDS), higher values of which indicate increasingly severe depression. The data are shown in [Table 8.1](#); a value of -9 in this table indicates that the observation is missing. The non-integer depression scores result from missing questionnaire items (in this case the average of all available items was multiplied by the total number of items). The variables are

- **subj**: patient identifier
- **group**: treatment group (1: estrogen patch, 0: placebo patch)
- **pre**: pretreatment or baseline EPDS depression score

■ dep1 to dep6: EPDS depression scores for visits 1 to 6

The main question of interest here is whether the estrogen patch is effective at reducing post-natal depression compared with the placebo.

Table 8.1 Data in depress.dat

subj	group	pre	dep1	dep2	dep3	dep4	dep5	dep6
1	0	18	17	18	15	17	14	15
2	0	27	26	23	18	17	12	10
3	0	16	17	14	-9	-9	-9	-9
4	0	17	14	23	17	13	12	12
5	0	15	12	10	8	4	5	5
6	0	20	19	11.54	9	8	6.82	5.05
7	0	16	13	13	9	7	8	7
8	0	28	26	27	-9	-9	-9	-9
9	0	28	26	24	19	13.94	11	9
10	0	25	9	12	15	12	13	20
11	0	24	14	-9	-9	-9	-9	-9
12	0	16	19	13	14	23	15	11
13	0	26	13	22	-9	-9	-9	-9
14	0	21	7	13	-9	-9	-9	-9
15	0	21	18	-9	-9	-9	-9	-9
16	0	22	18	-9	-9	-9	-9	-9
17	0	26	19	13	22	12	18	13
18	0	19	19	7	8	2	5	6
19	0	22	20	15	20	17	15	13.73
20	0	16	7	8	12	10	10	12
21	0	21	19	18	16	13	16	15
22	0	20	16	21	17	21	16	18
23	0	17	15	-9	-9	-9	-9	-9
24	0	22	20	21	17	14	14	10
25	0	19	16	19	-9	-9	-9	-9
26	0	21	7	4	4.19	4.73	3.03	3.45
27	0	18	19	-9	-9	-9	-9	-9
28	1	21	13	12	9	9	13	6
29	1	27	8	17	15	7	5	7
30	1	15	8	12.27	10	10	6	5.96
31	1	24	14	14	13	12	18	15
32	1	15	15	16	11	14	12	8
33	1	17	9	5	3	6	0	2
34	1	20	7	7	7	12	9	6
35	1	18	8	1	1	2	0	1
36	1	28	11	7	3	2	2	2
37	1	21	7	8	6	6.5	4.64	4.97
38	1	18	8	6	4	11	7	6
39	1	27.46	22	27	24	22	24	23
40	1	19	14	12	15	12	9	6
41	1	20	13	10	7	9	11	11
42	1	16	17	26	-9	-9	-9	-9
43	1	21	19	9	9	12	5	7
44	1	23	11	7	5	8	2	3
45	1	23	16	13	-9	-9	-9	-9
46	1	24	16	15	11	11	11	11
47	1	25	20	18	16	9	10	6
48	1	22	15	17.57	12	9	8	6.5
49	1	20	7	2	1	0	0	2
50	1	20	12.13	8	6	3	2	3

Data in depress.dat (continued)

51	1	25	15	24	18	15.19	13	12.32
52	1	18	17	6	2	2	0	1
53	1	26	1	18	10	13	12	10
54	1	20	27	13	9	8	4	5
55	1	17	20	10	8.89	8.49	7.02	6.79
56	1	22	12	-9	-9	-9	-9	-9
57	1	22	15.38	2	4	6	3	3
58	1	23	11	9	10	8	7	4
59	1	17	15	-9	-9	-9	-9	-9
60	1	22	7	12	15	-9	-9	-9
61	1	26	24	-9	-9	-9	-9	-9

8.2 The analysis of longitudinal data

The data in [Table 8.1](#) consist of repeated observations over time on each of the 61 patients; such data are generally referred to as *longitudinal data*, panel data or repeated measurements, and as *cross-sectional time-series* in Stata. There is a large body of methods that can be used to analyze longitudinal data, ranging from the simple to the complex. Some useful references are Diggle *et al.* (2002), Everitt (1995), and Hand and Crowder (1996). In this chapter we concentrate on the following approaches:

- Graphical displays
- Summary measure or response feature analysis

In the next chapter two chapters, more formal modeling techniques will be applied to the data.

8.3 Analysis using Stata

Assuming the data are in an ASCII file, `depress.dat`, as listed in [Table 8.1](#), they may be read into Stata for analysis using the following instructions:

```
infile subj group pre dep1 dep2 dep3 dep4 dep5 dep6 /*
      */ using depress.dat
mvdecode _all, mv(-9)
```

The second of these instructions converts values of `-9` in the data to missing values.

It is useful to begin examination of these data using the `summarize` procedure to calculate means, variances etc., within each of the two treatment groups:

```
summarize pre-dep6 if group==0
```

(see Display 8.1).

Variable	Obs	Mean	Std. Dev.	Min	Max
pre	27	20.77778	3.954874	15	28
dep1	27	16.48148	5.279644	7	26
dep2	22	15.88818	6.124177	4	27
dep3	17	14.12882	4.974648	4.19	22
dep4	17	12.27471	5.848791	2	23
dep5	17	11.40294	4.438702	3.03	18
dep6	17	10.89588	4.68157	3.45	20

Display 8.1

```
summarize pre-dep6 if group==1
```

(see Display 8.2). There is a general decline in the depression score

Variable	Obs	Mean	Std. Dev.	Min	Max
pre	34	21.24882	3.574432	15	28
dep1	34	13.36794	5.556373	1	27
dep2	31	11.73677	6.575079	1	27
dep3	29	9.134138	5.475564	1	24
dep4	28	8.827857	4.666653	0	22
dep5	28	7.309286	5.740988	0	24
dep6	28	6.590714	4.730158	1	23

Display 8.2

over time in both groups, with the values in the active treatment group appearing to be consistently lower.

8.3.1 Graphical displays

A useful preliminary step in the analysis of longitudinal data is to graph the observations in some way. The aim is to highlight two particular aspects of the data, namely, how they evolve over time and how the measurements made at different times are related. A number of graphical displays can be used, including:

- separate plots of each subject's responses against time, differentiating in some way between subjects in different groups
- boxplots of the observations at each time point by treatment group
- a plot of means and standard errors by treatment group for every time point
- a scatterplot matrix of the repeated measurements

To begin, plot the required scatterplot matrix, identifying treatment groups with the labels 0 and 1, using

```
graph matrix pre-dep6, mlabel(group) msymbol(none) /*
    */ mlabposition(0)
```

The resulting plot is shown in Figure 8.1. The most obvious feature of this diagram is the increasingly strong relationship between the measurements of depression as the time interval between them decreases. This has important implications for the models appropriate for longitudinal data, as we will see in [Chapter 10](#).

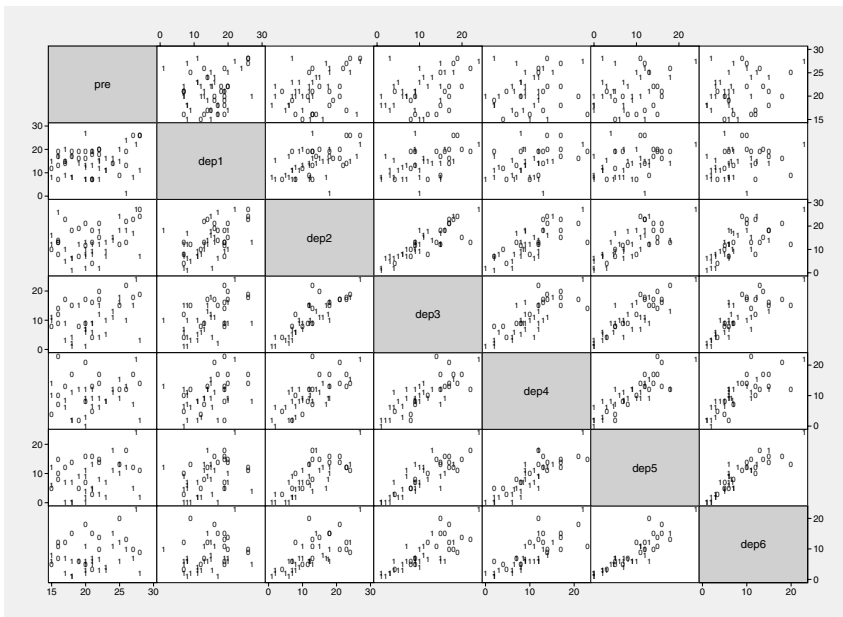


Figure 8.1: Scatter-plot matrix for depression scores at six visits.

To obtain the other graphs mentioned above, the dataset needs to be restructured from its present wide form (one column for each visit)

to the long form (one row for each visit) using the `reshape` command. Before running `reshape`, we will preserve the data using the `preserve` command so that they can later be restored using `restore`:

```
preserve
reshape long dep, i(subj) j(visit)
list in 1/13, clean
```

The first 13 observations of the data in long form are shown in Display 8.3.

	subj	visit	group	pre	dep
1.	1	1	0	18	17
2.	1	2	0	18	18
3.	1	3	0	18	15
4.	1	4	0	18	17
5.	1	5	0	18	14
6.	1	6	0	18	15
7.	2	1	0	27	26
8.	2	2	0	27	23
9.	2	3	0	27	18
10.	2	4	0	27	17
11.	2	5	0	27	12
12.	2	6	0	27	10
13.	3	1	0	16	17

Display 8.3

We will first plot the subjects' individual response profiles over the visits separately for each group using the `by()` option. To obtain the correct group labels with the `by()` option we must first label the values of `group`:

```
label define treat 0 "Placebo" 1 "Estrogen"
label values group treat
```

In each graph we want to connect the points belonging to a given subject, but avoid connecting points of different subjects. A simple way of achieving this is to use the `connect(ascending)` option. Before plotting, the data need to be sorted by the grouping variable and by the *x* variable (here `visit`):

```
sort group subj visit
twoway connected dep visit, connect(ascending) by(group) /*
    */ ytitle(Depression) xlabel(1/6)
```

The `connect(ascending)` option connects points only so long as `visit` is ascending. For the first subject (`subj=1`) this is true; but for the second subject, `visit` begins at 1 again, so the last point for subject one is

not connected with the first point for subject two. The remaining points for this subject are, however, connected and so on. The `xlabel()` option was used here to make the axis range start at 1 instead of 0. The diagram is shown in Figure 8.2. (Some points are connected at visit one; this happens when successive subjects have missing data for all subsequent visits so that `visit` does not decrease when `subj` increases.) The individual plots reflect the general decline in the depression scores over time indicated by the means obtained using the `summarize` command; there is, however, considerable variability. The phenomenon of ‘tracking’ is apparent whereby some individuals have consistently higher values than other individuals, leading to within-subject correlations. Notice that some profiles are not complete because of missing values.

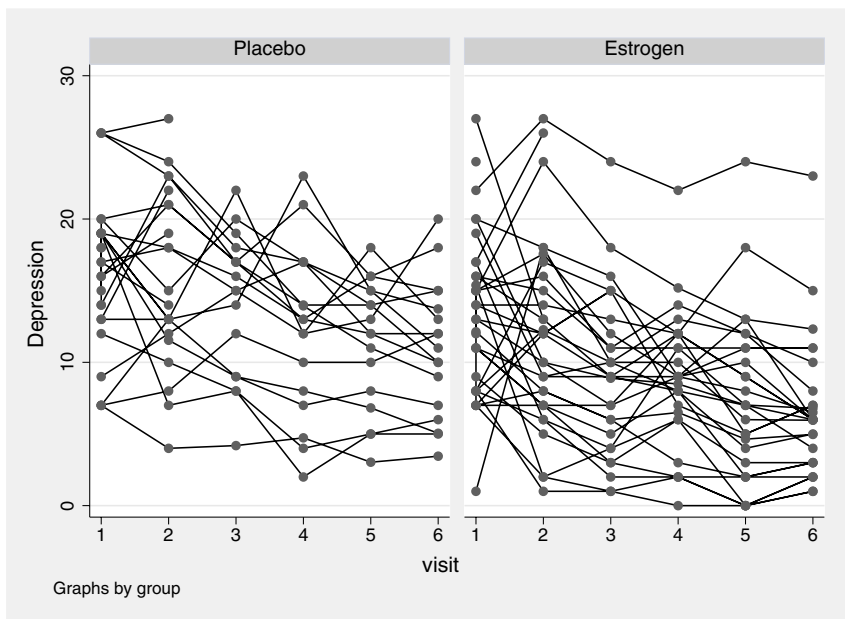


Figure 8.2: Individual response profiles by treatment group.

To obtain the boxplots of the depression scores at each visit for each treatment group, the following instruction can be used:

```
graph box dep, over(visit) over(group, /*
  */ relabel(1 "Placebo group" 2 "Estrogen group"))
```


Here the `over()` options specify two grouping variables, `visit` and `group` to plot the distributions by visit within groups. The `relabel()` option is used to define labels for the groups. Here '1' refers to the first level of `group` (0 in this case) and '2' to the second. The resulting graph is shown in Figure 8.3. Again, the general decline in depression scores in both treatment groups can be seen and, in the active treatment group, there is some evidence of outliers which may need to be examined. (Figure 8.2 shows that four of the outliers are due to one subject whose response profile lies above the others.)

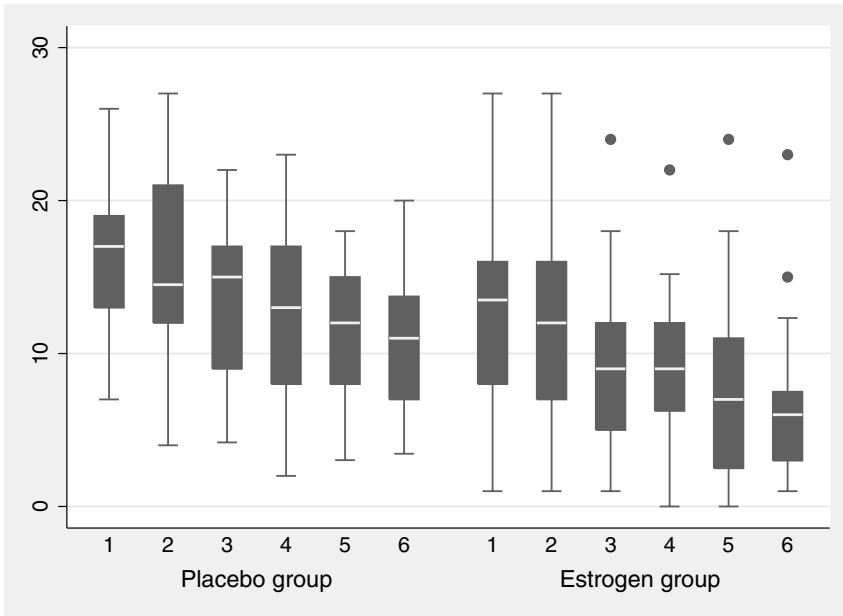


Figure 8.3: Boxplots for six visits by treatment group.

A plot of the mean profiles of each treatment group, which includes information about the standard errors of each mean, can be obtained using the `collapse` instruction that produces a dataset consisting of selected summary statistics. Here, we need the mean depression score on each visit for each group, the corresponding standard deviations, and a count of the number of observations on which these two statistics are based.

```
collapse (mean) dep (sd) sddep=dep (count) n=dep, /*
  */ by(visit group)
list in 1/10
```

(see Display 8.4). The mean value is now stored in `dep`; but since more

	visit	group	dep	sddep	n
1.	1	0	16.48148	5.279644	27
2.	1	1	13.36794	5.556373	34
3.	2	0	15.88818	6.124177	22
4.	2	1	11.73677	6.575079	31
5.	3	0	14.12882	4.974648	17
6.	3	1	9.134138	5.475564	29
7.	4	0	12.27471	5.848791	17
8.	4	1	8.827857	4.666653	28
9.	5	0	11.40294	4.438702	17
10.	5	1	7.309286	5.740988	28

Display 8.4

than one summary statistic for the depression scores were required, the remaining statistics were given new names in the `collapse` instruction.

The required mean and standard error plots can now be produced as follows:

```
sort group
gen high = dep + 2*sddep/sqrt(n)
gen low = dep - 2*sddep/sqrt(n)
twoway (rarea low high visit, bfcOLOR(gs12)) /*
      /* (connected dep visit, mcolor(black) /*
      /* clcolor(black)), by(group) /*
      /* legend(order(1 "95% CI" 2 "mean depression"))
```

Here `twoway rarea` produces a shaded area between the lines `low` versus `visit` and `high` versus `visit`, the 95% confidence limits for the mean. It is important that the line for the mean is plotted after the shaded area because it would otherwise be hidden underneath it. The resulting diagram is shown in [Figure 8.4](#).

8.3.2 Response feature analysis

A relatively straightforward approach to the analysis of longitudinal data is that involving the use of *summary measures*, sometimes known as *response feature analysis*. The responses of each subject are used to construct a single number that characterizes some relevant aspect of the subject's response profile. (In some situations more than a single summary measure may be required.) The summary measure needs to be chosen before the analysis of the data. The most commonly used measure is the mean of the responses over time because many inves-

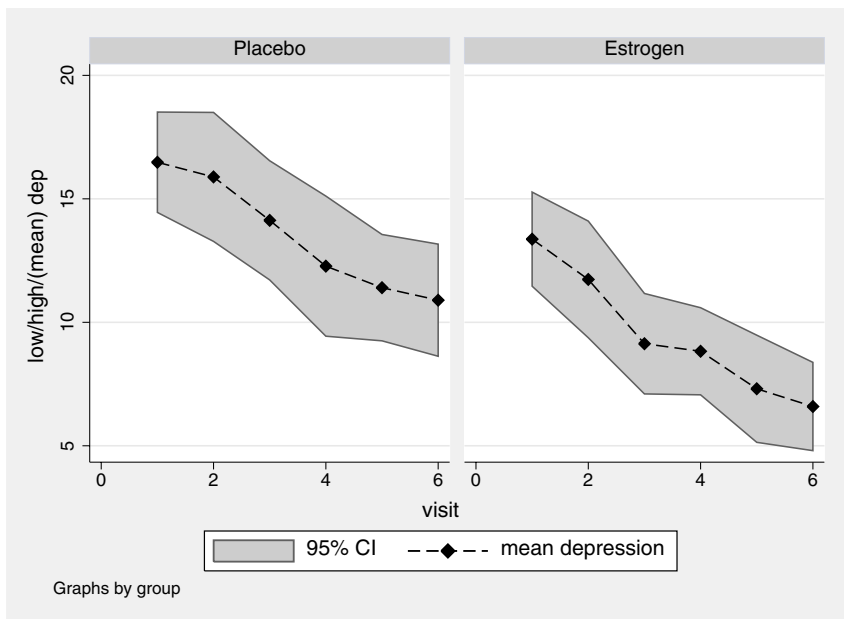


Figure 8.4: Mean and standard error plots; the shaded areas represent ± 2 standard errors.

tigations, e.g., clinical trials, are most concerned with differences in overall levels rather than more subtle effects. Other possible summary measures are listed in Matthews *et al.* (1990) and are shown here in Table 8.2.

Having identified a suitable summary measure, the analysis of the data generally involves the application of a simple univariate test (usually a *t*-test or its nonparametric equivalent) for group differences on the single measure now available for each subject. For the estrogen patch trial data, the mean over time seems an obvious summary measure. The mean of all non-missing values is obtained (after restoring the data) using

```
restore
egen avg=rmean(dep1 dep2 dep3 dep4 dep5 dep6)
```

The differences between these means may be tested using a *t*-test assuming equal variances in the populations:

```
ttest avg, by(group)
```

(see Display 8.5). The assumption of equal variances can be relaxed using the `unequal` option:

```
ttest avg, by(group) unequal
```

Table 8.2 Response features suggested in Matthews *et al.* (1990)

Type of data	Property to be compared between groups	Summary measure
Peaked	overall value of response	mean or area under curve
Peaked	value of most extreme response	maximum (minimum)
Peaked	delay in response	time to maximum or minimum
Growth	rate of change of response	linear regression coefficient
Growth	final level of response	final value or (relative) difference between first and last
Growth	delay in response	time to reach a particular value

Two-sample t test with equal variances

Group	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf. Interval]	
0	27	14.75605	.8782852	4.563704	12.95071	16.56139
1	34	10.55206	.9187872	5.357404	8.682772	12.42135
combined	61	12.41284	.6923949	5.407777	11.02785	13.79784
diff		4.20399	1.294842		1.613017	6.794964

Degrees of freedom: 59

Ho: mean(0) - mean(1) = diff = 0

Ha: diff < 0	Ha: diff != 0	Ha: diff > 0
t = 3.2467	t = 3.2467	t = 3.2467
P < t = 0.9990	P > t = 0.0019	P > t = 0.0010

Display 8.5

(see Display 8.6). In each case the conclusion is that the mean de-

Two-sample t test with unequal variances

Group	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf. Interval]	
0	27	14.75605	.8782852	4.563704	12.95071	16.56139
1	34	10.55206	.9187872	5.357404	8.682772	12.42135
combined	61	12.41284	.6923949	5.407777	11.02785	13.79784
diff		4.20399	1.271045		1.660343	6.747637

Satterthwaite's degrees of freedom: 58.6777

Ho: mean(0) - mean(1) = diff = 0

Ha: diff < 0	Ha: diff != 0	Ha: diff > 0
t = 3.3075	t = 3.3075	t = 3.3075
P < t = 0.9992	P > t = 0.0016	P > t = 0.0008

Display 8.6

pression score is substantially larger in the estrogen group than the placebo group. The difference in mean depression scores is estimated as 4.3 with a 95% confidence interval from 1.7 to 6.7.

The summary measure approach to longitudinal data has a number of advantages:

- Appropriate choice of summary measure ensures that the analysis is focused on relevant and interpretable aspects of the data,
- The method is easy to explain and intuitive, and
- To some extent missing and irregularly spaced observations can be accommodated.

However, the method is somewhat *ad hoc*, particularly in its treatment of missing data. For instance, if the summary measure is a mean, but there is actually a decline in the response over time, then the mean of all available data will overestimate the mean for those who dropped out early. Furthermore, response feature analysis treats all summaries as equally precise even if some are based on fewer observations due to missing data. In the next two chapters we will therefore discuss more formal approaches to longitudinal data, random effects modeling, and generalized estimating equations.

8.4 Exercises

1. How would you produce boxplots corresponding to those shown in [Figure 8.3](#) using the data in the wide form?
2. Compare the results of the t -tests given in the text with the corresponding t -tests calculated only for those subjects having observations on all six post-randomization visits.
3. Repeat the summary measures analysis described in the text using the maximum over time instead of the mean (see `help egen`).
4. Test for differences in the mean over time controlling for the baseline measurement using
 - a. a change score defined as the difference between the mean over time and the baseline measurement, and
 - b. analysis of covariance of the mean over time using the baseline measurement as a covariate.

See also Exercises in [Chapter 9](#).

Chapter 9

Random Effects Models: Thought Disorder and Schizophrenia

9.1 Description of data

In this chapter we will analyze data from the Madras Longitudinal Schizophrenia Study in which patients were followed up monthly after their first hospitalization for schizophrenia. The study is described in detail in Thara *et al.* (1994). Here we use a subset of the data analyzed by Diggle *et al.* (2002), namely data on thought disorder (1: present, 0: absent) at 0, 2, 6, 8, and 10 months after hospitalization on women only. The thought disorder responses are given as y_0 to y_{10} in [Table 9.1](#) where a ‘.’ indicates a missing value. The variable **early** is a dummy variable for early onset of disease (1: age-of-onset less than 20 years, 0: age-of-onset 20 years or above). An important question here is whether the course of illness differs between patients with early and late onset. We will also reanalyze the post-natal depression data described in the previous chapter.

9.2 Random effects models

The data listed in Table 9.1 consist of repeated observations on the same subject taken over time and are a further example of a set of *longitudinal data*. During the last decades, statisticians have considerably

Table 9.1 Data in `madras.dta`

id	early	y0	y2	y6	y8	y10
1	0	1	1	0	0	0
6	1	0	0	0	0	0
10	0	1	1	0	0	0
13	0	0	0	0	0	0
14	0	1	1	1	1	1
15	0	1	1	0	0	0
16	0	1	0	1	0	0
22	0	1	1	0	0	0
23	0	0	0	1	0	0
25	1	0	0	0	0	0
27	1	1	1	1	0	1
28	0	0	0	.	.	.
31	1	1	1	0	0	0
34	0	1	1	0	0	0
36	1	1	1	0	0	0
43	0	1	1	1	0	0
44	0	0	1	0	0	0
45	0	1	1	0	1	0
46	0	1	1	1	0	0
48	0	0	0	0	0	0
50	0	0	0	1	1	1
51	1	0	1	1	0	0
52	0	1	1	0	1	0
53	0	1	0	0	0	0
56	1	1	0	0	0	0
57	0	0	0	0	0	0
59	0	1	1	0	0	0
61	0	0	0	0	0	0
62	1	1	1	0	0	0
65	1	0	0	0	0	0
66	0	0	0	0	0	0
67	0	0	1	0	0	0
68	0	1	1	1	1	1
71	0	1	1	1	0	0
72	0	1	0	0	0	0
75	1	1	0	0	.	.
76	0	0	1	.	.	.
77	1	0	0	0	0	0
79	0	1
80	0	1	1	0	0	0
85	1	1	1	1	0	0
86	0	0	1	.	.	.
87	0	1	0	0	0	.
90	0	1	1	0	0	0

enriched the methodology available for the analysis of such data (see Lindsey, 1999, or Diggle *et al.* 2002) and many of these developments are implemented in Stata.

9.2.1 *Normally distributed responses*

Longitudinal data require special methods of analysis because the responses at different time points on the same individual may not be independent even after conditioning on the covariates. For a linear regression model this means that the residuals for the same individual are correlated. We can model these residual correlations by partitioning the total residual for subject i at time point j into a subject-specific *random intercept* or *permanent component* u_i which is constant over time plus a residual ϵ_{ij} which varies randomly over time. The resulting *random intercept model* can be written as

$$y_{ij} = \beta^T \mathbf{x}_{ij} + u_i + \epsilon_{ij}. \quad (9.1)$$

The random intercept and residual are each assumed to be independently normally distributed with zero means and constant variances τ^2 and σ^2 , respectively. Furthermore, these random terms are assumed to be independent of each other and the covariates \mathbf{x}_{ij} . (It should be noted that moment-based approaches do not require normality assumptions, see e.g. Wooldridge (2002), Section 10.4.)

The random intercept model implies that the total residual variance is

$$\text{Var}(u_i + \epsilon_{ij}) = \tau^2 + \sigma^2.$$

Due to this decomposition of the total residual variance into a between-subject component τ^2 and a within-subject component σ^2 , the model is sometimes referred to as a *variance component* model. The covariance between the total residuals at any two time points j and j' on the same subject i is

$$\text{Cov}(u_i + \epsilon_{ij}, u_i + \epsilon_{ij'}) = \tau^2.$$

Note that these covariances are induced by the shared random intercept: for subjects with $u_i > 0$, the total residuals will tend to be greater than the mean (0) and for subjects with $u_i < 0$ they will tend to be less than the mean.

It follows from the two relations above that the residual correlations

are given by

$$\text{Cor}(u_i + \epsilon_{ij}, u_i + \epsilon_{ij'}) = \frac{\tau^2}{\tau^2 + \sigma^2}.$$

This *intraclass correlation* can be interpreted as the proportion of the total residual variance (denominator) that is due to residual variability between subjects (numerator).

The random intercept can be interpreted as the combined effect of all unobserved subject-specific covariates, often referred to as *unobserved heterogeneity*. The random intercepts represent individual differences in the overall mean level of the response after controlling for covariates. Random coefficients of covariates can be used to allow for between-subject heterogeneity in the *effects* of the covariates. For instance, in longitudinal data, the shape of the response profile may vary between subjects in addition to variability in its vertical position. If the overall shape is linear in time t_{ij} , subjects may differ randomly in their slopes giving a model of the form

$$y_{ij} = \beta^T \mathbf{x}_{ij} + u_{0i} + u_{1i}t_{ij} + \epsilon_{ij}. \quad (9.2)$$

Here u_{0i} is a random intercept and u_{1i} a random coefficient or slope of t_{ij} . These random effects are assumed to have a bivariate normal distribution with zero means, variances τ_0^2 and τ_1^2 and covariance τ_{01} . They are furthermore assumed to be uncorrelated across subjects and uncorrelated with ϵ_{ij} or any of the covariates. If the covariate vector \mathbf{x}_{ij} includes t_{ij} , the corresponding fixed coefficient β_t represents the mean coefficient of time whereas the random slope u_{1i} represents the deviation from the mean coefficient for subject i . (Not including t_{ij} in the fixed part of the model would imply that the mean slope is zero.) The model can also include nonlinear functions of t_{ij} , typically powers of t_{ij} , whose coefficients may be fixed or random.

The total residual in (9.2) is $u_{0i} + u_{1i}t_{ij} + \epsilon_{ij}$ with variance

$$\text{Var}(u_{0i} + u_{1i}t_{ij} + \epsilon_{ij}) = \tau_0^2 + 2\tau_{01}t_{ij} + \tau_1^2t_{ij}^2 + \sigma^2$$

which is no longer constant over time but *heteroskedastic*. Similarly, the covariance between two total residuals of the same subject,

$$\text{Cov}(u_{0i} + u_{1i}t_{ij} + \epsilon_{ij}, u_{0i} + u_{1i}t_{ij'} + \epsilon_{ij'}) = \tau_0^2 + \tau_{01}(t_{ij} + t_{ij'}) + \tau_1^2t_{ij}t_{ij'},$$

is not constant over time. It should also be noted that both the random intercept variance and the correlation between the random coefficient and random intercept depend on the location of t_{ij} , i.e., re-estimating the model after adding a constant to t_{ij} will lead to different estimates.

General terms for random intercept or random coefficient models are *random effects* models, *mixed effects* or *mixed* models, where ‘mixed’ refers to the presence of both fixed effects β and random effects u_{0i} and u_{1i} . The models are also *hierarchical* or *multilevel* since the elementary observations at the individual time points (level 1) are nested in subjects (level 2). The models discussed in this chapter are appropriate for any kind of clustered or two-level data, not just longitudinal. Other examples of two-level data are people in families, households, neighborhoods, cities, schools, hospitals, firms, etc. In all these types of data, we can generally not assume that responses for subjects in the same cluster are independent after controlling for covariates because there is likely to be unobserved heterogeneity between clusters.

9.2.2 *Non-normal responses*

For non-normal responses (for example, binary responses) we can extend the generalized linear model discussed in Chapter 7 by introducing a random intercept u_i into the linear predictor,

$$\eta_{ij} = \beta^T \mathbf{x}_{ij} + u_i, \quad (9.3)$$

where the u_i are independently normally distributed with mean zero and variance τ^2 . (We have encountered a similar model in Chapter 7, namely the negative binomial model with a log link and Poisson distribution where u_i has a log-gamma distribution, and there is only one observation per subject.) We can further extend the random intercept model to include random coefficients as we did in the previous section.

Unfortunately, such *generalized linear mixed models* are difficult to estimate. This is because the likelihood involves integrals over the random effects distribution and these integrals generally do not have closed forms. Stata uses numerical integration by Gauss-Hermite quadrature for random intercept models. A user-written program `gllamm` can be used to estimate random coefficient models using either quadrature or an improvement of this method known as *adaptive quadrature*. The program can also be used to estimate multilevel models. Note that approximate methods such as penalized quasilielihood (e.g., Breslow and Clayton, 1993) and its refinements do not tend to work well for data with dichotomous responses and small cluster sizes such as the thought disorder data (see also Rabe-Hesketh *et al.*, 2002).

An important problem with many longitudinal data sets is the occurrence of dropouts, e.g., subjects failing to complete all scheduled visits in the post-natal depression data. A taxonomy of dropouts is given in Diggle *et al.* (2002). Fortunately, maximum likelihood estimation is consistent as long as the data are missing at random (MAR),

that is, the probability of missingness does not depend on the values that are missing. For example, if the model is correctly specified, we obtain consistent parameter estimates even if the probability of dropping out depends on the responses at earlier time points.

Useful books on random effects modeling include Snijders and Bosker (1999), Verbeke and Molenberghs (2000), Goldstein (2003), and Skroddal and Rabe-Hesketh (2004), as well general books on longitudinal data such as Lindsey (1999).

9.3 Analysis using Stata

9.3.1 *Post-natal depression data*

As an example of continuous responses, we first consider the post-natal depression data analyzed in the previous chapter. The data are read using

```
infile subj group pre dep1 dep2 dep3 dep4 dep5 dep6 /*
      */ using depress.dat
```

All responses must be stacked in a single variable, including the baseline score `pre`. This is achieved by first renaming `pre` to `dep0` and then using the `reshape` command:

```
rename pre dep0
reshape long dep, i(subj) j(visit)
```

We also define value labels for `group` and change ‘-9’ to missing values

```
label define treat 0 "Placebo" 1 "Estrogen"
label values group treat
mvdecode _all, mv(-9)
```

We now estimate a random intercept model using `xtreg`. (Note that commands for longitudinal data have the prefix `xt` in Stata which stands for *cross-sectional time series*). First assume that the mean depression score declines linearly from baseline with different slopes in the two groups:

```
gen gr_vis = group*visit
xtreg dep group visit gr_vis, i(subj) mle
```

The syntax is the same as for `regress` except that the `i()` option is used to specify the cluster identifier, here `subj`, and the `mle` option to obtain maximum likelihood estimates.

The estimates of the ‘fixed’ regression coefficients that do not vary over individuals are given in the first table in Display 9.1 whereas the estimates of the standard deviations τ of the random intercept and σ

```

Fitting constant-only model:
Iteration 0:  log likelihood = -1200.5805
Iteration 1:  log likelihood = -1166.1333
Iteration 2:  log likelihood = -1159.0981
Iteration 3:  log likelihood = -1158.585
Iteration 4:  log likelihood = -1158.5807

Fitting full model:
Iteration 0:  log likelihood = -1045.9826
Iteration 1:  log likelihood = -1045.7124
Iteration 2:  log likelihood = -1045.7117

Random-effects ML regression
Group variable (i): subj
Random effects u_i ~ Gaussian

Number of obs      =      356
Number of groups   =      61
Obs per group: min =       2
                  avg =      5.8
                  max =       7

LR chi2(3)         =     225.74
Prob > chi2        =     0.0000

Log likelihood = -1045.7117

```

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-1.644653	1.163462	-1.41	0.157	-3.924996	.6356901
visit	-1.531905	.1736977	-8.82	0.000	-1.872346	-1.191464
gr_vis	-.5564469	.2220225	-2.51	0.012	-.9916031	-.1212908
_cons	19.29632	.8717659	22.13	0.000	17.58769	21.00495
/sigma_u	3.560969	.3949951	9.02	0.000	2.786793	4.335145
/sigma_e	3.948191	.161907	24.39	0.000	3.630859	4.265523
rho	.4485701	.0598845			.3350721	.5664725

Likelihood-ratio test of sigma_u=0: chibar2(01)= 114.03 Prob>=chibar2 = 0.000

Display 9.1

of the residuals are given under `sigma_u` and `sigma_e` in the second table. The intraclass correlation `rho` in the third table is estimated as 0.45. The fixed regression coefficients show that the average decrease in depression score is 1.53 per visit in the placebo group and $1.53 + 0.56$ per visit in the estrogen group. We can obtain the estimated slope of time in the estrogen group with its p -value and confidence interval using `lincom`:

```
lincom visit + gr_vis
```

(see Display 9.2).

```
( 1) [dep]visit + [dep]gr_vis = 0
```

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
(1)	-2.088352	.1384264	-15.09	0.000	-2.359663	-1.817041

Display 9.2

The model assumes that the effect of `visit` is linear. However, it may well be that the depression score gradually levels off, remaining stable after some period of time. We can investigate this by adding a quadratic term of `visit`:

```
gen vis2 = visit^2
xtreg dep group visit gr_vis vis2, i(subj) mle nolog
```

The p -value for `vis2` in Display 9.3 suggests that the average curve is not linear but that the downward trend declines over time. To picture the mean curve, we now plot it together with the observed individual response profiles:

```
predict pred0, xb
sort subj visit
twoway (line pred0 visit, connect(ascending) /*
      */ clwidth(thick)) (line dep visit, /*
      */ connect(ascending) clpat(dash)), by(group) /*
      */ ytitle(Depression) legend(order(1 "Fitted mean" /*
      */ 2 "Observed scores"))
```

giving the graph shown in [Figure 9.1](#) which suggests that the response curves tend to level off.

Extending the model to include a random coefficient of `visit` requires the user-contributed program `gllamm` (for generalized linear la-

Random-effects ML regression	Number of obs	=	356
Group variable (i): subj	Number of groups	=	61
Random effects u_i ~ Gaussian	Obs per group: min	=	2
	avg	=	5.8
	max	=	7
	LR chi2(4)	=	268.39
Log likelihood = -1024.3838	Prob > chi2	=	0.0000

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-1.471391	1.139394	-1.29	0.197	-3.704563	.7617806
visit	-3.787308	.3710888	-10.21	0.000	-4.514628	-3.059987
gr_vis	-.5848499	.2073966	-2.82	0.005	-.9913396	-.1783601
vis2	.3851916	.0569336	6.77	0.000	.2736038	.4967793
_cons	20.91077	.8860177	23.60	0.000	19.17421	22.64734
/sigma_u	3.584665	.3869129	9.26	0.000	2.826329	4.343
/sigma_e	3.678709	.1508811	24.38	0.000	3.382987	3.97443
rho	.4870545	.0589037			.3737279	.6014435

Likelihood-ratio test of sigma_u=0: chibar2(01)= 133.51 Prob>=chibar2 = 0.000

Display 9.3

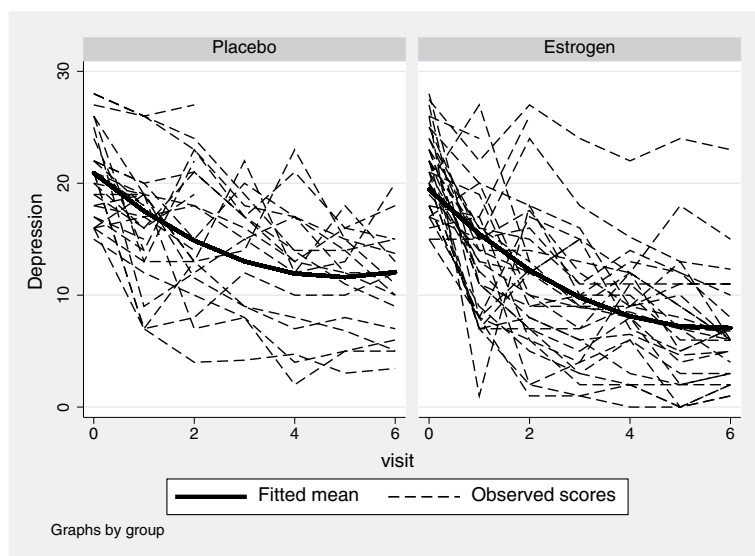


Figure 9.1: Response profiles and fitted mean curves by treatment group.

tent and mixed models) described in Rabe-Hesketh *et al.* (2002) and (2004b) (see also the webpage www.gllamm.org). The program can be obtained from the SSC archive using

```
ssc install gllamm
```

We will first re-estimate the random intercept model using `gllamm`:

```
gllamm dep group visit gr_vis vis2, i(subj) adapt
```

The syntax is as for `xtreg` except that the `mle` option is not required since `gllamm` always uses maximum likelihood, and we have specified `adapt` to use adaptive quadrature. The results in Display 9.4 are nearly identical to those using `xtreg`. This will not always be the case since `gllamm` uses numerical integration for all models, whereas `xtreg` exploits the availability of a closed form likelihood for linear mixed models. In `gllamm` the accuracy of the estimates can be improved by increasing the number of quadrature points used for numerical integration from its default of 8 using the `nip()` option, a strategy we would always recommend.

The format of the output for the random part is somewhat different from that of `xtreg`. ‘Variance at level 1’ refers to the residual variance σ^2 , whereas `var(1)` under ‘Variances and covariances of random effects’ refers to the random intercept variance τ^2 . The standard errors given in parentheses next to these estimates are not very useful and neither are the standard errors for the standard deviations reported by `xtreg`, since the sampling distributions of the estimates are unlikely to be well approximated by a normal distribution.

The most common method of predicting random effects is by their posterior mean, their expectation given the observed responses and covariates with the parameter estimates plugged in. These predictions are also known as *empirical Bayes* predictions, shrinkage estimates or, in linear mixed models, best linear unbiased predictions (BLUP). Adding predictions of the random intercept to the predicted mean response profile gives individual predicted response profiles. These can be obtained using `gllamm`’s prediction command `gllapred` with the `linpred` option (for ‘linear predictor’):

```
gllapred pred, linpred
```

A graph of the individual predicted profiles is obtained using

```
twoway (line pred1 visit, connect(ascending)), by(group) /*
*/ ytitle(Depression) xlabel(0/6)
```

and given in [Figure 9.2](#). It is clear that the mean profiles in [Figure 9.1](#) have simply been shifted up and down to fit the observed individual profiles more closely.

We can obtain the empirical Bayes predictions of u_i using

```

Running adaptive quadrature
Iteration 1:    log likelihood = -1027.2577
Iteration 2:    log likelihood = -1024.4834
Iteration 3:    log likelihood = -1024.3838
Iteration 4:    log likelihood = -1024.3838

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:    log likelihood = -1024.3838
Iteration 1:    log likelihood = -1024.3838  (backed up)

number of level 1 units = 356
number of level 2 units = 61

Condition Number = 88.726361

gllamm model

log likelihood = -1024.3838

```

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-1.47265	1.139489	-1.29	0.196	-3.706007	.7607073
visit	-3.786777	.3710823	-10.20	0.000	-4.514085	-3.059469
gr_vis	-.5847336	.2073934	-2.82	0.005	-.9912171	-.1782501
vis2	.3851253	.0569325	6.76	0.000	.2735397	.4967109
_cons	20.90994	.8860844	23.60	0.000	19.17324	22.64663

Variance at level 1

13.532227 (1.1100236)

Variances and covariances of random effects

***level 2 (subj)

var(1): 12.853237 (2.7749441)

Display 9.4

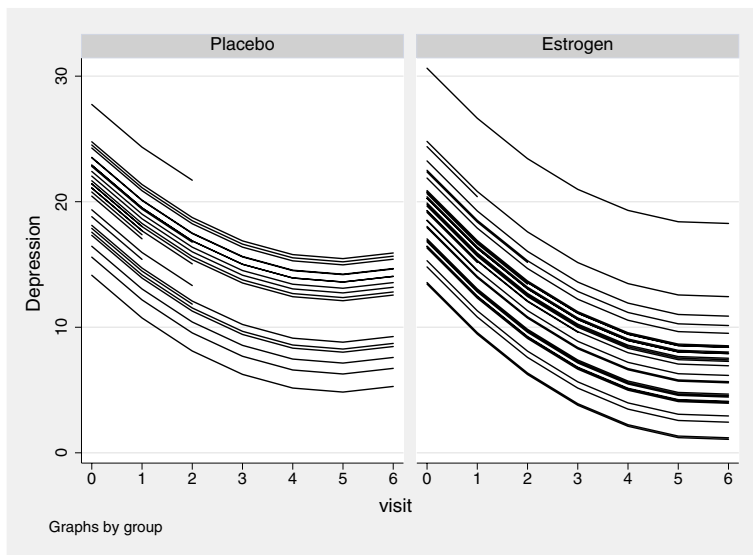


Figure 9.2: Predicted response curves for random intercept model.

```
gllapred rand, u
```

which creates new variables `randm1` for the predictions and `rands1` for the posterior standard deviations. For linear models the latter are equal to the prediction error standard deviations, also known as the ‘comparative standard errors’. We can obtain the sampling standard deviations or ‘diagnostic standard errors’ by taking the square root of the (prior) random intercept variance, estimated as 12.85, minus the posterior variance:

```
gen se = sqrt(12.853237-rands1^2)
```

(see Skrondal and Rabe-Hesketh (2004) for a detailed discussion of empirical Bayes and the different kinds of standard errors.) A graph of the predictions with their approximate 95% confidence intervals for the placebo group is then obtained using

```
gen f=visit==0
sort randm1
gen rank=sum(f)
serrbar randm1 se rank if visit==0&group==0, scale(2) /*
    */ xtitle(Rank) ytitle("Random intercept")
```

with the result shown in [Figure 9.3](#). In linear mixed models the predicted random effects should be normally distributed, so that we can

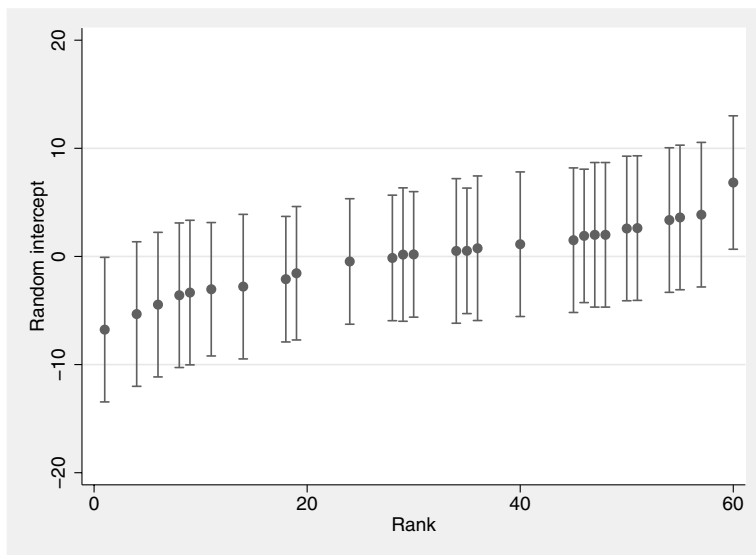


Figure 9.3: Predicted random intercepts and approximate 95% confidence intervals for the placebo group (based on the sampling standard deviations).

use graphs to assess the assumption that the ‘true’ random effects are normally distributed. One possibility is a kernel density plot with a normal density superimposed:

```
kdensity randm1 if visit==0, epanechnikov normal /*
*/ xtitle("Predicted random intercept")
```

In Figure 9.4, the normal density appears to approximate the empirical density well. Note that this diagnostic cannot be used for generalized linear mixed models where the predicted random effects are generally non-normal even under correct specification.

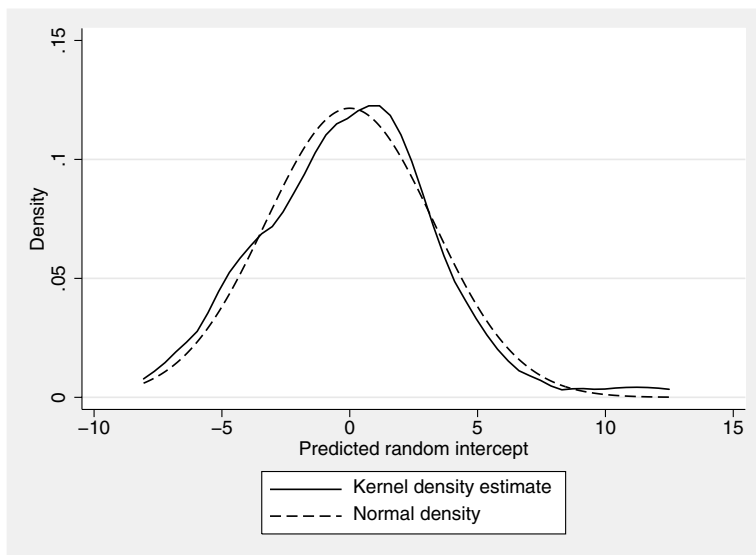


Figure 9.4: Kernel density estimate for empirical Bayes predictions and approximating normal density.

We will now allow the coefficient of `visit` to vary randomly between subjects by including a random slope in the model. When there are several random effects (here intercept and slope), we have to define an equation for each of them to specify the variable multiplying the random effect. The random intercept u_{0i} in equation (9.2) is not multiplied by anything, or equivalently it's multiplied by 1, whereas the random coefficient u_{1i} is multiplied by t_{ij} , the variable `visit`. We therefore define the equations as follows:

```

gen cons = 1
eq inter: cons
eq slope: visit

```

The syntax for the equations is simply `eq label: varlist`, where *label* is an arbitrary equation name. We can now run `gllamm` with two extra options, `nrf(2)` to specify that there are two random effects and `eqs(inter slope)` to define the variables multiplying the random effects:

```

gllamm dep group visit gr_vis vis2, i(subj) nrf(2) /*
      */ eqs(inter slope) adapt

```

In Display 9.5 we see that the output under ‘Variances and covariances of random effects’ has become more complex. We now have two variances `var(1)` and `var(2)` of the intercept and slope, respectively, and a covariance `cov(2,1)`. (The first random effect is the random intercept since `inter` was the first equation in the `eqs()` option.) We see that there is a small positive correlation between the slope and intercept. The log likelihood of this model is -1017.27 compared with -1024.38 for the random intercept model. A conventional likelihood ratio test would compare twice the difference in log likelihoods with a chi-squared distribution with two degrees of freedom (for an extra variance and covariance parameter). However, the null hypothesis that the slope has zero variance lies on the boundary of the parameter space (since a variance cannot be negative), and this test is therefore not valid. Snijders and Bosker (1999) suggest dividing the *p*-value of the conventional likelihood ratio test by two, giving a highly significant result here.

The empirical Bayes predictions for the random coefficient model can be obtained and plotted using

```

gllapred reff, u
twoway scatter reffm2 reffm1 if visit==0, /*
      */ xtitle("Intercept") ytitle("Slope")

```

giving the graph in [Figure 9.5](#). We could again assess the normality of the random effects graphically.

The predicted profiles can be plotted using

```

gllapred pred2, linpred
sort subj visit
twoway (line pred2 visit, connect(ascending)), by(group) /*
      */ ytitle(Depression) xlabel(0/6)

```

resulting in [Figure 9.6](#) where the curves are now no longer parallel due to the random slopes.

Finally, we can assess the fit of the model by plotting both observed and predicted profiles in a trellis graph containing a separate scatterplot

```

Running adaptive quadrature
Iteration 1:   log likelihood = -1041.9523
Iteration 2:   log likelihood = -1028.5971
Iteration 3:   log likelihood = -1017.3156
Iteration 4:   log likelihood = -1017.2726
Iteration 5:   log likelihood = -1017.2726

```

```

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -1017.2726
Iteration 1:   log likelihood = -1017.2726  (backed up)

```

```

number of level 1 units = 356
number of level 2 units = 61

```

```

Condition Number = 60.390774

```

```

gllamm model

```

```

log likelihood = -1017.2726

```

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-1.482598	1.023247	-1.45	0.147	-3.488125	.5229291
visit	-3.780784	.375162	-10.08	0.000	-4.516088	-3.04548
gr_vis	-.5855475	.2685742	-2.18	0.029	-1.111943	-.0591517
vis2	.3891232	.0536605	7.25	0.000	.2839505	.4942959
_cons	20.91118	.7985287	26.19	0.000	19.34609	22.47626

```

Variance at level 1

```

```

-----
11.959209 (1.0781307)

```

```

Variances and covariances of random effects

```

```

***level 2 (subj)

```

```

var(1): 9.7895918 (2.8720274)
cov(2,1): .17530873 (.52828878) cor(2,1): .09213814

var(2): .3697972 (.17467937)
-----

```

Display 9.5

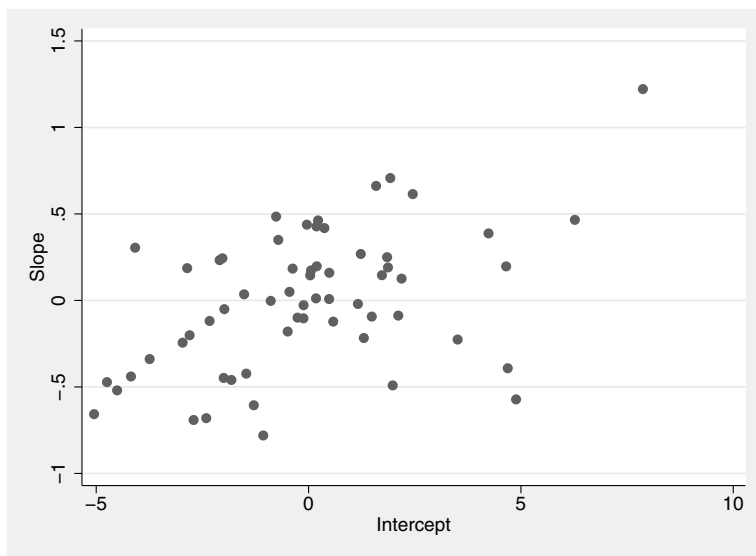


Figure 9.5: Scatterplot of predicted intercepts and slopes.

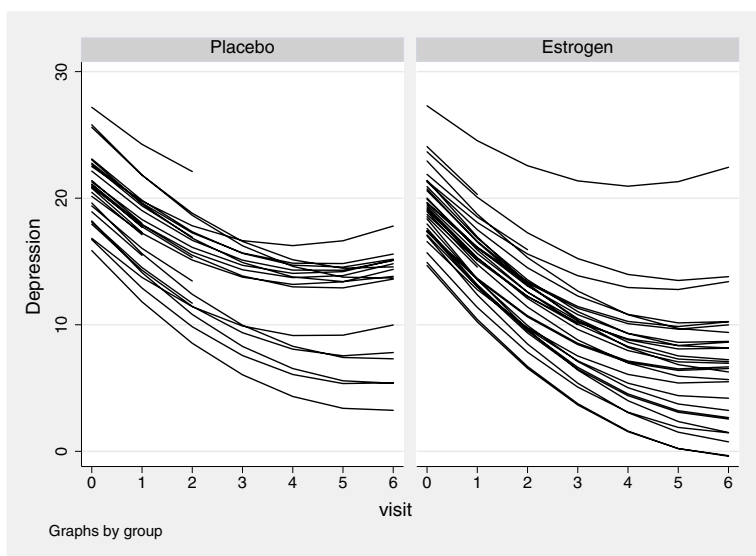


Figure 9.6: Predicted response profiles for random coefficient model.

for each subject. For the placebo group the command is

```
twoway (line pred2 visit) (connect dep visit, clpat(dash)) /*  
    */ if group==0, by(subj, style(compact)) ytitle(Depression)
```

and similarly for the treatment group. The resulting graphs are shown in [Figure 9.7](#). The model appears to represent the data reasonably well.

9.4 Thought disorder data

The thought disorder data are read in using

```
use madras
```

Next we stack the dichotomous responses `y0` to `y10` into a single variable `y`, and create a new variable `month` taking on values 0 to 10 using

```
reshape long y, i(id) j(month)
```

We wish to investigate how the risk of having thought disorder evolves over time and whether there are differences between early and late onset patients. An obvious first model to estimate is a logistic random intercept model with fixed effects of `month`, `early` and their interaction. This can be done using Stata's `xtlogit` command:

```
gen month_early = month*early  
xtlogit y month early month_early, i(id) nolog or
```

The output is shown in Display 9.6. The `or` option was used to obtain odds ratios in the first table. These suggest that there is a decrease in the odds of having thought disorder over time. However, patients with early onset schizophrenia do not differ significantly from late onset patients in their odds of thought disorder at the time of hospitalization ($OR=1.05$) nor do their odds change at a significantly different rate over time ($OR=0.94$). The log of the random intercept standard deviation is estimated as 1.02 and the standard deviation itself as 1.67. Here ρ is the intraclass correlation for the latent responses; see the latent response formulation of the ordinal logit model in Chapter 6.

The same model can be estimated in `gllamm` which offers adaptive quadrature as well as allowing posterior means and other predictions to be computed using `gllapred`:

```
gllamm y month early month_early, i(id) link(logit) /*  
    */ family(binom) adapt eform nolog  
estimates store mod1
```

Here we used syntax very similar to that of `glm` with the `link()`, `family()`, and `eform` options and stored the estimates for later using

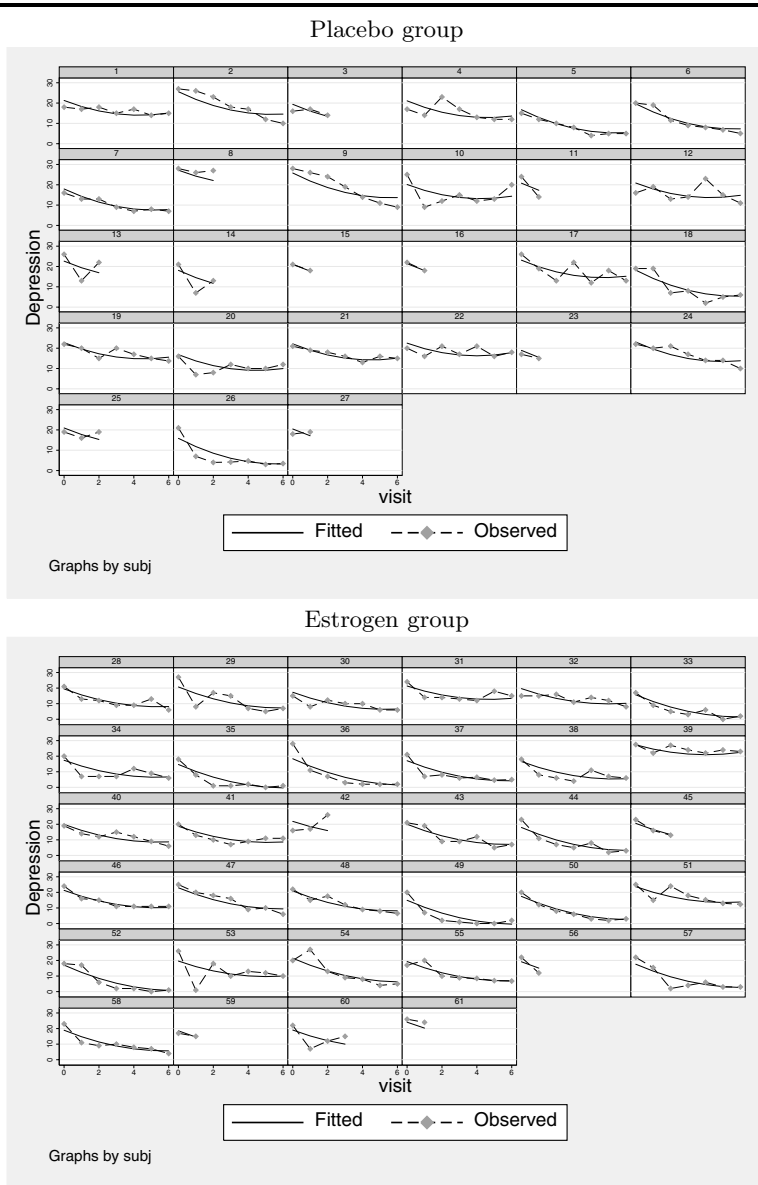


Figure 9.7: Observed and predicted response profiles for random coefficient model.

Random-effects logistic regression	Number of obs	=	244
Group variable (i): id	Number of groups	=	44
Random effects u_i ~ Gaussian	Obs per group: min	=	1
	avg	=	5.5
	max	=	6
Log likelihood = -124.73879	Wald chi2(3)	=	36.22
	Prob > chi2	=	0.0000

y	OR	Std. Err.	z	P> z	[95% Conf. Interval]	
month	.6676518	.0517672	-5.21	0.000	.5735233	.777229
early	1.045213	.9163235	0.05	0.960	.1874856	5.826951
month_early	.9351536	.1313966	-0.48	0.633	.7100391	1.23164
/lnsig2u	1.022531	.4610921			.1188072	1.926255
sigma_u	1.6674	.3844125			1.061203	2.619878
rho	.45802	.1144604			.2550153	.6759908

Likelihood-ratio test of rho=0: chibar2(01) = 26.13 Prob >= chibar2 = 0.000

Display 9.6

`estimates store`. In Display 9.7 we can see that the estimates are nearly the same as using `xtlogit`. Increasing the number of quadrature points to 12 using `nip(12)` again gives virtually the same estimates (not shown) suggesting that 8 points are sufficient.

We now include random slopes of `month` in the model using the same syntax as in the linear case to specify the random part:

```

gen cons = 1
eq slope: month
eq inter: cons
gllamm y month early month_early, i(id) nrf(2) /*
    */ eqs(inter slope) link(logit) family(binom) /*
    */ adapt nolog
estimates store mod2

```

giving the output in Display 9.8. The log likelihood has decreased by about 3.5 suggesting that the random slope is needed. The fixed effects estimates are very similar to those for the random intercept model. The estimated correlation between the random intercepts and slopes at the time of hospitalization is -0.71 . Note that both the random intercept variance and the covariance and correlation refer to the situation when `month` is zero and change if we move the origin by adding or subtracting a constant from `month`.

We now produce some graphs of the model predictions, considering first the random intercept model. For the post-natal depression data

number of level 1 units = 244
number of level 2 units = 44

Condition Number = 19.833627

gllamm model

log likelihood = -124.74702

y	exp(b)	Std. Err.	z	P> z	[95% Conf. Interval]	
month	.6677456	.0516849	-5.22	0.000	.5737548	.7771337
early	1.047086	.9197121	0.05	0.958	.1872089	5.856505
month_early	.935051	.1309953	-0.48	0.632	.7105372	1.230506

Variances and covariances of random effects

***level 2 (id)

var(1): 2.755484 (1.242513)

Display 9.7

number of level 1 units = 244
number of level 2 units = 44

Condition Number = 26.475484

gllamm model

log likelihood = -121.19981

y	exp(b)	Std. Err.	z	P> z	[95% Conf. Interval]	
month	.6030476	.0770181	-3.96	0.000	.4695053	.7745736
early	1.038847	1.252579	0.03	0.975	.0977741	11.03772
month_early	.9376264	.1940059	-0.31	0.756	.6250377	1.406544

Variances and covariances of random effects

***level 2 (id)

var(1): 7.1740353 (4.0302042)
cov(2,1): -.70917076 (.53491264) cor(2,1): -.70286756

var(2): .14190285 (.0944932)

Display 9.8

the mean profile in Figure 9.1 was simply equal to the fixed part of the random intercept model $\hat{\beta}^T \mathbf{x}_{ij}$ since the mean of the random intercept is zero. In the logistic model, things are more complicated because the probability of thought disorder given the random intercept (the *subject-specific* probability) is a nonlinear function of the random intercept:

$$\Pr(y_{ij} = 1|u_i) = \frac{\exp(\beta^T \mathbf{x}_{ij} + u_i)}{1 + \exp(\beta^T \mathbf{x}_{ij} + u_i)}. \quad (9.4)$$

The *population average* probability is therefore not equal to the above with $u_i = 0$,

$$\int \Pr(y_{ij} = 1|u_i) g(u_i) du_i \neq \frac{\exp(\beta^T \mathbf{x}_{ij})}{1 + \exp(\beta^T \mathbf{x}_{ij})}, \quad (9.5)$$

where $g(u_i)$ is the normal probability density function of u_i . For this reason the coefficients β , representing the *conditional* or *subject-specific* effects of covariates, for a given value of the random effect, cannot be interpreted as *population average* or *marginal effects* (which tend to be closer to zero). (Note that here the term ‘marginal effects’ means population average effects, a very different notion than ‘marginal effects’ in econometrics as computed by the Stata command `mf`.)

In `gllapred` we can use the `mu` and `marg` options to obtain the marginal probabilities on the left-hand side of (9.5) by numerical integration and the `mu` and `us()` options to obtain the conditional probabilities in (9.4) for given values of u_i . To obtain smooth curves, we first create a new dataset where `month` increases gradually from 0 to 10 and `early` equals 1:

```
replace month = 10*(_n-1)/(_N-1)
replace early = 1
replace month_early = month
```

Now we can obtain marginal probabilities for the random intercept model by first restoring the estimates and then using `gllapred`:

```
estimates restore mod1
gllapred probm1, mu marg
```

To calculate conditional probabilities, we must first define variables equal to the values at which we wish to evaluate u_i (0 and ± 1.7 , approximately one standard deviation). The variable names must end on ‘1’ since the random intercept is the first (and here the only) random effect:

```

gen m1 = 0
gen l1 = -1.7
gen u1 = 1.7
gllapred probc1_m, mu us(m)
gllapred probc1_l, mu us(l)
gllapred probc1_u, mu us(u)
drop m1 l1 u1

```

Here the `us(m)` option specifies that, if there were more than one random effect, the values would be in `m1`, `m2`, etc. Before producing the graph, we will make predictions for the random coefficient model:

```

estimates restore mod2
gllapred probm2, mu marg
gen m1 = 0
gen m2 = 0
gen l1 = -2.7
gen l2 = -0.4
gen u1 = 2.7
gen u2 = 0.4
gen ul1 = 2.7
gen ul2 = -0.4
gen lu1 = -2.7
gen lu2 = 0.4
gllapred probc2_m, mu us(m)
gllapred probc2_l, mu us(l)
gllapred probc2_u, mu us(u)
gllapred probc2_ul, mu us(ul)
gllapred probc2_lu, mu us(lu)

```

We have produced five conditional predictions; one with both random effects equal to 0 and 4 for all combinations of high and low values of the random intercept and slope. The graphs are obtained using

```

label variable month /*
*/ "Number of months since hospitalization"
twoway (line probm1 month, clwidth(medthick)) /*
*/ (line probc1_m month, clpat(dot) clwidth(medthick)) /*
*/ (line probc1_l month, clpat(dash)) /*
*/ (line probc1_u month, clpat(dash)), /*
*/ legend(order(1 "Marginal" 2 "Fixed part" /*
*/ 3 "Conditional")) ytitle("Predicted probability")

```

and similarly for the random coefficient model; see [Figures 9.8](#) and [9.9](#).

In [Figure 9.8](#) it is clear that the marginal or population average curve is flatter than the conditional or subject-specific curves. The dotted curve for $u_i = 0$ represents the curve of an average individual since 0 is the mean of the random effects distribution. Note that this curve of an average or typical individual differs from the population

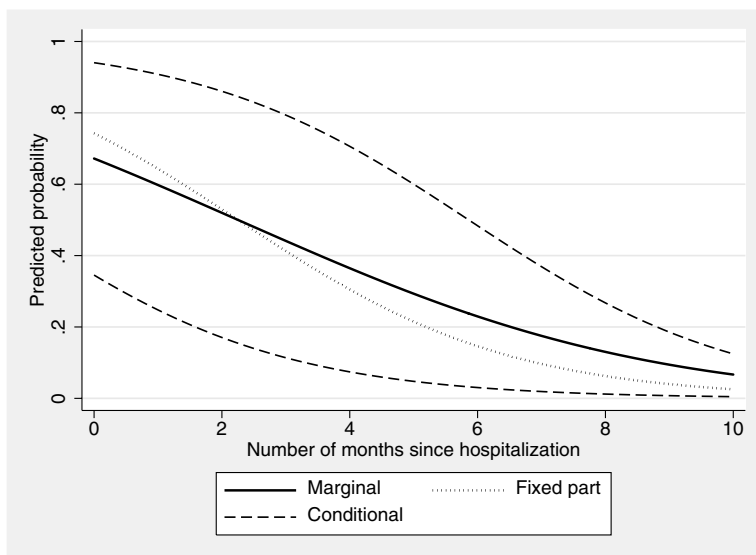


Figure 9.8: Marginal and conditional predicted probabilities for random intercept model. The dotted curve is the conditional probability when $u_i = 0$.

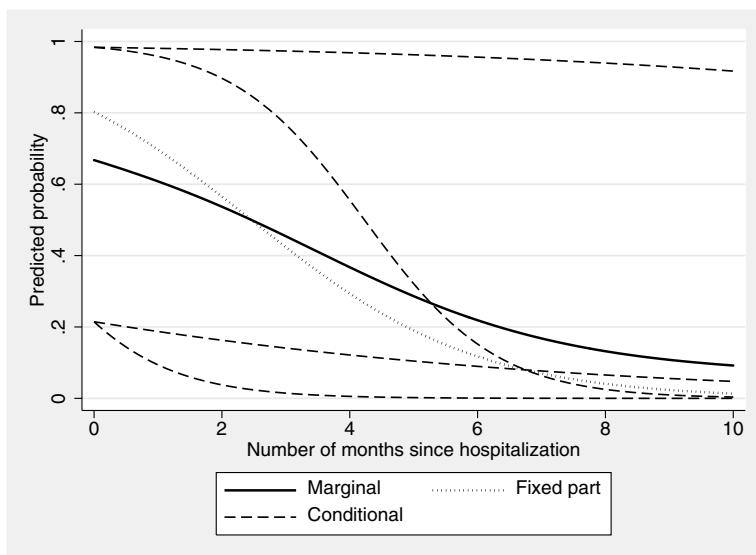


Figure 9.9: Conditional and marginal predicted probabilities for random coefficient model. The dotted curve is the conditional probability when $u_i = 0$.

average curve! In [Figure 9.9](#), we can see how different the trajectories for different patients can be in a random coefficient model. Again, the curve of the average individual differs from the average curve. Although the conditional predictions for the two models are quite different, the marginal predictions are nearly the same as can be seen by plotting the two marginal curves on the same graph:

```
twoway (line probm1 month)                                /*
      */ (line probm2 month, clpat(dash)), legend(order(1 /*
      */ "Random intercept" 2 "Random int. & slope"))      /*
      */ ytitle("Marginal probability of thought disorder")/*
      */ ylabel(0(.2)1)
```

(see [Figure 9.10](#)). Here the `ylabel()` option was used to extend the *y*-axis range to 1 and produce appropriate axis labels to make this graph comparable with [Figures 9.8](#) and [9.9](#).

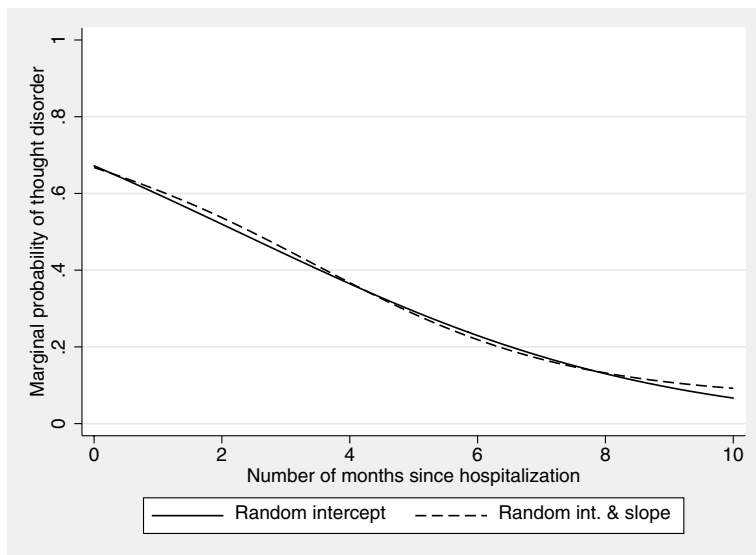


Figure 9.10: Marginal predicted probabilities for random intercept and random coefficient models.

Note that `gllamm` can be used for a wide range of models with random effects and other latent variables such as factors, including (multilevel) structural equation models and latent class models with many different response types as well as mixed responses (see Skron-

dal and Rabe-Hesketh, 2003; 2004; Rabe-Hesketh *et al.*, 2003; 2004a; 2004b). The webpage <http://www.gllamm.org> gives more references and up-to-date information.

9.5 Exercises

1. For the depression data, run the random effects model using different estimation methods available for `xtreg`. (See the *Stata Reference Manual* [XT] `xtreg` for the interpretation of the `fe` and `be` options.)
2. For the thought disorder data, produce graphs of the predicted probabilities for the individual patients separately for early and late onset (similar to [Figures 9.2](#) and [9.6](#) for the post-natal depression data). Hint: use `gllapred` with the `mu` option (not `marg` or `us()`) to obtain posterior mean probabilities.
3. Analyze the Australian school children data described in Chapter 7 using a Poisson model with a random intercept for each child and compare the estimates with those of the negative binomial model estimated in Section 7.3.4, where the exponentiated random intercept (frailty) has a gamma distribution instead of a log-normal distribution.
4. Analyze the chemotherapy data introduced in the next chapter using a random intercept model. Note that ordinary quadrature as provided by `xtpoisson` does not work here, and you should use `gllamm` with the `adapt` option (see Rabe-Hesketh *et al.*, 2002).

Chapter 10

Generalized Estimating Equations: Epileptic Seizures and Chemotherapy

10.1 Introduction

In a clinical trial reported by Thall and Vail (1990), 59 patients with epilepsy were randomized to groups receiving either the anti-epileptic drug progabide or a placebo in addition to standard chemotherapy. The number of seizures was counted over four two-week periods. In addition, a baseline seizure rate was recorded for each patient, based on the eight-week prerandomization seizure count. The age of each patient was also recorded. The main question of interest is whether the treatment progabide reduces the frequency of epileptic seizures compared with placebo. The data are shown in Table 10.1. (These data also appear in Hand *et al.*, 1994.)

Table 10.1 Data in `epil.dta`

subj	id	y1	y2	y3	y4	treat	base	age
1	104	5	3	3	3	0	11	31
2	106	3	5	3	3	0	11	30
3	107	2	4	0	5	0	6	25
4	114	4	4	1	4	0	8	36
5	116	7	18	9	21	0	66	22
6	118	5	2	8	7	0	27	29

Table 10.1 Data in `epil.dta` (continued)

7	123	6	4	0	2	0	12	31
8	126	40	20	23	12	0	52	42
9	130	5	6	6	5	0	23	37
10	135	14	13	6	0	0	10	28
11	141	26	12	6	22	0	52	36
12	145	12	6	8	4	0	33	24
13	201	4	4	6	2	0	18	23
14	202	7	9	12	14	0	42	36
15	205	16	24	10	9	0	87	26
16	206	11	0	0	5	0	50	26
17	210	0	0	3	3	0	18	28
18	213	37	29	28	29	0	111	31
19	215	3	5	2	5	0	18	32
20	217	3	0	6	7	0	20	21
21	219	3	4	3	4	0	12	29
22	220	3	4	3	4	0	9	21
23	222	2	3	3	5	0	17	32
24	226	8	12	2	8	0	28	25
25	227	18	24	76	25	0	55	30
26	230	2	1	2	1	0	9	40
27	234	3	1	4	2	0	10	19
28	238	13	15	13	12	0	47	22
29	101	11	14	9	8	1	76	18
30	102	8	7	9	4	1	38	32
31	103	0	4	3	0	1	19	20
32	108	3	6	1	3	1	10	30
33	110	2	6	7	4	1	19	18
34	111	4	3	1	3	1	24	24
35	112	22	17	19	16	1	31	30
36	113	5	4	7	4	1	14	35
37	117	2	4	0	4	1	11	27
38	121	3	7	7	7	1	67	20
39	122	4	18	2	5	1	41	22
40	124	2	1	1	0	1	7	28
41	128	0	2	4	0	1	22	23
42	129	5	4	0	3	1	13	40
43	137	11	14	25	15	1	46	33
44	139	10	5	3	8	1	36	21
45	143	19	7	6	7	1	38	35
46	147	1	1	2	3	1	7	25
47	203	6	10	8	8	1	36	26
48	204	2	1	0	0	1	11	25
49	207	102	65	72	63	1	151	22
50	208	4	3	2	4	1	22	32
51	209	8	6	5	7	1	41	25
52	211	1	3	1	5	1	32	35
53	214	18	11	28	13	1	56	21
54	218	6	3	4	0	1	24	41

Table 10.1 Data in `epil.dta` (continued)

55	221	3	5	4	3	1	16	32
56	225	1	23	19	8	1	22	26
57	228	2	3	0	1	1	25	21
58	232	0	0	0	0	1	13	36
59	236	1	4	3	2	1	12	37

10.2 Generalized estimating equations

In this chapter we consider an approach to the analysis of longitudinal data that is very different from random effects modeling described in the previous chapter. Instead of attempting to model the dependence between responses on the same individuals as arising from between-subject heterogeneity represented by random intercepts and possibly slopes, we will concentrate on estimating the marginal mean structure, treating the dependence as a nuisance.

10.2.1 *Normally distributed responses*

If we suppose that a normally distributed response is observed on each individual at T time points, then the basic regression model for longitudinal data becomes (cf. equation (3.3))

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i, \quad (10.1)$$

where $\mathbf{y}'_i = (y_{i1}, y_{i2}, \dots, y_{iT})$, $\boldsymbol{\epsilon}' = (\epsilon_{i1}, \epsilon_{i2}, \dots, \epsilon_{iT})$, \mathbf{X}_i is a $T \times (p+1)$ design matrix, and $\boldsymbol{\beta}' = (\beta_0, \dots, \beta_p)$ is a vector of regression parameters. The residual terms are assumed to have a multivariate normal distribution with a covariance matrix of some particular form that is a function of (hopefully) a small number of parameters. Maximum likelihood estimation can be used to estimate both the parameters in (10.1) and the parameters structuring the covariance matrix (details are given in Jennrich and Schluchter, 1986). The latter are often not of primary interest (they are often referred to as nuisance parameters), but using a covariance matrix that fails to match that of the repeated measurements can lead to invalid inferences about the parameters that *are* of concern, namely the $\boldsymbol{\beta}$ in (10.1).

If each non-replicated element of the covariance matrix is treated as a separate parameter, giving an unstructured covariance matrix, and if there are no missing data, then this approach is essentially equivalent to multivariate analysis of variance for longitudinal data (see Everitt, 2001). However, it is often more efficient to impose some meaningful structure on the covariance matrix. The simplest (and most unrealis-

tic) structure is *independence* with all off-diagonal elements (the covariances) equal to zero, and typically all diagonal elements (the variances) equal to each other. Another commonly used simple structure, known as *compound symmetry* (for example, see Winer, 1971), requires that all covariances are equal and all variances are equal. This is just the correlation structure of a linear random intercept model described in the previous chapter except that the random intercept model also requires that the correlation be positive.

Other correlation structures include autoregressive structures where the correlations decrease with the distance between time points. Whatever the assumed correlation structure, all models may be estimated by maximum likelihood.

10.2.2 *Non-normal responses*

Unfortunately, it is generally not straightforward to specify a multivariate model for non-normal responses. One solution, discussed in the previous chapter, is to induce residual dependence among the responses using random effects. An alternative approach is to give up the idea of a model altogether by using *generalized estimating equations* (GEE) as introduced by Liang and Zeger (1986). Generalized estimating equations are essentially a multivariate extension of the quasi-likelihood approach discussed in [Chapter 7](#) (see also Wedderburn, 1974). In GEE the parameters are estimated using ‘estimating equations’ resembling the score equations for maximum likelihood estimation of the linear model described in the previous section. These estimating equations only require specification of a link and variance function and a correlation structure for the observed responses conditional on the covariates. As in the quasi-likelihood approach, the parameters can be estimated even if the specification does not correspond to any statistical model.

The regression coefficients represent marginal effects, i.e., they determine the population average relationships. Liang and Zeger (1986) show that the estimates of these coefficients are valid even when the correlation structure is incorrectly specified. Correct inferences can be obtained using robust estimates of the standard errors based on the sandwich estimator for clustered data (e.g. Binder, 1983; Williams, 2000). The parameters of the correlation matrix, referred to as the *working correlation matrix*, are treated as *nuisance parameters*. However, Lindsey and Lambert (1998) and Crouchley and Davies (1999) point out that estimates are no longer consistent if ‘endogenous’ covariates such as baseline responses are included in the model. Fortunately, inclusion of the baseline response as a covariate does yield consistent estimates of treatment effects in clinical trial data such as the chemother-

any data considered here (see Crouchley and Davies, 1999) as long as the model does not contain a baseline by treatment interaction.

There are some important differences between GEE and random effects modeling. First, while random effects modeling is based on a statistical model and typically maximum likelihood estimation, GEE is just an estimation method that is not based on a statistical model. Second, there is an important difference in the interpretation of the regression coefficients. In random effects models, the regression coefficients represent *conditional* or *subject-specific* effects for given values of the random effects. For GEE, on the other hand, the regression coefficients represent *marginal* or *population average* effects. As we saw in the thought disorder data in the previous chapter, conditional and marginal relationships can be very different. Either may be of interest; for instance patients are likely to want to know the subject-specific effect of treatments, whereas health economists may be interested in population average effects. Whereas random effects models allow the marginal relationship to be derived, GEE does not allow derivation of the conditional relationship. Note that conditional and marginal relationships are the same if an identity link is used and, in the case of random intercept models (no random coefficients), if a log link is specified (see Diggle *et al.*, 2002). Third, GEE is often preferred because, in contrast to the random effects approach, the parameter estimates are consistent even if the correlation structure is misspecified, although this is only true if the mean structure is correctly specified. Fourth, while maximum likelihood estimation of a correctly specified model is consistent if data are missing at random (MAR), this is not the case for GEE which requires that responses are missing completely at random (MCAR). See Hardin and Hilbe (2002) for a thorough introduction to GEE.

10.3 Analysis using Stata

The generalized estimating equations approach, as described in Liang and Zeger (1986), is implemented in Stata's `xtgee` command. The main components which have to be specified are:

- the assumed distribution of the response variable, specified in the `family()` option – this determines the variance function,
- the link between the response variable and its linear predictor, specified in the `link()` option, and
- the structure of the working correlation matrix, specified in the `correlations()` option.

In general, it is not necessary to specify both `family()` and `link()` since, as explained in [Chapter 7](#), the default link is the canonical link for the specified family.

Since the `xtgee` command will often be used with the `family(gauss)` option, together with the identity link function, we will illustrate this option on the post-natal depression data used in the previous two chapters before moving on to deal with the data in [Table 10.1](#).

10.3.1 *Post-natal depression data*

The data are obtained using

```
infile subj group dep0 dep1 dep2 dep3 dep4 dep5 dep6 /*
      */ using depress.dat
reshape long dep, i(subj) j(visit)
mvdecode _all, mv(-9)
```

To begin, we fit a model that regresses depression on `group`, `visit`, their interaction and `visit` squared as in the previous chapter but under the unrealistic assumptions of independence. The necessary instruction written out in its fullest form is

```
gen gr_vis = group*visit
gen vis2 = visit^2
xtgee dep group visit gr_vis vis2, i(subj) t(visit) /*
      */ corr(indep) link(iden) family(gauss)
```

(see Display 10.1). Here, the fitted model is simply a multiple regression model for 295 observations which are assumed to be independent of one another; the scale parameter is equal to the residual mean square, and the deviance is equal to the residual sum of squares. The estimated regression coefficients and their associated standard errors indicate that the `group` by `visit` interaction is significant at the 5% level. However, treating the observations as independent is unrealistic and will almost certainly lead to poor estimates of the standard errors. Standard errors for between-subject factors (here `group`) are likely to be underestimated because we are treating observations from the same subject as independent, thus increasing the apparent sample size; standard errors for within-subject factors (here `visit`, `gr_vis`, and `vis2`) are likely to be overestimated since we are not controlling for residual between-subject variability.

We therefore now abandon the assumption of independence and estimate a correlation matrix having compound symmetry (i.e., constraining the correlations between the observations at any pair of time points to be equal). Such a correlational structure is specified using `corr(exchangeable)`, or the abbreviated form `corr(exc)`. The model can be fitted as follows:

Iteration 1: tolerance = 3.543e-14

GEE population-averaged model		Number of obs	=	356
Group variable:	subj	Number of groups	=	61
Link:	identity	Obs per group: min	=	2
Family:	Gaussian	avg	=	5.8
Correlation:	independent	max	=	7
		Wald chi2(4)	=	269.51
Scale parameter:	26.89935	Prob > chi2	=	0.0000
Pearson chi2(356):	9576.17	Deviance	=	9576.17
Dispersion (Pearson):	26.89935	Dispersion	=	26.89935

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-1.506834	.9383647	-1.61	0.108	-3.345995	.3323274
visit	-3.849465	.5091836	-7.56	0.000	-4.847447	-2.851483
gr_vis	-.6090744	.277417	-2.20	0.028	-1.152802	-.0653471
vis2	.3904383	.079783	4.89	0.000	.2340665	.5468102
_cons	20.96533	.7826299	26.79	0.000	19.4314	22.49925

Display 10.1

```
xtgee dep group visit gr_vis vis2, i(subj) t(visit) /*
      */ corr(exc) link(iden) fam(gauss)
```

Instead of specifying the subject and time identifiers using the options `i()` and `t()`, we can also declare the data as being of the form `xt` (for cross-sectional time series) as follows:

```
iis subj
tis visit
```

and omit the `i()` and `t()` options from now on. Since both the link and the family correspond to the default options, the same analysis may be carried out using the shorter command

```
xtgee dep group visit gr_vis vis2, corr(exc)
```

(see Display 10.2). After estimation, `xtcorr` reports the estimated working correlation matrix

```
xtcorr
```

which is shown in Display 10.3.

Note that the standard error for `group` has increased whereas those for `visit`, `gr_vis`, and `vis2` have decreased as expected. The estimated within-subject correlation matrix is compound symmetric. This structure is frequently not acceptable since correlations between pairs of observations widely separated in time will often be lower than for

```

Iteration 1: tolerance = .01567568
Iteration 2: tolerance = .00003608
Iteration 3: tolerance = 6.529e-08

```

```

GEE population-averaged model
Group variable:          subj      Number of obs      =      356
Link:                    identity   Number of groups   =      61
Family:                  Gaussian   Obs per group: min =      2
Correlation:             exchangeable      avg =      5.8
                                          max =      7
                                          Wald chi2(4)      =     421.11
Scale parameter:         26.92726      Prob > chi2        =     0.0000

```

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-1.470155	1.162063	-1.27	0.206	-3.747756	.8074468
visit	-3.785601	.3648345	-10.38	0.000	-4.500664	-3.070539
gr_vis	-.5837938	.2040368	-2.86	0.004	-.9836985	-.183889
vis2	.3850221	.0559386	6.88	0.000	.2753845	.4946598
_cons	20.90907	.901082	23.20	0.000	19.14298	22.67516

Display 10.2

Estimated within-subj correlation matrix R:

	c1	c2	c3	c4	c5	c6	c7
r1	1.0000						
r2	0.5150	1.0000					
r3	0.5150	0.5150	1.0000				
r4	0.5150	0.5150	0.5150	1.0000			
r5	0.5150	0.5150	0.5150	0.5150	1.0000		
r6	0.5150	0.5150	0.5150	0.5150	0.5150	1.0000	
r7	0.5150	0.5150	0.5150	0.5150	0.5150	0.5150	1.0000

Display 10.3

observations closer together. This pattern was apparent from the scatterplot matrix given in [Chapter 8](#).

To allow for a more complex pattern of correlations among the repeated observations, we can move to an *autoregressive structure*. For example, in a first-order autoregressive specification the correlation between time points r and s is assumed to be $\rho^{|r-s|}$. The necessary instruction for fitting the previously considered model but with this first-order autoregressive structure for the correlations is

```
xtgee dep group visit gr_vis vis2, corr(ar1)
```

```
Iteration 1: tolerance = .3759114
Iteration 2: tolerance = .01586983
Iteration 3: tolerance = .00037881
Iteration 4: tolerance = 8.875e-06
Iteration 5: tolerance = 2.078e-07
```

```
GEE population-averaged model
Group and time vars:      subj visit      Number of obs      =      356
Link:                    identity      Number of groups    =      61
Family:                  Gaussian      Obs per group: min  =       2
Correlation:             AR(1)         avg =      5.8
                                max =       7
                                Wald chi2(4)    =     213.85
Scale parameter:         27.10248      Prob > chi2         =     0.0000
```

dep	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
group	-.539061	1.277002	-0.42	0.673	-3.041938	1.963816
visit	-4.061961	.4741241	-8.57	0.000	-4.991227	-3.132695
gr_vis	-.7815801	.3332716	-2.35	0.019	-1.43478	-.1283796
vis2	.4207375	.0693395	6.07	0.000	.2848346	.5566404
_cons	21.10085	.9732406	21.68	0.000	19.19334	23.00837

Display 10.4

The estimates of the regression coefficients and their standard errors in Display 10.4 have changed but not substantially. The estimated within-subject correlation matrix may again be obtained using

```
xtcorr
```

(see Display 10.5) which has the expected pattern in which correlations decrease substantially as the separation between the observations increases.

Other correlation structures are available for `xtgee`, including the option `correlation(unstructured)` in which no constraints are placed on the correlations. (This is essentially equivalent to multivariate anal-

Estimated within-subj correlation matrix R:

	c1	c2	c3	c4	c5	c6	c7
r1	1.0000						
r2	0.6475	1.0000					
r3	0.4192	0.6475	1.0000				
r4	0.2714	0.4192	0.6475	1.0000			
r5	0.1757	0.2714	0.4192	0.6475	1.0000		
r6	0.1138	0.1757	0.2714	0.4192	0.6475	1.0000	
r7	0.0737	0.1138	0.1757	0.2714	0.4192	0.6475	1.0000

Display 10.5

ysis of variance for longitudinal data, except that the variance is assumed to be constant over time.) It might appear that using this option would be the most sensible one to choose for *all* data sets. This is not, however, the case since it necessitates the estimation of many nuisance parameters. This can, in some circumstances, cause problems in the estimation of those parameters of most interest, particularly when the sample size is small and the number of time points is large.

10.3.2 Chemotherapy data

We now analyze the chemotherapy data using a similar model as for the depression data, but using the Poisson distribution and log link. The data are available in a Stata file `epil.dta` and can be read using

```
use epil, clear
```

We will treat the baseline measure as one of the responses:

```
gen y0 = baseline
```

Some useful summary statistics can be obtained using

```
summarize y0 y1 y2 y3 y4 if treat==0
```

```
summarize y0 y1 y2 y3 y4 if treat==1
```

(see Displays 10.6 and 10.7).

The largest value of `y1` in the progabide group seems out of step with the other maximum values and may indicate an outlier. Some graphics of the data may be useful for investigating this possibility further, but first it is convenient to reshape the data from its present ‘wide’ form to the ‘long’ form. We now reshape the data as follows:

```
reshape long y, i(subj) j(visit)
sort subj treat visit
list in 1/12, clean
```

Variable	Obs	Mean	Std. Dev.	Min	Max
y0	28	30.78571	26.10429	6	111
y1	28	9.357143	10.13689	0	40
y2	28	8.285714	8.164318	0	29
y3	28	8.785714	14.67262	0	76
y4	28	7.964286	7.627835	0	29

Display 10.6

Variable	Obs	Mean	Std. Dev.	Min	Max
y0	31	31.6129	27.98175	7	151
y1	31	8.580645	18.24057	0	102
y2	31	8.419355	11.85966	0	65
y3	31	8.129032	13.89422	0	72
y4	31	6.709677	11.26408	0	63

Display 10.7

	subj	visit	y	treat	baseline	age
1.	1	0	11	0	11	31
2.	1	1	5	0	11	31
3.	1	2	3	0	11	31
4.	1	3	3	0	11	31
5.	1	4	3	0	11	31
6.	2	0	11	0	11	30
7.	2	1	3	0	11	30
8.	2	2	5	0	11	30
9.	2	3	3	0	11	30
10.	2	4	3	0	11	30
11.	3	0	6	0	6	25
12.	3	1	2	0	6	25

Display 10.8

(see Display 10.8).

Perhaps the most useful graphical display for investigating the data is a set of graphs of individual response profiles. Since we are planning to fit a Poisson model with the log link to the data, we take the log transformation before plotting the response profiles. (We need to add a positive number, say 1, because some seizure counts are zero.)

```
gen ly = log(y+1)
```

However, the baseline measure represents seizure counts over an 8-week period, compared with 2-week periods for each of the other time points. We therefore divide the baseline count by 4:

```
replace ly = log(y/4+1) if visit==0
```

and then plot the log-counts:

```
twoway connect ly visit if treat==0, by(subj, /*  
    */ style(compact)) ytitle("Log count")  
twoway connect ly visit if treat==1, by(subj, /*  
    */ style(compact)) ytitle("Log count")
```

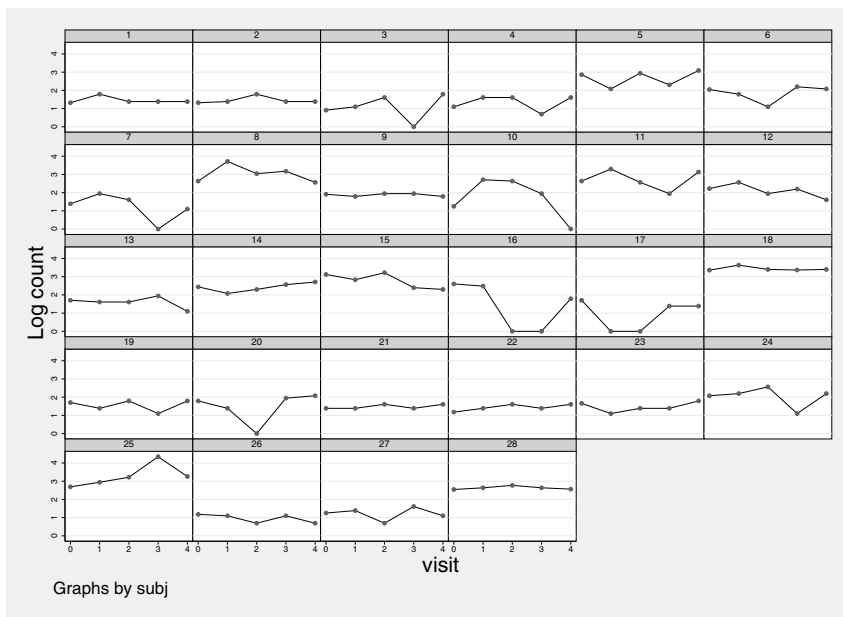


Figure 10.1: Response profiles in placebo group.

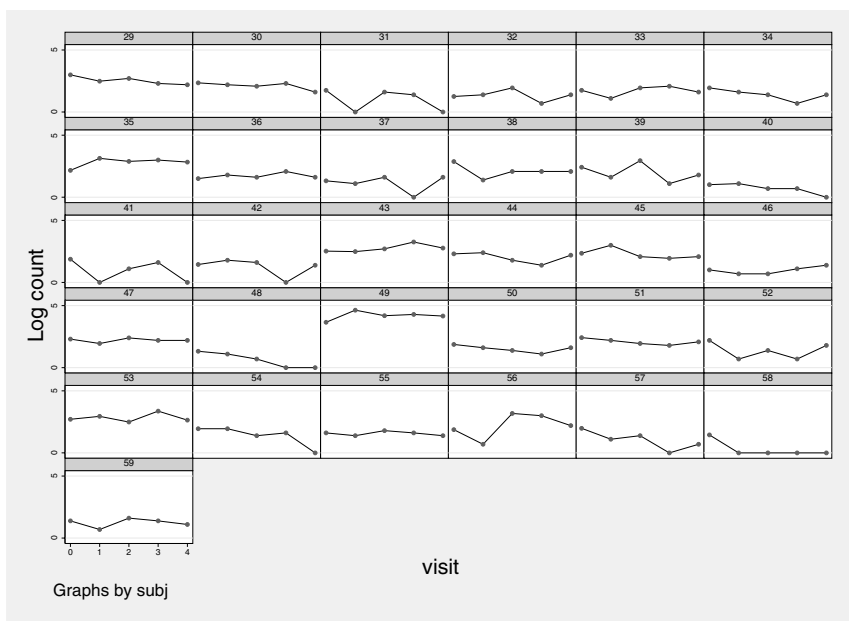


Figure 10.2: Response profiles in the treated group.

The resulting graphs are shown in [Figures 10.1](#) and 10.2. There is no obvious improvement in the progabide group. Subject 49 had more epileptic fits overall than any other subject and could well be an outlier (see Exercise 3).

As discussed in [Chapter 7](#), the most plausible distribution for count data is the Poisson distribution. The Poisson distribution is specified in `xtgee` models using the option `family(poisson)`. The log link is implied (since it is the canonical link). The baseline counts were obtained over an 8-week period whereas all subsequent counts are over 2-week periods. To model the seizure *rate* in counts per week, we must therefore use the log observation period $\log(p_i)$ as an offset (a covariate with regression coefficient set to 1). The model for the mean count μ_{ij} then becomes

$$\log(\mu_{ij}) = \beta^T \mathbf{x}_{ij} + \log(p_i),$$

so that the rate is modeled as

$$\mu_{ij}/p_i = \exp(\beta^T \mathbf{x}_{ij}).$$

We can compute the required offset using

```
gen lnoobs = cond(visit==0,ln(8),ln(2))
```

Following Diggle *et al.* (2002), we will allow the log rate to change by a treatment group-specific constant after the baseline assessment. The necessary covariates, an indicator for the post-baseline visits and an interaction between that indicator and treatment group, are created using

```
gen post = visit>0
gen tr_post = treat*post
```

We will also control for the age of the patients. The summary tables for the seizure data given on page 188 provide strong empirical evidence that there is overdispersion (the variances are greater than the means), and this can be incorporated using the `scale(x2)` option to allow the dispersion parameter ϕ to be estimated (see also [Chapter 7](#)).

```
iis subj
xtgee y age treat post tr_post, corr(exc) family(pois) /*
    */ offset(lnoobs) scale(x2)
```

```
Iteration 1: tolerance = .12201855
Iteration 2: tolerance = .00074577
Iteration 3: tolerance = 2.826e-06
Iteration 4: tolerance = 8.790e-11
```

GEE population-averaged model

Group variable:	subj	Number of obs	=	295
Link:	log	Number of groups	=	59
Family:	Poisson	Obs per group: min	=	5
Correlation:	exchangeable	avg	=	5.0
		max	=	5
		Wald chi2(4)	=	5.43
Scale parameter:	18.48008	Prob > chi2	=	0.2458

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	-.0322513	.0148644	-2.17	0.030	-.061385	-.0031176
treat	-.0177873	.201945	-0.09	0.930	-.4135922	.3780176
post	.1107981	.1500635	0.74	0.460	-.183321	.4049173
tr_post	-.1036807	.213317	-0.49	0.627	-.5217742	.3144129
_cons	2.265255	.4400816	5.15	0.000	1.402711	3.1278
lnoobs	(offset)					

(Standard errors scaled using square root of Pearson X2-based dispersion)

Display 10.9

The output assuming an exchangeable correlation structure is given in Display 10.9, and the estimated correlation matrix is obtained using `xtcorr`.

xtcorr

(see Display 10.10).

Estimated within-subj correlation matrix R:

	c1	c2	c3	c4	c5
r1	1.0000				
r2	0.7686	1.0000			
r3	0.7686	0.7686	1.0000		
r4	0.7686	0.7686	0.7686	1.0000	
r5	0.7686	0.7686	0.7686	0.7686	1.0000

Display 10.10

In Display 10.9, the parameter ϕ is estimated as 18.5, indicating severe overdispersion in these data. We briefly illustrate how important it was to allow for overdispersion by omitting the `scale(x2)` option:

```
xtgee y age treat post tr_post, corr(exc) family(pois) /*
      */ offset(lnobs)
```

Iteration 1: tolerance = .12201855
Iteration 2: tolerance = .00074577
Iteration 3: tolerance = 2.826e-06
Iteration 4: tolerance = 8.790e-11

GEE population-averaged model		Number of obs	=	295
Group variable:	subj	Number of groups	=	59
Link:	log	Obs per group: min	=	5
Family:	Poisson	avg	=	5.0
Correlation:	exchangeable	max	=	5
		Wald chi2(4)	=	100.38
Scale parameter:	1	Prob > chi2	=	0.0000

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	-.0322513	.0034578	-9.33	0.000	-.0390284	-.0254742
treat	-.0177873	.0469765	-0.38	0.705	-.1098596	.074285
post	.1107981	.0349079	3.17	0.002	.04238	.1792163
tr_post	-.1036807	.0496219	-2.09	0.037	-.2009378	-.0064235
_cons	2.265255	.102372	22.13	0.000	2.06461	2.465901
lnobs	(offset)					

Display 10.11

The results given in Display 10.11 show that the standard errors are now much smaller than before. Even if overdispersion had not been suspected, this error could have been detected by using the `robust` option (see [Chapter 7](#)):

```
xtgee y age treat post tr_post, corr(exc) family(pois) /*
    */ offset(lnobs) robust
```

```
Iteration 1: tolerance = .12201855
Iteration 2: tolerance = .00074577
Iteration 3: tolerance = 2.826e-06
Iteration 4: tolerance = 8.790e-11

GEE population-averaged model
Group variable:          subj      Number of obs      =      295
Link:                   log      Number of groups   =      59
Family:                 Poisson  Obs per group: min =      5
Correlation:            exchangeable      avg =      5.0
                                      max =      5
                                      Wald chi2(4)   =      6.85
Scale parameter:        1      Prob > chi2      =      0.1442
                                   (standard errors adjusted for clustering on subj)
```

y	Coef.	Semi-robust Std. Err.	z	P> z	[95% Conf. Interval]	
age	-.0322513	.0148746	-2.17	0.030	-.0614049	-.0030976
treat	-.0177873	.216051	-0.08	0.934	-.4412395	.4056649
post	.1107981	.1170963	0.95	0.344	-.1187064	.3403027
tr_post	-.1036807	.2154436	-0.48	0.630	-.5259424	.3185811
_cons	2.265255	.4371124	5.18	0.000	1.408531	3.12198
lnobs	(offset)					

Display 10.12

The results of the robust regression in Display 10.12 are remarkably similar to those of the overdispersed Poisson model, suggesting that the latter is a reasonable ‘model’ for the data.

The estimated coefficient of `tr_post` describes the difference in the change in log seizure rate from baseline to post randomization between the placebo and progabide groups. The small negative value indicates that the treatment is a little more effective than the placebo in reducing the seizure rate, although this is not at all significant. The exponentiated coefficient gives an incidence rate ratio, here the ratio of the relative reduction in seizure rate for the treated patients compared with the control patients. The exponentiated coefficient and the corresponding confidence interval can be obtained directly using the `eform`

option in `xtgee`:

```
xtgee y age treat post tr_post, corr(exc)      /*
      */ family(pois) offset(lnobs) scale(x2) eform
```

```
Iteration 1: tolerance = .12201855
Iteration 2: tolerance = .00074577
Iteration 3: tolerance = 2.826e-06
Iteration 4: tolerance = 8.790e-11
```

```
GEE population-averaged model
Group variable:          subj      Number of obs      =      295
Link:                  log      Number of groups     =      59
Family:                Poisson   Obs per group: min =      5
Correlation:           exchangeable   avg =      5.0
                                      max =      5
Scale parameter:        18.48008   Wald chi2(4)      =      5.43
                                      Prob > chi2       =      0.2458
```

y	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
age	.9682632	.0143927	-2.17	0.030	.9404611	.9968873
treat	.98237	.1983847	-0.09	0.930	.6612706	1.459389
post	1.117169	.1676464	0.74	0.460	.8325009	1.499179
tr_post	.9015131	.192308	-0.49	0.627	.5934667	1.369455
lnobs	(offset)					

(Standard errors scaled using square root of Pearson X2-based dispersion)

Display 10.13

The results in Display 10.13 indicate that the relative reduction in seizure rate is 10% greater in the treated group compared with the control group, with a 95% confidence interval from a 41% reduction to a 37% increase.

However, before interpreting these estimates, we should perform some diagnostics. Standardized Pearson residuals can be useful for identifying potential outliers (see equation (7.9)). These can be found by first using the `predict` command to obtain predicted counts, subtracting the observed counts, and dividing by the estimated standard deviation $\sqrt{\hat{\phi}\mu_{ij}}$, where $\hat{\phi}$ is the estimated dispersion parameter:

```
quietly xtgee y treat baseline age visit, corr(exc) /*
      */ family(pois) scale(x2)
predict pred, mu
gen pres = (y-pred)/sqrt(e(chi2_dis)*pred)
```

Boxplots of these residuals at each visit are obtained using

```

sort visit
graph box stpres, medtype(line) over(visit,      /*
  */ relabel(1 "visit 1" 2 "visit 2" 3 "visit 3" /*
  */ 4 "visit 4"))

```

The resulting graph is shown in Figure 10.3. Pearson residuals greater

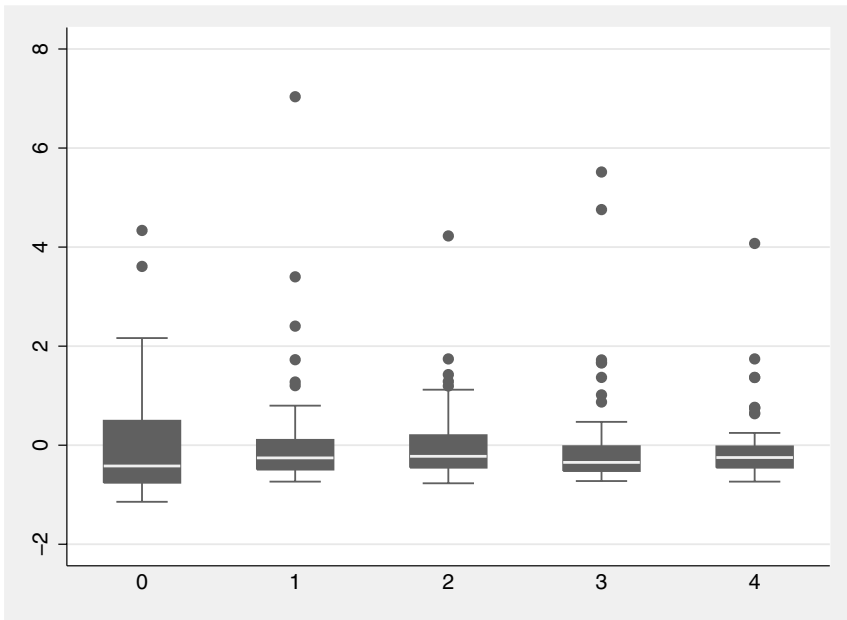


Figure 10.3: Standardized Pearson residuals.

than 4 are certainly a cause for concern, so we can check which subjects they belong to using

```
list subj id if stpres>4
```

	subj	id
41.	49	207
96.	49	207
176.	49	207
178.	49	207
185.	25	227
292.	49	207

Subject 49 appears to be an outlier due to extremely large counts as we saw in [Figure 10.2](#). Subject 25 also has an unusually large count at visit 3. It would be a good idea to repeat the analysis without subject 49 to see how much the results are affected by this unusual subject (see Exercise 3). This can be viewed as a *sensitivity analysis*.

10.4 Exercises

1. For the depression data, compare the result of GEE with a compound symmetric structure with ordinary linear regression where standard errors are corrected for the within-subject correlation using:
 - a. the options, `robust cluster(subj)`, to obtain the sandwich estimator for clustered data (see help for `regress`), and
 - b. bootstrapping, by sampling subjects with replacement, not observations. This may be achieved using the `bs` command, together with the option `cluster(subj)`.
2. Explore other possible correlational structures for the seizure data in the context of a Poisson model. Examine the robust standard errors in each case.
3. Investigate what Poisson models are most suitable when subject 49 is excluded from the analysis.
4. For the thought disorder data discussed in the previous chapter, estimate the effect of `early`, `month` and their interaction on the logit of thought disorder assuming an exchangeable correlation structure. Plot the predicted probability for early onset women (using `graph twoway function`, see Section 6.3.2), and compare the curve with the curves in [Figure 9.10](#).

Chapter 11

Some Epidemiology

11.1 Description of data

This chapter illustrates analysis of different epidemiological designs, namely cohort studies and matched as well as unmatched case-control studies. Four datasets will be used which are presented in the form of cross-tabulations in [Tables 11.1 to 11.4](#). (Tables 11.1 and 11.4 are taken from Clayton and Hills (1993) with permission of their publisher, Oxford University Press.)

The data in Table 11.1 result from a *cohort study* which investigated the relationship between diet and ischemic heart disease (IHD). Here we consider low energy intake as a risk factor since it is highly associated with lack of physical exercise. The table gives frequencies of IHD by ten-year age-band and exposure to a high or low calorie diet. The total person-years of observation are also given for each cell.

The dataset in [Table 11.2](#) is the result of a *case-control study* investigating whether keeping a pet bird is a risk factor for lung cancer. This dataset is given in Hand *et al.* (1994).

The datasets in [Tables 11.3](#) and 11.4 are from *matched* case-control studies, the first with a single matched control and the second with three matched controls. Table 11.3 arises from a matched case-control study of endometrial cancer where cases were matched on age, race, date of admission, and hospital of admission to a suitable control not suffering from cancer. Past exposure to conjugated estrogens was determined. The dataset is described in Everitt (1994). Finally, the data in Table 11.4, described in Clayton and Hills (1993), arise from a case-control study of breast cancer screening. Women who had died of breast cancer were matched with three control women. The screening

history of each control was assessed over the period up to the time of diagnosis of the matched case.

11.2 Introduction to epidemiology

Epidemiology can be described as the study of diseases in populations, in particular the search for causes of disease. For ethical reasons, subjects cannot be randomized to possible risk factors in order to establish whether these are associated with an increase in the incidence of disease, and therefore epidemiology is based on observational studies. The most important types of studies in epidemiology are cohort studies and case-control studies. We will give a very brief description of the design and analysis of these two types of studies, following closely the explanations and notation given in the excellent book, *Statistical Models in Epidemiology* by Clayton and Hills (1993).

11.2.1 Cohort studies

In a cohort study, a group of subjects free of the disease is followed up, and the presence of risk factors as well as the occurrence of the disease of interest are recorded. This design is illustrated in Figure 11.1. An

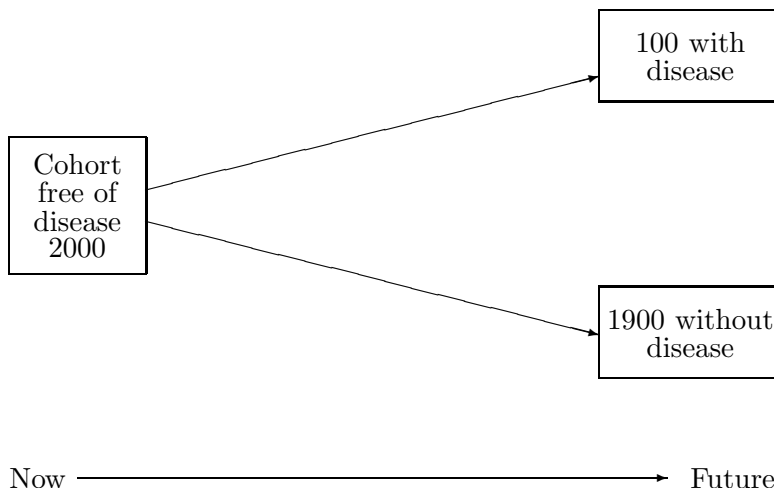


Figure 11.1: Cohort study.

Table 11.1 Number of IHD cases and person-years of observation by age and exposure to low energy diet

			Exposed < 2750 kcal		Unexposed ≥ 2750 kcal	
Age	Cases	Pers-yrs	Cases	Pers-yrs		
40-49	2	311.9	4	607.9		
50-59	12	878.1	5	1272.1		
60-69	14	667.5	8	888.9		

Table 11.2 Number of lung cancer cases and controls who keep a pet bird

Kept pet birds	Cases	Controls
Yes	98	101
No	141	328
Total	239	429

Table 11.3 Frequency of exposure to oral conjugated estrogens among cases of endometrial cancer and their matched controls

		Controls		
Cases		+	-	Total
	+	12	43	55
	-	7	121	128
Total		19	164	183

Table 11.4 Screening history in subjects who died of breast cancer and 3 matched controls

		Number of controls screened			
Status of the case		0	1	2	3
Screened		1	4	3	1
Unscreened		11	10	12	4

example of a cohort study is the study described in the previous section where subjects were followed up to monitor the occurrence of ischemic heart disease in two risk groups, those with high and low energy intake, giving the results in [Table 11.1](#).

The incidence rate of the disease λ may be estimated by the number of new cases of the disease D during a time interval divided by the person-time of observation Y , the sum of all subjects' periods of observation during the time interval:

$$\hat{\lambda} = \frac{D}{Y}.$$

This is the maximum likelihood estimator of λ assuming that D follows a Poisson distribution (independent events occurring at a constant probability rate in continuous time) with mean λY .

The most important quantity of interest in a cohort study is the *incidence rate ratio* (or relative risk), the ratio λ_1/λ_0 of incidence rates for those exposed to a risk factor and those not exposed to the risk factor (subscripts 1 and 0 denote exposed and unexposed, respectively). The incidence rate ratio may be estimated by

$$\hat{\theta} = \frac{D_1/Y_1}{D_0/Y_0}.$$

This estimator can be derived by maximizing the conditional (binomial) likelihood that there were D_1 cases in the exposed group conditional on there being a total of $D = D_0 + D_1$ cases.

However, a potential problem in estimating this rate ratio is confounding arising from systematic differences in prognostic factors between the exposure groups. This problem can be dealt with by dividing the cohort into groups or strata according to prognostic factors and assuming that the rate ratio for exposed and unexposed subjects is the same across strata. If there are D^s cases and Y^s person-years of observation in stratum s , then the common rate ratio may be estimated using the method of Mantel and Haenszel by

$$\hat{\theta}_{MH} = \frac{\sum_s D_1^s Y_0^s / Y^s}{\sum_s D_0^s Y_1^s / Y^s}.$$

Note that the strata might not correspond to groups of subjects. For example, if the confounder is age, subjects who cross from one age-band into the next during the study contribute parts of their periods of observation to different strata. This is how Table 11.1 was constructed.

A more general way of controlling for confounding variables is to use Poisson regression to model the number of occurrences of disease

or ‘failures’. Such an approach allows inclusion of several covariates. The complexity of the model can be decided by model selection criteria, often leading to smoothing through the omission of higher order interactions. If a log link is used, the expected number of failures can be made proportional to the person-years of observation by adding the log of the person-years of observation to the linear predictor as an offset (an explanatory variable with regression coefficient set to 1), giving

$$\log[E(D)] = \log(Y) + \beta^T \mathbf{x}.$$

Exponentiating the equation and dividing by Y gives

$$\frac{E(D)}{Y} = \exp(\beta^T \mathbf{x})$$

as required.

11.2.2 Case-control studies

If the incidence rate of a disease is small, a cohort study requires a large number of person-years of observation making it very expensive. A more feasible type of study in this situation is a case-control study in which cases of the disease of interest are compared with non-cases, often called controls, with respect to exposure to possible risk factors in the past. The basic idea of case-control studies is shown in [Figure 11.2](#). The assumption here is that the probability of selection into the study is independent of the exposures of interest. The data in [Table 11.2](#) derive from a case-control study in which cases with lung cancer and healthy controls were interviewed to ascertain whether they had been ‘exposed’ to a pet bird.

Let D and H be the number of cases and controls, respectively, and let the subscripts 0 and 1 denote ‘unexposed’ and ‘exposed’. Since the proportion of cases was determined by the design, it is not possible to estimate the relative risk of disease between exposed and nonexposed subjects. However, the odds of exposure in the cases or controls *can* be estimated, and the ratio of these odds is equal to the odds ratio of being a case in the exposed group compared with the unexposed group

$$\frac{D_1/D_0}{H_1/H_0} = \frac{D_1/H_1}{D_0/H_0}.$$

We model the (log) odds of being a case using logistic regression with the exposure as an explanatory variable. Then the coefficient of the exposure variable is an estimate of the desired log odds ratio even

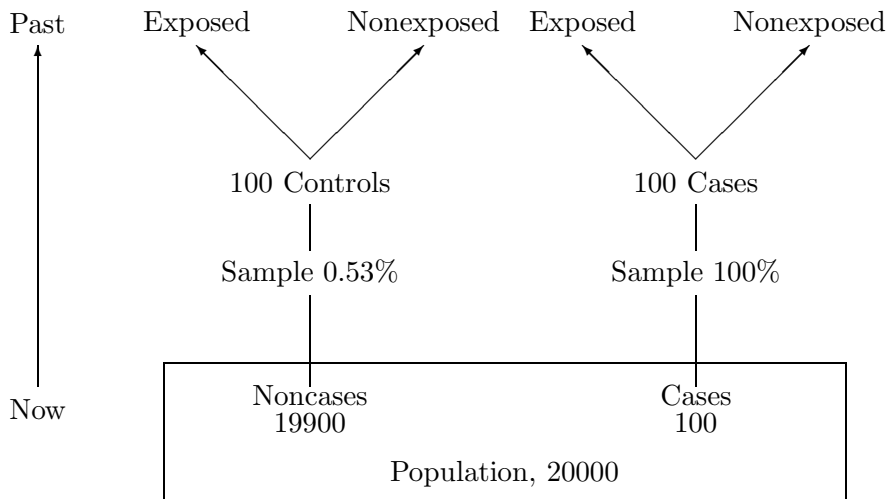


Figure 11.2: Case-control study.

though the estimate of the odds itself (reflected by the constant) is determined by the proportion of cases in the study. Logistic regression is the most popular method for estimating adjusted odds ratios for risk factors of interest, controlling for confounding variables.

11.2.3 *Matched case-control studies*

A major difficulty with case-control studies is to find suitable controls who are similar enough to the cases (so that differences in exposure can reasonably be assumed to be due to their association with the disease) without being overmatched, which can result in very similar exposure patterns. The problem of finding controls who are sufficiently similar is often addressed by matching controls individually to cases according to important variables such as age and sex. Examples of such matched case-control studies are given in [Tables 11.3](#) and [11.4](#). In the screening study, matching had the following additional advantage noted in Clayton and Hills (1993). The screening history of controls could be determined by considering only the period up to the diagnosis of the case, ensuring that cases did not have a decreased opportunity for screening because they would not have been screened after their diagnosis.

The statistical analysis has to take account of the matching. Two methods of analysis are McNemar's test in the simple case of 2×2

tables and conditional logistic regression in the case of several controls per case and/or several explanatory variables. Since the case-control sets have been matched on variables that are believed to be associated with disease status, the sets can be thought of as strata with subjects in one stratum having higher or lower odds of being a case than those in another after controlling for the exposures. A logistic regression model would have to accommodate these differences by including a parameter α_c for each case-control set c , so that the log odds of being a case for subject i in case-control set c would be

$$\log(\Omega_{ci}) = \log(\alpha_c) + \beta^T \mathbf{x}_i. \quad (11.1)$$

However, this would result in too many parameters to be estimated (the incidental parameter problem). Furthermore, the parameters α_c are of no interest to us.

In conditional logistic regression, the nuisance parameters α_c are eliminated as follows. In a 1:1 matched case-control study, ignoring the fact that each set has one case, the probability that subject 1 in the set is a case and subject 2 is a noncase is

$$\Pr(1) = \frac{\Omega_{c1}}{1 + \Omega_{c1}} \times \frac{1}{1 + \Omega_{c2}},$$

and the probability that subject 1 is a noncase and subject 2 is a case is

$$\Pr(2) = \frac{1}{1 + \Omega_{c1}} \times \frac{\Omega_{c2}}{1 + \Omega_{c2}}.$$

However, conditional on there being one case in a set, the probability of subject 1 being the case is simply

$$\frac{\Pr(1)}{\Pr(1) + \Pr(2)} = \Omega_{c1}/(\Omega_{c1} + \Omega_{c2}) = \frac{\exp(\beta^T \mathbf{x}_1)}{\exp(\beta^T \mathbf{x}_1) + \exp(\beta^T \mathbf{x}_2)}, \quad (11.2)$$

since α_c cancels out; see equation (11.1). The expression on the right-hand side of equation (11.2) is the contribution of a single case-control set to the conditional likelihood of the sample. Similarly, it can be shown that if there are k controls per case and the subjects within each case-control set are labeled 1 for the case and 2 to $k + 1$ for the controls then the log likelihood becomes

$$\sum_c \log \left(\frac{\exp(\beta^T \mathbf{x}_1)}{\sum_{i=1}^{k+1} \exp(\beta^T \mathbf{x}_i)} \right).$$

11.3 Analysis using Stata

11.3.1 Cohort study

There is a collection of instructions in Stata, the `epitab` commands, that may be used to analyze small tables in epidemiology. These commands either refer to variables in an existing dataset or can take cell counts as arguments (i.e., they are immediate commands; see [Chapter 1](#)).

The first cohort dataset in [Table 11.1](#) is given in a file as tabulated and may be read using

```
infile str5 age num1 py1 num0 py0 using ihd.dat,clear
gen agegr=_n
reshape long num py, i(agegr) j(exposed)
```

Ignoring `age`, the incidence rate ratio may be estimated using

```
ir num exposed py
```

giving the table in Display 11.1. The incidence rate ratio of ischemic

	exposed			
	Exposed	Unexposed	Total	
num	28	17	45	
py	1857	2769	4626	
Incidence Rate	.0150781	.0061394	.0097276	
	Point estimate		[95% Conf. Interval]	
Inc. rate diff.	.0089387		.0026372	.0152401
Inc. rate ratio	2.455954		1.298154	4.782555 (exact)
Attr. frac. ex.	.5928262		.2296752	.7909067 (exact)
Attr. frac. pop	.3688696			
	(midp)	Pr(k>=28) =		0.0016 (exact)
	(midp)	2*Pr(k>=28) =		0.0031 (exact)

Display 11.1

heart disease, comparing low energy with high energy intake, is estimated as 2.46 with a 95% confidence interval from 1.29 to 4.78. (Note that we could report the reciprocals of these figures if we wished to consider high energy intake as the risk factor.) The terms (exact) imply that the confidence intervals are exact (no approximation was used).

Controlling for age using the `epitab` command

```
ir num exposed py, by(age)
```

(see Display 11.2) gives very similar estimates as shown in the row

age	IRR	[95% Conf. Interval]		M-H Weight	
40-49	.974359	.0881386	6.798632	1.356522	(exact)
50-59	3.476993	1.14023	12.59828	2.04186	(exact)
60-69	2.328967	.9118074	6.407519	3.432241	(exact)
Crude	2.455954	1.298154	4.782555		(exact)
M-H combined	2.403126	1.306459	4.420356		

Test of homogeneity (M-H) $\chi^2(2) =$ 1.57 $\Pr > \chi^2 =$ 0.4554

Display 11.2

labeled `M-H combined` (the Mantel-Haenszel estimate).

Another way of controlling for age is to carry out Poisson regression with the log of `py` as an offset. The exponentiated offset `py` may be specified using the `exposure(py)` option:

```
xi: poisson num exposed i.age, exposure(py) irr
```

(see Display 11.3) showing that there is an estimated age-adjusted in-

i.age	_Iage_1-3		(_Iage_1 for age==40-49 omitted)			
Iteration 0:	log likelihood = -11.899364					
Iteration 1:	log likelihood = -11.898229					
Iteration 2:	log likelihood = -11.898228					
Poisson regression				Number of obs	=	6
				LR chi2(3)	=	12.91
				Prob > chi2	=	0.0048
Log likelihood = -11.898228				Pseudo R2	=	0.3516

num	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
exposed	2.386096	.7350226	2.82	0.005	1.304609	4.364108
_Iage_2	1.137701	.5408325	0.27	0.786	.4481154	2.888461
_Iage_3	1.997803	.9218379	1.50	0.134	.8086976	4.935362
py	(exposure)					

Display 11.3

cidence rate ratio of 2.39 with a 95% confidence interval from 1.30 to 4.36. The coefficients of `_Iage_2` and `_Iage_3` show that the incidence increases with age (although the rate ratios for adjacent age groups are not significant) as would be expected. Another way of achieving the same result is using `glm` with the `lnoffset()` option

```
xi: glm num exposed i.age, fam(poisson) link(log) /*
    */ lnoffset(lpy) eform
```

where the `eform` option is used to obtain exponentiated coefficients (incidence rate ratios instead of their logarithms). An advantage of this modeling approach is that we can investigate the possibility of an interaction between `exposed` and `age`. If there were more age categories, we could attempt to model the effect of age as a smooth function.

11.3.2 Case-control study

We will analyze the case-control study using the ‘immediate’ command `cci`. The following notation is used for `cci`:

	Exposed	Unexposed
Cases	<i>a</i>	<i>b</i>
Noncases	<i>c</i>	<i>d</i>

where the quantities *a*, *b*, etc. in the table are specified in alphabetical order, i.e.,

```
cci a b c d
```

(See `help epitab` for the arguments required for other immediate epitab commands.) The bird data may therefore be analyzed as follows:

```
cci 98 141 101 328
```

giving the output in Display 11.4. The odds ratio of lung cancer, comparing those with pet birds with those without pet birds, is estimated as 2.26 with an exact 95% confidence interval from 1.58 to 3.22. The *p*-value for the null hypothesis of no association between pet birds and lung cancer is < 0.001. This *p*-value is based on a chi-squared test; an exact *p*-value could be obtained using the `exact` option.

11.3.3 Matched case-control studies

The matched case-control study with *one control* per case may be analyzed using the immediate command `mcci` which requires four numbers

	Exposed	Unexposed	Total	Proportion Exposed
Cases	98	141	239	0.4100
Controls	101	328	429	0.2354
Total	199	469	668	0.2979
	Point estimate		[95% Conf. Interval]	
Odds ratio	2.257145		1.580935	3.218756 (exact)
Attr. frac. ex.	.5569624		.367463	.689321 (exact)
Attr. frac. pop	.2283779			
chi2(1) = 22.37 Pr>chi2 = 0.0000				

Display 11.4

a to d defined as in [Table 11.3](#) where the columns are exposed and un-exposed controls wheareas the rows are exposed and unexposed cases:

```
mcci 12 43 7 121
```

The results in Display 11.5 suggest that there is an increased odds of endometrial cancer in subjects exposed to oral conjugated estrogens (odds ratio = 2.89, 95% confidence interval from 1.89 to 4.44).

Cases	Controls		Total
	Exposed	Unexposed	
Exposed	12	43	55
Unexposed	7	121	128
Total	19	164	183

McNemar's chi2(1) = 25.92 Prob > chi2 = 0.0000
Exact McNemar significance probability = 0.0000

Proportion with factor

Cases	.3005464		
Controls	.1038251	[95% Conf. Interval]	
difference	.1967213	.1210924	.2723502
ratio	2.894737	1.885462	4.444269
rel. diff.	.2195122	.1448549	.2941695
odds ratio	6.142857	2.739772	16.18458 (exact)

Display 11.5

The matched case-control study with *three controls* per case cannot be analyzed using **epitab**. Instead, we will use conditional logistic regression. We need to convert the data in [Table 11.4](#) into the form required for conditional logistic regression; that is, one observation per subject (including cases and controls); an indicator variable, **cancer**, for cases; another indicator variable, **screen**, for screening and a third variable, **caseid**, an identifier for each case-control set of four women.

First, read the data which are in the form shown in Table 11.4. Then transpose the data so that the first column contains frequencies for unscreened cases (variable **ncases0**) and the second for screened cases (variable **ncases1**).

```
infile v1-v4 using screen.dat,clear
gen str8 _varname= cond(_n==1,"ncases1","ncases0")
xpose, clear
```

The four rows in this transposed dataset correspond to 0, 1, 2, and 3 matched controls who have been screened. We will define a variable **nconscr** taking on these four values. We can then stack the two columns into a single variable **ncases** and create an indicator **casescr** for whether or not the case was screened using the **reshape** command:

```
gen nconscr=_n-1
reshape long ncases, i(nconscr) j(casescr)
list
```

(see Display 11.6). The next step is to replicate each of the records

	nconscr	casescr	ncases
1.	0	0	11
2.	0	1	1
3.	1	0	10
4.	1	1	4
5.	2	0	12
6.	2	1	3
7.	3	0	4
8.	3	1	1

Display 11.6

ncases times so that we have one record per case-control set. Then define the variable **caseid**, and expand the dataset four times in order

to have one record per subject. The four subjects within each case-control set are arbitrarily labeled 0 to 3 in the variable `control` where 0 stands for ‘the case’ and 1, 2, and 3 for the controls.

```
expand ncases
sort casescr nconscr
gen caseid=_n
expand 4
quietly bysort caseid: gen control=_n-1
list in 1/8
```

(see Display 11.7). Now `screen`, the indicator whether a subject was

	nconscr	casescr	ncases	caseid	control
1.	0	0	11	1	0
2.	0	0	11	1	1
3.	0	0	11	1	2
4.	0	0	11	1	3
5.	0	0	11	2	0
6.	0	0	11	2	1
7.	0	0	11	2	2
8.	0	0	11	2	3

Display 11.7

screened, is defined to be 0 except for the cases who were screened and for as many controls as were screened according to `nconscr`. The variable `cancer` is 1 for cases and 0 otherwise.

```
gen screen=0
replace screen=1 if control==0&casescr==1 /* the case */
replace screen=1 if control==1&nconscr>0
replace screen=1 if control==2&nconscr>1
replace screen=1 if control==3&nconscr>2
gen cancer=control==0
```

We can reproduce [Table 11.4](#) by temporarily collapsing the data (using `preserve` and `restore` to revert to the original data) as follows:

```
preserve
collapse (sum) screen (mean) casescr , by(caseid)
gen nconscr=screen-casescr
tabulate casescr nconscr
```

restore
(see Display 11.8).

(mean) casescr	nconscr				Total
	0	1	2	3	
0	11	10	12	4	37
1	1	4	3	1	9
Total	12	14	15	5	46

Display 11.8

We are now ready to carry out conditional logistic regression:

clogit cancer screen, group(caseid) or
(see Display 11.9). Screening therefore seems to be protective of death

Iteration 0: log likelihood = -62.404527						
Iteration 1: log likelihood = -59.212727						
Iteration 2: log likelihood = -59.18163						
Iteration 3: log likelihood = -59.181616						
Conditional (fixed-effects) logistic regression			Number of obs	=	184	
			LR chi2(1)	=	9.18	
			Prob > chi2	=	0.0025	
Log likelihood = -59.181616			Pseudo R2	=	0.0719	

cancer	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
screen	.2995581	.1278367	-2.82	0.005	.1297866	.6914043

Display 11.9

from breast cancer, reducing the odds to about a third (95% confidence interval from 0.13 to 0.69).

11.4 Exercises

1. Carry out conditional logistic regression to estimate the odds ratio for the data in [Table 11.3](#). The data are given in the same form as

in the table in a file called `estrogen.dat`.

2. For the data `ihd.dat`, use the command `iri` to calculate the incidence rate ratio for IHD without controlling for age.
3. Again for data in `ihd.dat`, use Poisson regression to test whether the effect of exposure on incidence of IHD differs between age groups.

Chapter 12

Survival Analysis: Retention of Heroin Addicts in Methadone Maintenance Treatment

12.1 Description of data

The data to be analyzed in this chapter are on 131 heroin addicts in two different clinics receiving methadone maintenance treatment to help them overcome their addiction. Early dropout is an important problem with this treatment. We will therefore analyze the time from admission to termination of treatment (in days), given as **time** in Table 12.1. For patients still in treatment when these data were collected, **time** is the time from admission to the time of data collection. The variable **status** is an indicator for whether **time** refers to dropout (1) or end of study (0). Possible explanatory variables for retention in treatment are maximum methadone dose and a prison record as well as which of two clinics the addict was treated in. These variables are called **dose**, **prison**, and **clinic**, respectively. The data were first analyzed by Caplehorn and Bell (1991) and also appear in Hand *et al.* (1994).

Table 12.1 Data in heroin.dat

id	clinic	status	time	prison	dose	id	clinic	status	time	prison	dose
1	1	1	428	0	50	132	2	0	633	0	70
2	1	1	275	1	55	133	2	1	661	0	40
3	1	1	262	0	55	134	2	1	232	1	70
4	1	1	183	0	30	135	2	1	13	1	60

Table 12.1 Data in heroin.dat (continued)

id	clinic	status	time	prison	dose	id	clinic	status	time	prison	dose
5	1	1	259	1	65	137	2	0	563	0	70
6	1	1	714	0	55	138	2	0	969	0	80
7	1	1	438	1	65	143	2	0	1052	0	80
8	1	0	796	1	60	144	2	0	944	1	80
9	1	1	892	0	50	145	2	0	881	0	80
10	1	1	393	1	65	146	2	1	190	1	50
11	1	0	161	1	80	148	2	1	79	0	40
12	1	1	836	1	60	149	2	0	884	1	50
13	1	1	523	0	55	150	2	1	170	0	40
14	1	1	612	0	70	153	2	1	286	0	45
15	1	1	212	1	60	156	2	0	358	0	60
16	1	1	399	1	60	158	2	0	326	1	60
17	1	1	771	1	75	159	2	0	769	1	40
18	1	1	514	1	80	160	2	1	161	0	40
19	1	1	512	0	80	161	2	0	564	1	80
21	1	1	624	1	80	162	2	1	268	1	70
22	1	1	209	1	60	163	2	0	611	1	40
23	1	1	341	1	60	164	2	1	322	0	55
24	1	1	299	0	55	165	2	0	1076	1	80
25	1	0	826	0	80	166	2	0	2	1	40
26	1	1	262	1	65	168	2	0	788	0	70
27	1	0	566	1	45	169	2	0	575	0	80
28	1	1	368	1	55	170	2	1	109	1	70
30	1	1	302	1	50	171	2	0	730	1	80
31	1	0	602	0	60	172	2	0	790	0	90
32	1	1	652	0	80	173	2	0	456	1	70
33	1	1	293	0	65	175	2	1	231	1	60
34	1	0	564	0	60	176	2	1	143	1	70
36	1	1	394	1	55	177	2	0	86	1	40
37	1	1	755	1	65	178	2	0	1021	0	80
38	1	1	591	0	55	179	2	0	684	1	80
39	1	0	787	0	80	180	2	1	878	1	60
40	1	1	739	0	60	181	2	1	216	0	100
41	1	1	550	1	60	182	2	0	808	0	60
42	1	1	837	0	60	183	2	1	268	1	40
43	1	1	612	0	65	184	2	0	222	0	40
44	1	0	581	0	70	186	2	0	683	0	100
45	1	1	523	0	60	187	2	0	496	0	40
46	1	1	504	1	60	188	2	1	389	0	55
48	1	1	785	1	80	189	1	1	126	1	75
49	1	1	774	1	65	190	1	1	17	1	40
50	1	1	560	0	65	192	1	1	350	0	60
51	1	1	160	0	35	193	2	0	531	1	65
52	1	1	482	0	30	194	1	0	317	1	50
53	1	1	518	0	65	195	1	0	461	1	75
54	1	1	683	0	50	196	1	1	37	0	60
55	1	1	147	0	65	197	1	1	167	1	55
57	1	1	563	1	70	198	1	1	358	0	45
58	1	1	646	1	60	199	1	1	49	0	60
59	1	1	899	0	60	200	1	1	457	1	40
60	1	1	857	0	60	201	1	1	127	0	20
61	1	1	180	1	70	202	1	1	7	1	40
62	1	1	452	0	60	203	1	1	29	1	60
63	1	1	760	0	60	204	1	1	62	0	40
64	1	1	496	0	65	205	1	0	150	1	60
65	1	1	258	1	40	206	1	1	223	1	40
66	1	1	181	1	60	207	1	0	129	1	40
67	1	1	386	0	60	208	1	0	204	1	65
68	1	0	439	0	80	209	1	1	129	1	50
69	1	0	563	0	75	210	1	1	581	0	65
70	1	1	337	0	65	211	1	1	176	0	55
71	1	0	613	1	60	212	1	1	30	0	60
72	1	1	192	1	80	213	1	1	41	0	60
73	1	0	405	0	80	214	1	0	543	0	40
74	1	1	667	0	50	215	1	0	210	1	50

Table 12.1 Data in heroin.dat (continued)

id	clinic	status	time	prison	dose	id	clinic	status	time	prison	dose
75	1	0	905	0	80	216	1	1	193	1	70
76	1	1	247	0	70	217	1	1	434	0	55
77	1	1	821	0	80	218	1	1	367	0	45
78	1	1	821	1	75	219	1	1	348	1	60
79	1	0	517	0	45	220	1	0	28	0	50
80	1	0	346	1	60	221	1	0	337	0	40
81	1	1	294	0	65	222	1	0	175	1	60
82	1	1	244	1	60	223	2	1	149	1	80
83	1	1	95	1	60	224	1	1	546	1	50
84	1	1	376	1	55	225	1	1	84	0	45
85	1	1	212	0	40	226	1	0	283	1	80
86	1	1	96	0	70	227	1	1	533	0	55
87	1	1	532	0	80	228	1	1	207	1	50
88	1	1	522	1	70	229	1	1	216	0	50
89	1	1	679	0	35	230	1	0	28	0	50
90	1	0	408	0	50	231	1	1	67	1	50
91	1	0	840	0	80	232	1	0	62	1	60
92	1	0	148	1	65	233	1	0	111	0	55
93	1	1	168	0	65	234	1	1	257	1	60
94	1	1	489	0	80	235	1	1	136	1	55
95	1	0	541	0	80	236	1	0	342	0	60
96	1	1	205	0	50	237	2	1	41	0	40
97	1	0	475	1	75	238	2	0	531	1	45
98	1	1	237	0	45	239	1	0	98	0	40
99	1	1	517	0	70	240	1	1	145	1	55
100	1	1	749	0	70	241	1	1	50	0	50
101	1	1	150	1	80	242	1	0	53	0	50
102	1	1	465	0	65	243	1	0	103	1	50
103	2	1	708	1	60	244	1	0	2	1	60
104	2	0	713	0	50	245	1	1	157	1	60
105	2	0	146	0	50	246	1	1	75	1	55
106	2	1	450	0	55	247	1	1	19	1	40
109	2	0	555	0	80	248	1	1	35	0	60
110	2	1	460	0	50	249	2	0	394	1	80
111	2	0	53	1	60	250	1	1	117	0	40
113	2	1	122	1	60	251	1	1	175	1	60
114	2	1	35	1	40	252	1	1	180	1	60
118	2	0	532	0	70	253	1	1	314	0	70
119	2	0	684	0	65	254	1	0	480	0	50
120	2	0	769	1	70	255	1	0	325	1	60
121	2	0	591	0	70	256	2	1	280	0	90
122	2	0	769	1	40	257	1	1	204	0	50
123	2	0	609	1	100	258	2	1	366	0	55
124	2	0	932	1	80	259	2	0	531	1	50
125	2	0	932	1	80	260	1	1	59	1	45
126	2	0	587	0	110	261	1	1	33	1	60
127	2	1	26	0	40	262	2	1	540	0	80
128	2	0	72	1	40	263	2	0	551	0	65
129	2	0	641	0	70	264	1	1	90	0	40
131	2	0	367	0	70	266	1	1	47	0	45

The data can be described as *survival data*, although the ‘endpoint’ is not death in this case, but dropout from treatment. From engineering applications, another commonly used term for the endpoint is ‘failure’. Duration or survival data can generally not be analyzed by conventional methods such as linear regression. The main reason for this is that some durations are usually right-censored; that is, the endpoint of interest has not occurred during the period of observation and all that is known about the duration is that it exceeds the observation period. In the present dataset, this applies to all observations where **status**

is 0. Another reason why conventional linear regression would not be appropriate is that survival times tend to have positively skewed distributions. A third reason is that time-varying covariates, such as the time of year, could not be handled. In the next section, we therefore describe methods specifically developed for survival data.

12.2 Survival analysis

12.2.1 Introduction

The survival time T may be regarded as a random variable with a probability distribution $F(t)$ and probability density function $f(t)$. An obvious quantity of interest is the probability of surviving to time t or beyond, the *survivor function* or survival curve $S(t)$, which is given by

$$S(t) = P(T \geq t) = 1 - F(t). \quad (12.1)$$

A further function which is of interest for survival data is the *hazard function*. This represents the instantaneous failure rate, that is, the probability that an individual experiences the event of interest at a time point given that the event has not yet occurred. It can be shown that the hazard function is given by

$$h(t) = \frac{f(t)}{S(t)}, \quad (12.2)$$

the instantaneous probability of failure at time t divided by the probability of surviving up to time t . Note that the hazard function is just the incidence rate discussed in [Chapter 11](#). It follows from equations (12.1) and (12.2) that

$$-\frac{d \log(S(t))}{dt} = h(t),$$

so that

$$S(t) = \exp(-H(t)), \quad (12.3)$$

where $H(t)$ is the integrated hazard function, also known as the *cumulative hazard* function.

12.2.2 Kaplan-Meier estimator

The Kaplan-Meier estimator is a nonparametric estimator of the survivor function $S(t)$. If all the failure times, or times at which the event occurs in the sample, are ordered and labeled $t_{(j)}$ such that $t_{(1)} \leq t_{(2)} \leq \dots \leq t_{(n)}$, the estimator is given by

$$\hat{S}(t) = \prod_{j|t_{(j)} \leq t} \left(1 - \frac{d_j}{n_j}\right),$$

where d_j is the number of individuals who experience the event at time $t_{(j)}$, and n_j is the number of individuals who have not yet experienced the event at that time and are therefore still ‘at risk’ of experiencing it (including those censored at $t_{(j)}$). The product is over all failure times less than or equal to t .

12.2.3 Cox Regression

We can compare survival in different subgroups by plotting the Kaplan-Meier estimators of the group-specific survivor functions and applying simple significance tests (such as the log-rank test). However, when there are several explanatory variables, and in particular when some of these are continuous, it is much more useful to use a regression method such as Cox regression. Here the hazard function for individual i is modeled as

$$h_i(t) = h_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_i), \quad (12.4)$$

where $h_0(t)$ is the *baseline hazard function*, $\boldsymbol{\beta}$ are regression coefficients, and \mathbf{x}_i covariates. The baseline hazard is the hazard when all covariates are zero, and this quantity is left unspecified. This nonparametric treatment of the baseline hazard combined with a parametric representation of the effects of covariates gives rise to the term *semiparametric* model. The main assumption of the model is that the hazard of any individual i is a time-constant multiple of the hazard function of any other individual j , the factor being $\exp(\boldsymbol{\beta}^T (\mathbf{x}_i - \mathbf{x}_j))$, the *hazard ratio* or incidence rate ratio. This property is called the *proportional hazards* assumption. The exponentiated regression parameters can therefore be interpreted as hazard ratios when the corresponding explanatory variable increases by one unit if all other covariates remain constant.

The parameters $\boldsymbol{\beta}$ are estimated by maximizing the *partial log like-*

likelihood given by

$$\sum_f \log \left(\frac{\exp(\boldsymbol{\beta}^T \mathbf{x}_f)}{\sum_{i \in r(f)} \exp(\boldsymbol{\beta}^T \mathbf{x}_i)} \right) \quad (12.5)$$

where the first summation is over all failures f , and the second summation is over all subjects $r(f)$ who are still at risk at the time of failure. It can be shown that this log likelihood is a log profile likelihood (i.e., the log of the likelihood in which the baseline hazard parameters have been replaced by functions of $\boldsymbol{\beta}$ which maximize the likelihood for fixed $\boldsymbol{\beta}$). Note also that the likelihood in equation (12.5) is equivalent to the likelihood for matched case-control studies described in [Chapter 11](#) if the subjects at risk at the time of a failure (the *risk set*) are regarded as controls matched to the case failing at that point in time (see Clayton and Hills, 1993).

The baseline hazard function may be estimated by maximizing the full log likelihood with the regression parameters evaluated at their estimated values, giving nonzero values only when a failure occurs. Integrating the hazard function gives the cumulative hazard function

$$H_i(t) = H_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_i), \quad (12.6)$$

where $H_0(t)$ is the integral of $h_0(t)$. The survival curve may be obtained from $H(t)$ using equation (12.3). This leads to the Kaplan-Meier estimator when there are no covariates.

It follows from equation (12.3) that the survival curve for a Cox model is given by

$$S_i(t) = S_0(t)^{\exp(\boldsymbol{\beta}^T \mathbf{x}_i)}. \quad (12.7)$$

The log of the cumulative hazard function predicted by the Cox model is given by

$$\log(H_i(t)) = \log H_0(t) + \boldsymbol{\beta}^T \mathbf{x}_i, \quad (12.8)$$

so that the log cumulative hazard functions of any two subjects i and j are parallel with constant difference given by $\boldsymbol{\beta}^T(\mathbf{x}_i - \mathbf{x}_j)$.

Stratified Cox regression can be used to relax the assumption of proportional hazards for a categorical predictor. The partial likelihood of a stratified Cox model has the same form as equation (12.5) except that the risk set for each failure is now confined to subjects in the same stratum as the subject contributing to the numerator.

Survival analysis is described in Allison (1984), Clayton and Hills (1993), Collett (2003), and Klein and Moeschberger (2003). Cleves *et al.* (2004) discuss survival analysis using Stata.

12.3 Analysis using Stata

The data are available as an ASCII file called `heroin.dat` on the disk accompanying Hand *et al.* (1994). Since the data are stored in a two-column format with the set of variables repeated twice in each row, as shown in Table 12.1, we have to use `reshape` to bring the data into the usual form:

```
infile id1 clinic1 status1 time1 prison1 dose1 /*
      */ id2 clinic2 status2 time2 prison2 dose2 /*
      */ using heroin.dat
gen row=_n
reshape long id clinic status time prison dose, /*
      */ i(row) j(col)
drop row col
```

Before fitting any survival models, we declare the data as being of the form `st` (for survival time) using the `stset` command

```
stset time, failure(status)

      failure event:  status != 0 & status < .
obs. time interval:  (0, time]
exit on or before:  failure
```

238	total obs.	
0	exclusions	

238	obs. remaining, representing	
150	failures in single record/single failure data	
95812	total analysis time at risk, at risk from t =	0
	earliest observed entry t =	0
	last observed exit t =	1076

and look at a summary of the data using

```
stsum
```

	failure _d: status					
	analysis time _t: time					
	time at risk	incidence rate	no. of subjects	Survival time		
				25%	50%	75%
total	95812	.0015656	238	212	504	821

There are 238 subjects with a median survival time of 504 days. If the incidence rate (i.e., the hazard function) could be assumed to be constant, it would be estimated as 0.0016 per day which corresponds to 0.57 per year.

The Kaplan-Meier estimator of the survivor functions for the two clinics are obtained and plotted using

```
sts graph, by(clinic)
```

giving the graph in Figure 12.1. Dropout seems to occur more quickly in clinic 1.

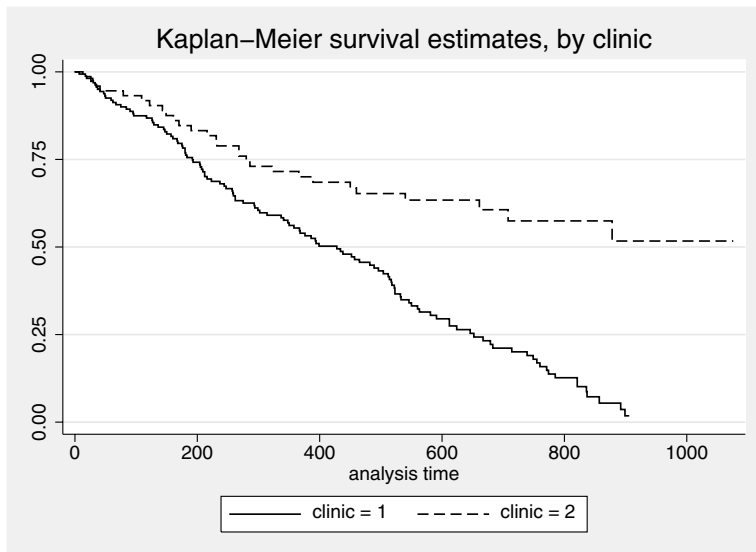


Figure 12.1: Kaplan-Meier survival curves.

To investigate the effects of **dose** and **prison** on survival, we will use Cox regression. We will allow the hazard functions for the two clinics to be non-proportional. A Cox regression model with clinics as strata is fitted using the **stcox** command with the **strata()** option:

```
stcox dose prison, strata(clinic)
```

giving the output shown in Display 12.1. Therefore, subjects with a prison history are 47.5% more likely to drop out at any given time (given that they remained in treatment until that time) than those without a prison history. For every increase in methadone dose by one

```

failure _d: status
analysis time _t: time
Iteration 0: log likelihood = -614.68365
Iteration 1: log likelihood = -597.73516
Iteration 2: log likelihood = -597.714
Refining estimates:
Iteration 0: log likelihood = -597.714
Stratified Cox regr. -- Breslow method for ties
No. of subjects = 238 Number of obs = 238
No. of failures = 150
Time at risk = 95812
Log likelihood = -597.714 LR chi2(2) = 33.94
Prob > chi2 = 0.0000

```

_t	Haz. Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
dose	.9654655	.0062418	-5.44	0.000	.953309	.977777
prison	1.475192	.2491827	2.30	0.021	1.059418	2.054138

Stratified by clinic

Display 12.1

unit (1 mg), the hazard is multiplied by 0.965. This coefficient is very close to one, but this may be because one unit of methadone dose is not a large quantity. Even if we know little about methadone maintenance treatment, we can assess how much one unit of methadone dose is by finding the sample standard deviation:

```
summarize dose
```

Variable	Obs	Mean	Std. Dev.	Min	Max
dose	238	60.39916	14.45013	20	110

indicating that a unit is not much at all; subjects often differ from each other by 10 to 15 units. To find the hazard ratio of two subjects differing by one standard deviation, we need to raise the hazard ratio to the power of one standard deviation, giving $0.9654655^{14.45013} = 0.60179167$. We can obtain the same result (with greater precision) using the stored macros `_b[dose]` for the log hazard ratio and `r(Var)` for the variance,

```
disp exp(_b[dose]*sqrt(r(Var)))

.60178874
```

In the above calculation, we simply rescaled the regression coefficient before taking the exponential. To obtain this hazard ratio in the Cox regression, we need to standardize `dose` to have unit standard deviation.

In the command below we also standardize to mean zero, although this will make no difference to the estimated coefficients (only the baseline hazards are affected):

```
egen zdose=std(dose)
```

We repeat the Cox regression with the option `bases(s)` which results in the baseline survival function $S_0(t)$ being estimated and stored in `s`:

```
stcox zdose prison, strata(clinic) bases(s)
```

(see Display 12.2). The coefficient of `zdose` is identical to that calcu-

failure _d:	status					
analysis time _t:	time					
Iteration 0:	log likelihood =	-614.68365				
Iteration 1:	log likelihood =	-597.73516				
Iteration 2:	log likelihood =	-597.714				
Refining estimates:						
Iteration 0:	log likelihood =	-597.714				
Stratified Cox regr. -- Breslow method for ties						
No. of subjects =	238				Number of obs =	238
No. of failures =	150					
Time at risk =	95812					
					LR chi2(2) =	33.94
Log likelihood =	-597.714				Prob > chi2 =	0.0000

_t	Haz. Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
zdose	.6017887	.0562195	-5.44	0.000	.5010998	.7227097
prison	1.475192	.2491827	2.30	0.021	1.059418	2.054138

Stratified by clinic

Display 12.2

lated previously and may now be interpreted as indicating a decrease of the hazard by 40% when the methadone dose increases by one standard deviation.

Assuming the variables `prison` and `zdose` satisfy the proportional hazards assumption (see Section 12.3.1), we now present the model graphically. To do this, we will plot the predicted survival curves separately for the two clinics and for those with and without a prison record where `zdose` is evaluated at its clinic-specific mean. Such a graph may be produced by using `stcox` with the `bases()` option to generate a variable containing the predicted baseline survival fractions and then applying equation (12.7) to obtain the predicted survival fractions for

particular covariate values:

```
stcox zdose prison, strata(clinic) bases(s)
egen mdose=mean(zdose), by(clinic)
gen surv=s^exp(_b[zdose]*mdose + _b[prison]*prison)
```

Note that these survival fractions represent predicted values for subjects having the clinic-specific mean dose. We now transform `time` to time in years, and plot the survival curves separately for each clinic:

```
gen tt=time/365.25
label variable tt "time in years"
label define clin 1 "Clinic 1" 2 "Clinic 2"
label values clinic clin

sort clinic time
twoway (line surv tt if prison==0, connect(stairstep)) /*
      */ (line surv tt if prison==1, connect(stairstep)) /*
      */ clpat(dash)), by(clinic) ylabel(0(0.2)1) /*
      */ legend(order(1 "Prison record" 2 "No prison record"))
```

Here the `connect(stairstep)` option was used to produce the step shaped survival curves shown in Figure 12.2. According to Caplehorn

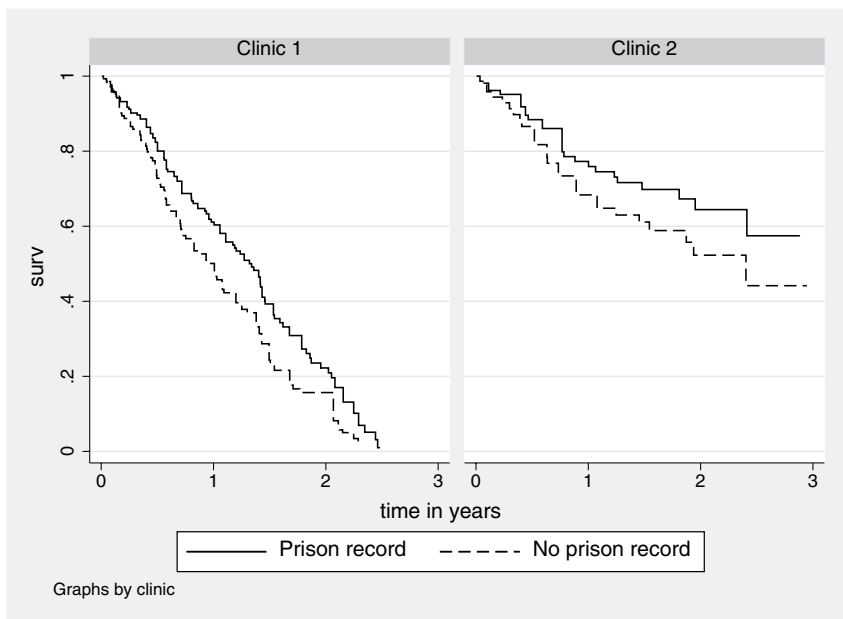


Figure 12.2: Survival curves.

and Bell (1991), the more rapid decline in the proportion remaining in clinic 1 compared with clinic 2 may be due to the policy of clinic 1 to attempt to limit the duration of maintenance to two years.

The partial likelihood is appropriate for continuous survival times, which should theoretically never take on the same value for any two individuals or be *tied*. However, in practice survival times are measured in discrete units, days in the present example, and ties will frequently occur. For example, subjects 61 and 252 in clinic 1 both dropped out after 180 days. By default `stcox` uses *Breslow's method* for ties, where the risk sets in (12.5) contain all subjects who failed *at* or after the failure time of the subject contributing to the numerator. For a group of subjects with tied survival times, the contributions to the partial likelihood therefore each have the same denominator. However, risk sets usually decrease by one after each failure. In *Efron's method*, contributions to the risk set from the subjects with tied failure times are therefore downweighted in successive risk sets. In the *exact method* (referred to as 'exact marginal log likelihood' in [ST] `stcox`, the Stata reference manual for survival analysis), the contribution to the partial likelihood from a group of tied survival times is the sum, over all possible orderings (or permutations) of the tied survival times, of the contributions to the partial likelihood corresponding to these orderings. Efron's method can be obtained using the `efron` option and the exact method using the `exactm` option (see Exercise 4).

12.3.1 Assessing the proportional hazards assumption

12.3.1.1 Graphical methods

We now discuss methods for assessing the proportional hazards assumption. A graphical approach is available for categorical predictors if there are sufficient observations for each value of the predictor. In this case the model is first estimated by stratifying on the categorical predictor of interest, thus not making any assumption regarding the relationship between the baseline hazards for different values of the predictor or strata. The log cumulative baseline hazards for the strata are then derived from the estimated model and plotted against time. According to equation (12.8), the resulting curves should be parallel if the proportional hazards assumption holds. Here we demonstrate this method for the variable `clinic`. The cumulative baseline hazard can be obtained using `stcox` with the `basechazard()` option as follows:

```
quietly stcox zdose prison, strata(clinic) basech(ch)
```

We now compute and then plot the logarithm of the cumulative baseline hazard function using


```

gen lh = log(ch)
sort time
twoway (line lh time if clinic==1, connect(stairstep)) /*
      */ (line lh time if clinic==2, connect(stairstep)) /*
      */ clpat(dash)), xtitle("Time in days")           /*
      */ ytitle("Log cumulative hazard")                 /*
      */ legend(order(1 "Clinic 1" 2 "Clinic 2"))

```

giving the graph shown in Figure 12.3. Clearly, the curves are not

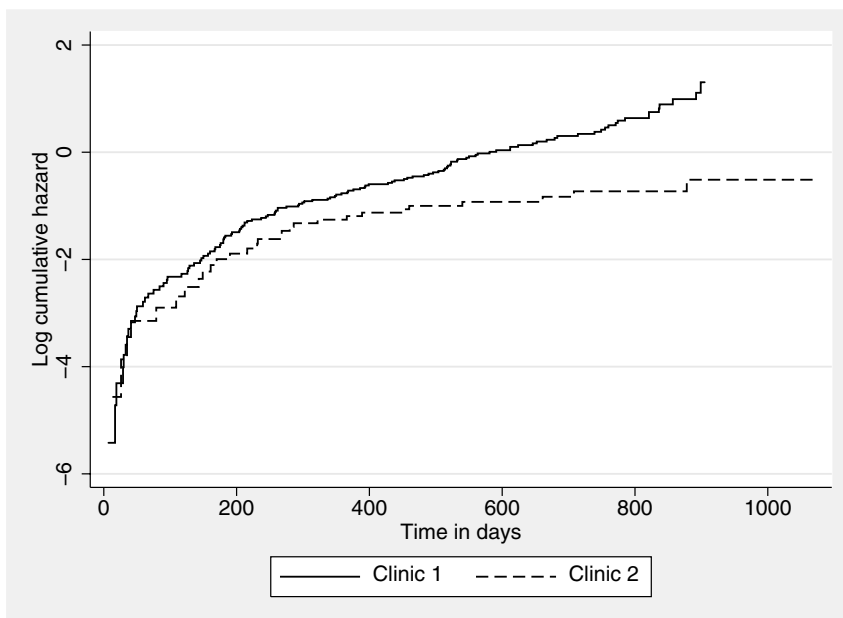


Figure 12.3: Log of minus the log of the survival functions for the two clinics estimated by stratified Cox regression.

parallel, and we will therefore continue treating the clinics as strata. Note that a quicker way of producing a similar graph would be to use the `stphplot` command as follows:

```

stphplot, strata(clinic) adjust(zdose prison) zero /*
      */ xlabel(1/7)

```

Here the `adjust()` option specifies the covariates to be used in the Cox regression, and the `zero` option specifies that these covariates are to be evaluated at zero. As a result, minus the logs of the cumulative baseline hazard functions (stratified by clinic) are plotted against the

log of the survival time, see Figure 12.4.

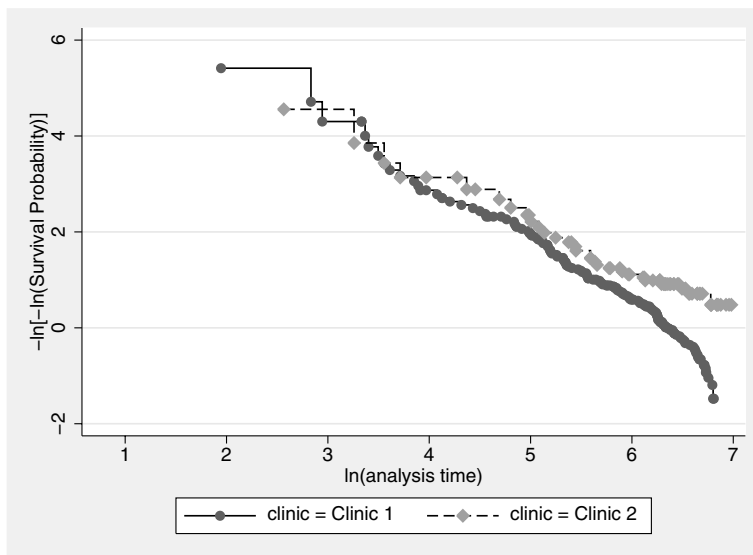


Figure 12.4: Minus the log of minus the log of the survival functions for the two clinics versus log survival time, estimated by stratified Cox regression.

To determine whether the hazard functions for those with and without a prison history are proportional, we could split the data into four strata by `clinic` and `prison`. However, as the strata get smaller, the estimated survival functions become less precise (because the risk sets in equation (12.5) become smaller). Also, a similar method could not be used to check the proportional hazard assumption for the continuous variable `zdose` without splitting it into arbitrary categories.

12.3.1.2 Time-varying covariates

Another way of testing the proportional hazards assumption of `zdose`, say, is to introduce a time-varying covariate equal to the interaction between time (since admission) and `zdose`, thus allowing the effect of `zdose` to change over time. To estimate this model, the terms in equation (12.5) need to be evaluated for values of the time-varying covariates *at the times of the failure* in the numerator. These values are not available for the denominator in the present dataset since each subject is represented only once, at the time of their own failure (and

not at all previous failure times). One possibility is to create the required dataset (see below). A simpler option is to simply use the `stcox` command with the `tv` and `te` options:

```
stcox zdose prison, strata(clinic) tv(zdose) /*
      */ te((_t-504)/365.25)
```

The `tv` option specifies the variable that should interact with (a function of) time and the `te` option specifies the function of time to be used. Here we have simply subtracted the median survival time so that the effect of the explanatory variable `zdose` can be interpreted as the effect of `zdose` at the median survival time. We have divided by 365.25 to see by how much the effect of `zdose` changes between intervals of one year. The output is shown in Display 12.3 where the

failure_d: status

analysis time_t: time

Iteration 0: log likelihood = -614.68365

Iteration 1: log likelihood = -597.32655

Iteration 2: log likelihood = -597.29131

Iteration 3: log likelihood = -597.29131

Refining estimates:

Iteration 0: log likelihood = -597.29131

Stratified Cox regr. -- Breslow method for ties

No. of subjects = 238

Number of obs = 238

No. of failures = 150

Time at risk = 95812

LR chi2(3) = 34.78

Log likelihood = -597.29131

Prob > chi2 = 0.0000

_t	Haz. Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
rh						
zdose	.6442974	.0767535	-3.69	0.000	.5101348	.8137442
prison	1.481193	.249978	2.33	0.020	1.064036	2.061899
t						
zdose	1.147853	.1720104	0.92	0.357	.8557175	1.539722

Stratified by clinic

Note: Second equation contains variables that continuously vary with respect to time; variables are interacted with current values of (_t-504)/365.25.

Display 12.3

estimated increase in the hazard ratio for `zdose` is 15% per year. This small effect is not significant at the 5% level which is confirmed by carrying out the likelihood ratio test as follows:

```
estimates store model1
quietly stcox zdose prison, strata(clinic)
lrtest model1 .
```

```
likelihood-ratio test                                LR chi2(1) =      0.85
(Assumption: . nested in model1)                   Prob > chi2 =    0.3579
```

giving a very similar p -value as before and confirming that there is no evidence that the effect of dose on the hazard varies with time. A similar test can be carried out for **prison** (see Exercise 6).

Although Stata makes it very easy to include an interaction between a variable and a function of time, inclusion of other time-varying covariates, or of more than a single time-varying covariate, requires an expanded version of the current dataset. In the expanded dataset each subject's record should appear (at least) as many times as that subject contributes to a risk set in equation (12.5), with the time variable equal to the corresponding failure times. This can be achieved very easily using the **stsplit** command, but only after defining an **id** variable using **stset**:

```
stset time, failure(status) id(id)
stsplit, at(failures) strata(clinic)
```

The **stsplit** command generates new time and censoring variables **_t** and **_d**, respectively. For subject 103, these are listed using

```
sort id _t
list _t _d if id==103, clean noobs
```

giving the values shown in Display 12.4. The last value of **_t** (708) is just the value of the original variable **time**, the subject's survival or censoring time, whereas the previous values are all unique survival times (at which failures occurred) in the same stratum (clinic 2) *which are less than* the subject's own survival time. These 'invented' survival times are times beyond which the subject survives, so the censoring variable **_d** is set to zero for all invented times and equal to **status** for the original survival time. This new survival dataset is equivalent to the original one, and we obtain the same results as before if we run

```
stcox zdose prison, strata(clinic)
```

(output not shown). However, we can now create time-varying covariates making use of the new time variable **_t**. To assess the proportional hazards assumption for **zdose**, we generate an interaction between **zdose** and the linear transformation of **_t** we used previously:

```
gen tdose=zdose*(t-504)/365.25
```

_t	_d
13	0
26	0
35	0
41	0
79	0
109	0
122	0
143	0
149	0
161	0
170	0
190	0
216	0
231	0
232	0
268	0
280	0
286	0
322	0
366	0
389	0
450	0
460	0
540	0
661	0
708	1

Display 12.4

We now fit the Cox regression, allowing the effect of `zdose` to vary with time:

```
stcox zdose tdose prison, strata(clinic)
```

giving the same result as before in Display 12.5.

failure _d:	status					
analysis time _t:	time					
	id:	id				
Iteration 0:	log likelihood =	-614.68365				
Iteration 1:	log likelihood =	-597.32655				
Iteration 2:	log likelihood =	-597.29131				
Iteration 3:	log likelihood =	-597.29131				
Refining estimates:						
Iteration 0:	log likelihood =	-597.29131				
Stratified Cox regr. -- Breslow method for ties						
No. of subjects =	238				Number of obs =	11228
No. of failures =	150					
Time at risk =	95812					
					LR chi2(3) =	34.78
Log likelihood =	-597.29131				Prob > chi2 =	0.0000

	_t	Haz. Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
	zdose	.6442974	.0767535	-3.69	0.000	.5101348	.8137442
	prison	1.481193	.249978	2.33	0.020	1.064036	2.061899
	tdose	1.147853	.1720104	0.92	0.357	.8557175	1.539722

Stratified by clinic

Display 12.5

We can restore the original data using the `stjoin` command after deleting any time-varying covariates (apart from `_t` and `_d`):

```
drop tdose
stjoin
```

```
(option censored(0) assumed)
(10990 obs. eliminated)
```

A test of proportional hazards based on rescaled Schoenfeld or efficient score residuals (see below), suggested by Grambsch and Therneau (1994), is also available using the `stphtest` command (see for example Cleves *et al.*, 2002).

12.3.2 Residuals

It is a good idea to produce some residual plots, for example a graph of the deviance residuals against the linear predictor. In order for `predict` to be able to compute the deviance residuals, we must first store the martingale residuals (see for example Collett, 2003) using `stcox` with the `mgale()` option:

```
stcox zdose prison, strata(clinic) mgale(mart)
predict devr, deviance
```

A scatterplot is produced using

```
predict xb, xb
twoway scatter devr xb, mlabel(id) mlabpos(0) /*
    */ msymbol(none)
```

with the result shown in Figure 12.5. There appear to be no serious outliers.

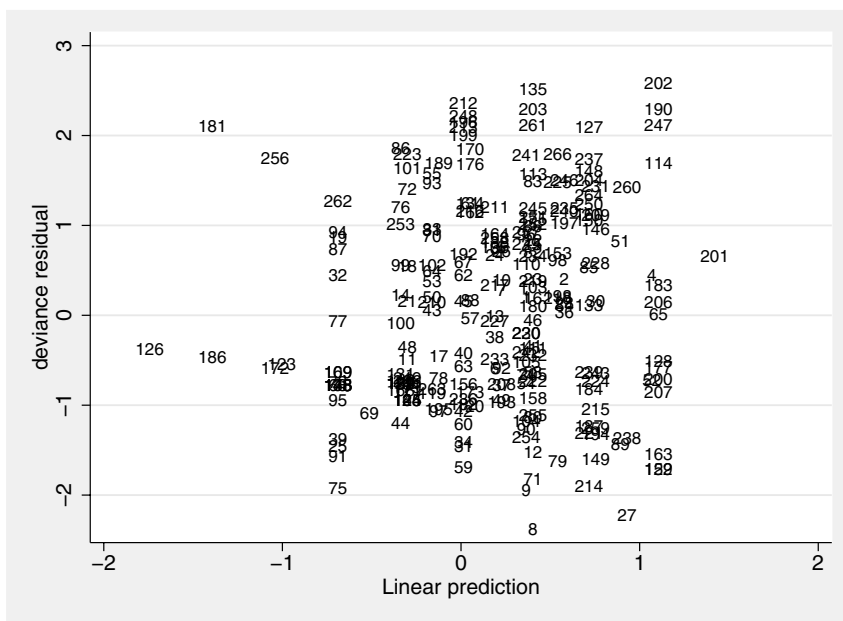


Figure 12.5: Deviance residuals for survival analysis.

Another type of residual is the Schoenfeld or efficient score residual, defined as the first derivative of the partial log likelihood function with respect to an explanatory variable. The score residual is large in absolute value if a case's explanatory variable differs substantially from

the explanatory variables of subjects whose estimated risk of failure is large at the case's time of failure or censoring. Since our model has two explanatory variables, we can compute the efficient score residuals for `zdose` and `prison` and store them in `score1` and `score2` using `stcox` with the `esr` option:

```
stcox zdose prison, strata(clinic) esr(score*)
```

These residuals can be plotted against survival time using

```
twoway scatter score1 tt, mlabel(id) mlabpos(0) msymbol(none)
```

and similarly for `score2`. The resulting graphs are shown in Figures 12.6 and 12.7. Subject 89 has a low value of `zdose` (-1.75) compared with other subjects at risk of failure at such a late time. Subjects 8, 27, 12, and 71 drop out relatively late considering that they have a police record, whereas others remaining beyond their time of dropout (or censoring) tend to have no police record.

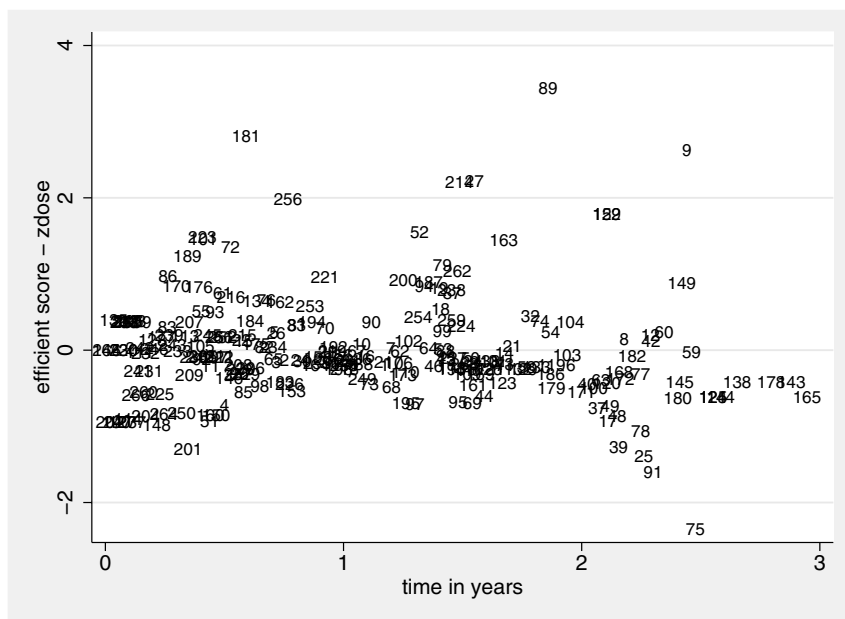


Figure 12.6: Score residuals for `zdose`.

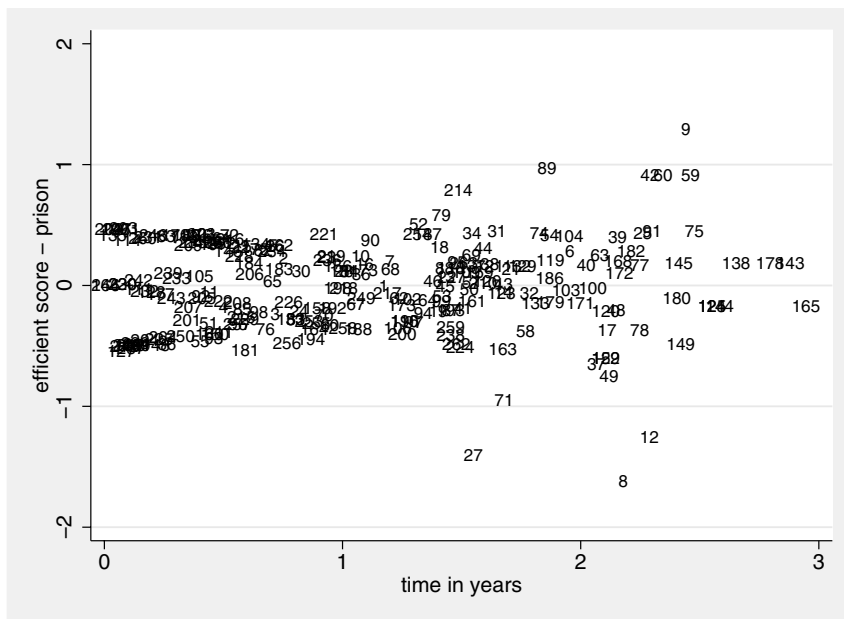


Figure 12.7: Score residuals for prison.

12.4 Exercises

1. In the original analysis of this data, Caplehorn and Bell (1991) judged that the hazards were approximately proportional for the first 450 days (see [Figure 12.3](#)). They therefore analyzed the data for this time period using `clinic` as a covariate instead of stratifying by clinic. Repeat this analysis, using `prison` and `dose` as further covariates.
2. Following Caplehorn and Bell (1991), repeat the above analysis treating dose as a categorical variable with three levels (< 60 , $60 - 79$, ≥ 80), and plot the predicted survival curves for the three dose categories when `prison` and `clinic` take on one of their values.
3. Test for an interaction between clinic and the methadone dose, treating dose as both continuous and categorical.
4. For the model treating dose as categorical and containing no interaction, compare the estimates using three different methods of handling ties: the Breslow, Efron, and exact methods.
5. Check the proportional hazards assumption for `prison` using the same method we used for `dose`.

Chapter 13

Maximum Likelihood Estimation: Age of Onset of Schizophrenia

13.1 Description of data

Table 13.1 gives the ages of onset of schizophrenia (determined as age on first admission) for 99 women. These data will be used to investigate whether there is any evidence for the subtype model of schizophrenia (see Lewine, 1981), according to which there are two types of schizophrenia characterized by early and late onset.

13.2 Finite mixture distributions

The most common type of finite mixture distribution for continuous responses is a mixture of univariate normal distributions of the form

$$f(y_i; \mathbf{p}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = p_1 g(y_i; \mu_1, \sigma_1) + p_2 g(y_i; \mu_2, \sigma_2) + \cdots + p_k g(y_i; \mu_k, \sigma_k),$$

where $g(y; \mu, \sigma)$ is the normal or Gaussian density with mean μ and standard deviation σ ,

$$g(y; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{y - \mu}{\sigma} \right)^2 \right\}, \quad (13.1)$$

Table 13.1 Age of onset of schizophrenia
data in `onset.dat`

Ages of onset in <code>onset.dat</code>							
20	30	21	23	30	25	13	19
16	25	20	25	27	43	6	21
15	26	23	21	23	23	34	14
17	18	21	16	35	32	48	53
51	48	29	25	44	23	36	58
28	51	40	43	21	48	17	23
28	44	28	21	31	22	56	60
15	21	30	26	28	23	21	20
43	39	40	26	50	17	17	23
44	30	35	20	41	18	39	27
28	30	34	33	30	29	46	36
58	28	30	28	37	31	29	32
48	49	30					

and p_1, \dots, p_k are mixing probabilities.

The parameters p_1, \dots, p_k , μ_1, \dots, μ_k , and $\sigma_1, \dots, \sigma_k$ are usually estimated by maximum likelihood. Standard errors can be obtained in the usual way from the observed information matrix (i.e., from the inverse of the Hessian matrix, the matrix of second derivatives of the log likelihood). Determining the number k of components in the mixture is more problematic since the conventional likelihood ratio test cannot be used to compare models with different k , a point we will return to later.

For a short introduction to finite mixture modeling, see Everitt (1996); a more comprehensive account is given in McLachlan and Peel (2000). Maximum likelihood estimation using Stata is described in detail by Gould *et al.* (2003).

13.3 Analysis using Stata

Stata has a command called `m1`, which can be used to maximize a user-specified log likelihood using the Newton-Raphson algorithm. The algorithm is iterative. Starting with initial parameter values, the program evaluates the first and second derivatives of the log likelihood at the parameter values to find a new set of parameter values where the likelihood is likely to be greater. The derivatives are then evaluated at the new parameters to update the parameters again, etc., until the maximum has been found (where the first derivatives are zero and the second derivatives negative). The EM algorithm, an alternative to Newton-Raphson, is often believed to be superior for finite mixture

models. However, in our experience the implementation of Newton-Raphson in Stata works very well for these models.

To use `ml`, the user needs to write a program that evaluates the log likelihood and possibly its derivatives. `ml` provides four likelihood-evaluator methods: `d0`, `d1`, `d2`, and `lf`. The `d0` method does not require the user's program to evaluate any derivatives of the log likelihood, `d1` requires first derivatives only, and `d2` requires both first and second derivatives. When `d0` is chosen, first and second derivatives are found numerically; this makes this alternative slower and less accurate than `d1` or `d2`.

The simplest approach to use is `lf` which also does not require any derivatives to be programmed. Instead, the structure of most likelihood problems is used to increase both the speed and the accuracy of the numerical differentiation. Whereas `d0` to `d2` can be used for any maximum likelihood problem, method `lf` may only be used if the likelihood satisfies the following two criteria:

1. The observations in the dataset are independent, i.e., the log likelihood is the sum of the log likelihood contributions of the observations.
2. The log likelihood contributions have a *linear form*, i.e., they are (not necessarily linear) functions of linear predictors of the form $\eta_i = x_{1i}\beta_1 + \cdots + x_{ki}\beta_k$.

The first restriction is usually met, an exception being clustered data. The second restriction is not as severe as it appears because there may be several linear predictors, as we shall see later. These restrictions allow `lf` to evaluate derivatives efficiently and accurately using chain rules. For example, first derivatives are obtained as

$$\frac{\partial \ell_i}{\partial \beta_1} = \frac{\partial \ell_i}{\partial \eta_i} x_{1i},$$

where ℓ_i is the log likelihood contribution from the i th observation. All that is required are the derivatives with respect to the linear predictor(s) from which the derivatives with respect to the individual parameters follow automatically.

In this chapter, we will give only a brief introduction to maximum likelihood estimation using Stata, restricting our examples to the `lf` method. We recommend the book on *Maximum Likelihood Estimation with Stata* by Gould *et al.* (2003) for a thorough treatment of the topic.

We will eventually fit a mixture of normals to the age of onset data, but will introduce the `ml` procedure by a series of simpler models.

13.3.1 Single normal density

To begin with, we will fit a normal distribution with standard deviation fixed at 1 to a set of simulated data. A (pseudo)-random sample from the normal distribution with mean 5 and standard deviation 1 may be obtained using the instructions

```
clear
set obs 100
set seed 12345678
gen y = invnorm(uniform()))+5
```

where the purpose of the `set seed` command is simply to ensure that the same data will be generated each time we repeat this sequence of commands. We use `summarize` to confirm that the sample has a mean close to 5 and a standard deviation close to 1.

```
summarize y
```

Variable	Obs	Mean	Std. Dev.	Min	Max
y	100	5.002311	1.053095	2.112869	7.351898

First, define a program, `mixing0`, to evaluate the log likelihood contributions when called from `ml`. The function must have two arguments; the first is the variable name where `ml` will look for the computed log likelihood contributions; the second is the variable name containing the ‘current’ value of the linear predictor during the iterative maximization procedure:

```
capture program drop mixing0
program mixing0
    version 8.1
    args lj xb
    tempname s
    scalar `s' = 1
    quietly replace `lj' = ln(normden($ML_y1,`xb',`s'))
end
```

After giving names `lj` and `xb` to the arguments passed to `mixing0` by `ml`, `mixing0` defines a temporary name stored in the local macro `s` and sets it equal to the standard deviation (1) in the next command. The reason for defining `s` is to make subsequent commands easier to read. Using temporary names avoids any confusion with variables that may exist in the dataset. The final command returns the log of the normal density in ``lj'` as required by the calling program. Here we used the `normden(y, mean, sd)` function to calculate the normal density in

equation (13.1) for the dependent variable whose name is stored in the global macro `ML_y1` by `m1`. The mean is just the linear predictor ``xb'`, and the standard deviation is the scalar ``s'`.

Note that all variables defined within a program to be used with `m1` should be of storage type `double` to enable `m1` to estimate accurate numerical derivatives. Scalars should be used instead of local macros to hold constants as scalars have higher precision.

To define the model use

```
m1 model lf mixing0 (xb: y=)
```

which specifies the method as `lf` and the program to evaluate the log likelihood contributions as `mixing0`. The response and explanatory variable are given by the 'equation' in parentheses. Here the name before the colon, `xb`, is the name of the equation, the variable after the colon, `y`, is the response variable, and the variables after the '=' are the explanatory variables contributing to the linear predictor. No explanatory variables are given here, so a constant only model will be fitted.

As a result of this model definition, the global `ML_y1` will be equal to `y` and in `mixing0` ``xb'` will be equal to the intercept parameter (the mean) that is going to be estimated by `m1`.

Now maximize the likelihood using

```
m1 maximize, noheader
```

giving the results shown in Display 13.1. The program converged in

```
initial:      log likelihood = -1397.9454
alternative:  log likelihood = -1160.3298
rescale:      log likelihood = -197.02113
Iteration 0:  log likelihood = -197.02113
Iteration 1:  log likelihood = -146.79244
Iteration 2:  log likelihood = -146.78981
Iteration 3:  log likelihood = -146.78981
```

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_cons	5.002311	.1	50.02	0.000	4.806314	5.198307

Display 13.1

three iterations and the maximum likelihood estimate of the mean is equal to the sample mean of 5.002311. If we were interested in observing the value of the mean parameter in each iteration, we could use the

trace option in the `ml maximize` command. We used the `noheader` option to suppress output relating to the likelihood ratio test against a ‘null’ model since we have not specified such a model.

We now need to extend the program step by step until it can be used to estimate a mixture of two Gaussians. The first step is to allow the standard deviation to be estimated. Since this parameter does not contribute linearly to the linear predictor used to estimate the mean, we need to define another linear predictor by specifying another equation with no dependent variable in the `ml model` command. Assuming that the program to evaluate the log likelihood contributions is called `mixing1`, the `ml model` command becomes:

```
ml model lf mixing1 (xb: y=) (lsd:)
```

The new equation has the name `lsd`, has no dependent variable (since `y` is the only dependent variable), and the linear predictor is simply a constant. A short-form of the above command is

```
ml model lf mixing1 (xb: y=) /lsd
```

We intend to use `lsd` as the log standard deviation. Estimating the log of the standard deviation will ensure that the standard deviation itself is positive. We now need to modify the function `mixing0` so that it has an additional argument for the log standard deviation:

```
capture program drop mixing1
program mixing1
    version 8.1
    args lj xb ls

    tempvar s

    quietly gen double `s' = exp(`ls')
    quietly replace `lj' = ln(normden($ML_y1,`xb',`s'))
end
```

We now define a temporary *variable* `s` instead of a scalar because, in principle, the linear predictor ``ls'`, defined in the `ml model` command, could contain covariates (see exercises) and hence differ between observations. The temporary variable name will not clash with any existing variable names, and the variable will automatically be deleted when the program has finished running. Running

```
ml maximize, noheader
```

gives the output shown in Display 13.2.

The standard deviation estimate is obtained by exponentiating the estimated coefficient of `_cons` in equation `lsd`. Instead of typing `display exp(0.0467083)`, we can use the following syntax for accessing coefficients and their standard errors:

```

initial:      log likelihood = -1397.9454
alternative:  log likelihood = -534.94948
rescale:      log likelihood = -301.15405
rescale eq:   log likelihood = -180.56802
Iteration 0:  log likelihood = -180.56802
Iteration 1:  log likelihood = -147.50379
Iteration 2:  log likelihood = -146.58464
Iteration 3:  log likelihood = -146.56469
Iteration 4:  log likelihood = -146.56469

```

	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
xb							
	_cons	5.002311	.1047816	47.74	0.000	4.796942	5.207679
lsd							
	_cons	.0467083	.0707107	0.66	0.509	-.0918821	.1852986

Display 13.2

```
display [lsd]_b[_cons]
```

```
.04670833
```

```
display [lsd]_se[_cons]
```

```
.07071068
```

We can also omit ‘_b’ from the first expression, and compute the required standard deviation using

```
display exp([lsd]_var[_cons])
```

```
1.0478163
```

This is smaller than the sample standard deviation from `summarize` because the maximum likelihood estimate of the standard deviation is given by

$$\hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}, \quad (13.2)$$

where n is the sample size, whereas the factor $\frac{1}{n-1}$ is used in `summarize`. Since n is 100 in this case, the maximum likelihood estimate must

be ‘blown up’ by a factor of $\sqrt{100/99}$ to obtain the sample standard deviation:

```
display exp([lsd][_cons])*sqrt(100/99)

1.053095
```

13.3.2 Two-component mixture model

The program can now be extended to estimate a mixture of two Gaussians. To allow us to test the program on data from a known distribution, we will simulate a sample from a mixture of two Gaussians with standard deviations equal to 1 and means equal to 0 and 5 and with mixing probabilities $p_1 = p_2 = 0.5$. This can be done in two stages; first randomly allocate observations to groups (variable **z**) with probabilities p_1 and p_2 , and then sample from the different component densities according to group membership.

```
clear
set obs 300
set seed 12345678
gen z = cond(uniform()<0.5,1,2)
gen y = invnorm(uniform())
replace y = y + 5 if z==2
```

We now need five equations, one for each parameter to be estimated: μ_1 , μ_2 , σ_1 , σ_2 , and p_1 (since $p_2 = 1 - p_1$). As before, we can ensure that the estimated standard deviations are positive by taking the exponential inside the program. The mixing proportion p_1 must lie in the range $0 \leq p_1 \leq 1$. One way of ensuring this is to interpret the linear predictor as representing the log odds (see [Chapter 6](#)) so that p_1 is obtained from the linear predictor of the log odds, **lo1** using the transformation $1/(1+\exp(-\text{lo1}))$. The program now becomes

```
capture program drop mixing2
program mixing2
    version 8.1
    args lj xb1 xb2 lo1 ls1 ls2

    tempvar f1 f2 p s1 s2

    quietly {
        gen double `s1' = exp(`ls1')
        gen double `s2' = exp(`ls2')
        gen double `p' = 1/(1+exp(-`lo1'))
```

```

        gen double `f1' = normden($ML_y1,`xb1',`s1')
        gen double `f2' = normden($ML_y1,`xb2',`s2')
        replace `lj' = ln(`p'*`f1' + (1-`p')*`f2')
    }
end

```

Here we have used `applied quietly` to a whole block of commands by enclosing them in braces.

Stata simply uses zero as initial values for all parameters. However, it is not advisable here to start with the same initial value for both component means. Therefore the starting values should be set using the `ml init` commands as follows:

```

ml model lf mixing2 (xb1: y=) /xb2 /lo1 /lsd1 /lsd2
ml init 1 6 0 0.2 -0.2, copy
ml maximize, noheader

```

In the `ml init` command, the first two values are initial values for the means, the third for the log odds, and the fourth and fifth for the logs of the standard deviations.

The results are shown in Display 13.3 where the standard deviations

```

initial:      log likelihood = -746.68918
rescale:      log likelihood = -746.68918
rescale eq:   log likelihood = -676.61764
Iteration 0:   log likelihood = -676.61764    (not concave)
Iteration 1:   log likelihood = -630.70988
Iteration 2:   log likelihood = -625.88897
Iteration 3:   log likelihood = -622.23406
Iteration 4:   log likelihood = -622.21162
Iteration 5:   log likelihood = -622.21162

```

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
xb1						
_cons	-.0457452	.0806427	-0.57	0.571	-.203802	.1123117
xb2						
_cons	4.983717	.0863492	57.72	0.000	4.814476	5.152958
lo1						
_cons	.1122415	.1172032	0.96	0.338	-.1174725	.3419555
lsd1						
_cons	-.027468	.0627809	-0.44	0.662	-.1505164	.0955804
lsd2						
_cons	-.0173444	.0663055	-0.26	0.794	-.1473008	.1126121

Display 13.3

are estimated as

```
display exp([lsd1][_cons])

.97290582
```

and

```
display exp([lsd2][_cons])

.9828052
```

and the probability of membership in group 1 as

```
display 1/(1 + exp(-[lo1][_cons]))

.52803094
```

The maximum likelihood estimates agree quite closely with the true parameter values.

Alternative estimates of the mixing proportion and means and standard deviations, treating group membership as known (usually not possible!), are obtained using

```
table z, contents(freq mean y sd y)
```

z	Freq.	mean(y)	sd(y)
1	160	-.0206726	1.003254
2	140	5.012208	.954237

and these are also similar to the maximum likelihood estimates. The maximum likelihood estimate of the proportion (0.528) is closer to the realized proportion $160/300 = 0.533$ than the ‘true’ proportion 0.5.

The standard errors of the estimated means are given in the regression table in Display 13.3 (0.081 and 0.086). We can estimate the standard errors of the standard deviations and of the probability from the standard errors of the log standard deviations and log odds using the delta method (see for example Agresti, 2002, pages 577-581). According to the delta method, if $y = f(x)$, then approximately, $se(y) = |f'(x)|se(x)$ where $f'(x)$ is the first derivative of $f(x)$ with respect to x evaluated at the estimated value of x . For the standard deviation, $sd = \exp(lsd)$, so that, by the delta method,

$$se(sd) = sd \times se(lsd). \quad (13.3)$$

For the probability, $p = 1/(1 + \exp(-lo))$, so that

$$se(p) = p(1 - p) \times se(lo). \quad (13.4)$$

However, an even easier way of obtaining and displaying a function of coefficients with the correct standard error in Stata is using the `nlcom` command:

```
nlcom (sd1: exp([_b_1])) (sd2: exp([_b_2])) /*
      */ (p: invlogit([_b_1]))
```

```
sd1: exp([_b_1])
sd2: exp([_b_2])
p: invlogit([_b_1])
```

y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
sd1	.9729058	.0610799	15.93	0.000	.8531913	1.09262
sd2	.9828052	.0651654	15.08	0.000	.8550833	1.110527
p	.5280309	.0292087	18.08	0.000	.4707829	.585279

Display 13.4

In each set of parentheses, we specify a label, followed by a colon and then an expression defining the function of stored estimates we are interested in. Here we used the `invlogit()` function to obtain the probability from the log odds. Stata then uses numerical derivatives to work out the correct standard error using the delta-method giving the results shown in Display 13.4. The z -statistic, p -value and confidence interval should be ignored unless it is reasonable to assume a normal sampling distribution for the derived parameter.

We can now apply the same program to the age of onset data. The data can be read in using

```
infile y using onset.dat, clear
label variable y "age of onset of schizophrenia"
```

A useful graphical display of the data is a histogram produced using

```
histogram y, bin(12)
```

which is shown in [Figure 13.1](#).

It seems reasonable to use initial values of 20 and 45 for the two means. In addition, we will use a mixing proportion of 0.5 and log standard deviations of 2 as initial values.

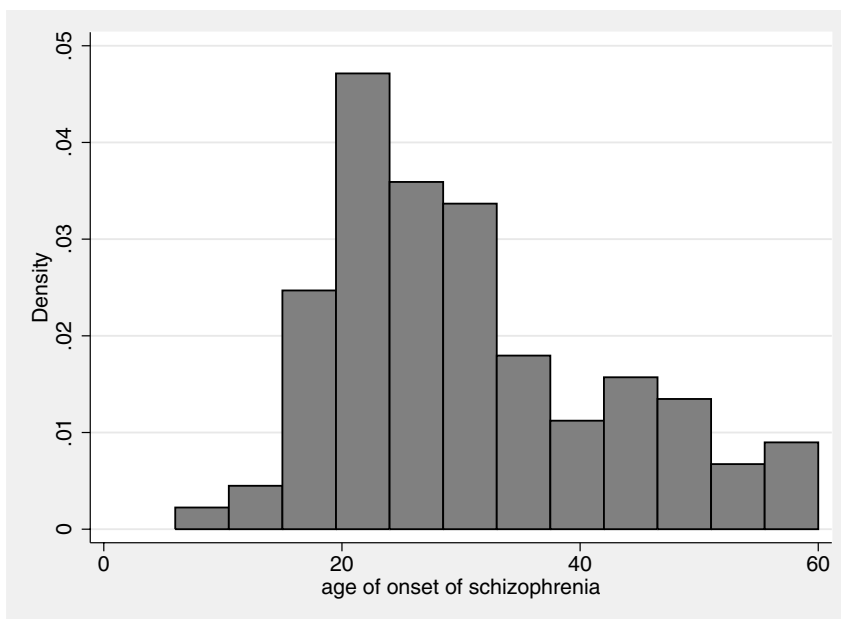


Figure 13.1: Histogram of age of onset of schizophrenia in women.

```
ml model lf mixing2 (y: y=) /xb2 /lo1 /lsd1 /lsd2
ml init 20 45 0 2 2, copy
ml maximize, noheader
```

The output is given in Display 13.5. The means are estimated as 24.8 and 46.4, the standard deviations are estimated as 6.5 and 7.1 for groups 1 and 2, respectively, and the mixing proportions are estimated as 0.74 and 0.26. The approximate standard errors may be obtained as before; see exercises.

We will now plot the estimated mixture density together with kernel density estimates. Instead of using the command `kdensity`, we will use `twoway kdensity`, allowing us to add the mixture density onto the same graph:

```
graph twoway (kdensity y, width(3)) (function          /*
  /* scalar(p1)*normden(x, [xb1]_cons, scalar(sd1)) /*
  /* +scalar(1-p1)*normden(x, [xb2]_cons, scalar(sd2)) /*
  /* , range(y) clpat(dash)), xtitle("Age")          /*
  /* ytitle("Density") legend(order(1 "Kernel density" /*
  /* 2 "Mixture model"))
```

In [Figure 13.2](#) the two estimates of the density are surprisingly similar. (Admittedly, we did choose the half-width of the kernel using the

```

initial:      log likelihood = -391.61146
rescale:      log likelihood = -391.61146
rescale eq:   log likelihood = -391.61146
Iteration 0:  log likelihood = -391.61146    (not concave)
Iteration 1:  log likelihood = -374.66689    (not concave)
Iteration 2:  log likelihood = -374.36521
Iteration 3:  log likelihood = -373.94028
Iteration 4:  log likelihood = -373.6709
Iteration 5:  log likelihood = -373.66896
Iteration 6:  log likelihood = -373.66896

```

	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
xb1							
	_cons	24.79772	1.133545	21.88	0.000	22.57601	27.01942
xb2							
	_cons	46.44685	2.740866	16.95	0.000	41.07485	51.81885
lo1							
	_cons	1.034415	.3697502	2.80	0.005	.3097178	1.759112
lsd1							
	_cons	1.877698	.1261092	14.89	0.000	1.630529	2.124868
lsd2							
	_cons	1.955009	.25692	7.61	0.000	1.451455	2.458563

Display 13.5

`width()` option to obtain a good fit!)

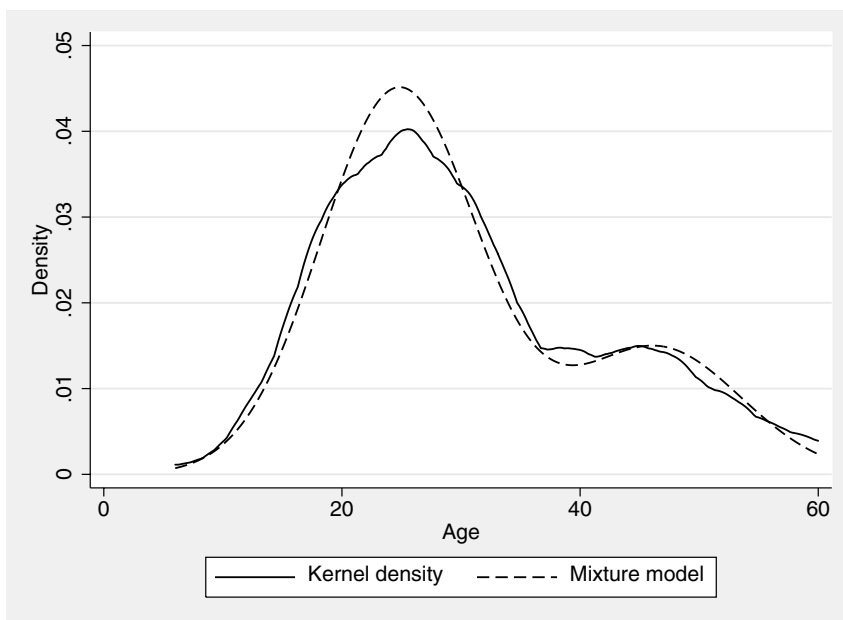


Figure 13.2: Kernel and mixture model densities for the age of onset data.

The histogram and kernel density suggest that there are two subpopulations. To test this more formally, we could also fit a single normal distribution and compare the likelihoods. However, as mentioned earlier, the conventional likelihood ratio test is not valid here. Wolfe (1971) suggests, on the basis of a limited simulation study, that the difference in minus twice the log likelihood for a model with k components compared with a model with $k + 1$ components has approximately a χ^2 distribution with $2\nu - 2$ degrees of freedom: here ν is the number of extra parameters in the $k + 1$ component mixture. The log likelihood of the current model may be accessed using `e(11)`. We store this in a local macro

```
local ll = e(11)
```

and fit the single normal model using the program `mixing1` as follows:

```
ml model lf mixing1 (xb: y=) /lsd
ml init 30 1.9, copy
ml maximize, noheader
```

with the result shown in Display 13.6. Comparing the log likelihoods

```

initial:      log likelihood = -429.14924
rescale:      log likelihood = -429.14924
rescale eq:   log likelihood = -429.14924
Iteration 0:  log likelihood = -429.14924
Iteration 1:  log likelihood = -383.84531
Iteration 2:  log likelihood = -383.39857
Iteration 3:  log likelihood = -383.39585
Iteration 4:  log likelihood = -383.39585

```

	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
xb							
	_cons	30.47475	1.169045	26.07	0.000	28.18346	32.76603
lsd							
	_cons	2.453747	.0710669	34.53	0.000	2.314458	2.593035

Display 13.6

using the method proposed by Wolfe,

```

local chi2 = 2*(`l1'-e(l1))
display chiprob(4,`chi2')

```

```
.00063994
```

confirms that there appear to be two subpopulations.

13.4 Exercises

1. Create a do-file with the commands necessary to fit the mixture model.
2. Add commands to the end of the do-file to calculate the standard deviations and mixing probability and the standard errors of these parameters. What are the standard errors of the estimates for the age of onset data?
3. Simulate values from two normals trying out different values for the various parameters. Do the estimated values tend to lie within two estimated standard errors from the ‘true’ values?
4. Use program `mixing1` to fit a linear regression model to the slimming data from [Chapter 5](#) using only `status` as the explanatory

variable. Compare the standard deviation with the root mean square error obtained using the command **regress**.

5. Use the same program again to fit a linear regression model where the variance is allowed to differ between the groups defined by **status**. Is there any evidence for heteroscedasticity? How do the results compare with those of **sdtest**?
6. Extend the program **mixing2** to fit a mixture of three normals and test this on simulated data (hint: use transformations $p1 = 1/d$, $p2 = \exp(lo1)/d$, and $p3 = \exp(lo2)/d$ where $d = 1 + \exp(lo1) + \exp(lo2)$).

Chapter 14

Principal Components Analysis: Hearing Measurement using an Audiometer

14.1 Description of data

The data in Table 14.1 are adapted from those given in Jackson (1991), and relate to hearing measurements with an instrument called an audiometer. An individual is exposed to a signal of a given frequency with an increasing intensity until the signal is perceived. The lowest intensity at which the signal is perceived is a measure of hearing loss, calibrated in units referred to as *decibel loss* in comparison with a reference standard for that particular instrument. Observations are obtained one ear at a time for a number of frequencies. In this example, the frequencies used were 500 Hz, 1000 Hz, 2000 Hz, and 4000 Hz. The limits of the instrument are -10 to 99 decibels. (A negative value does not imply better than average hearing; the audiometer had a calibration ‘zero’, and these observations are in relation to that.)

**Table 14.1 Data in `hear.dat` (taken from Jackson (1991)
with permission of his publisher, John Wiley & Sons)**

id	l500	l1000	l2000	l4000	r500	r1000	r2000	r4000
1	0	5	10	15	0	5	5	15
2	-5	0	-10	0	0	5	5	15
3	-5	0	15	15	0	0	5	15
4	-5	0	-10	-10	-10	-5	-10	10

Table 14.1 Data in hear.dat (continued)

5	-5	-5	-10	10	0	-10	-10	50
6	5	5	5	-10	0	5	0	20
7	0	0	0	20	5	5	5	10
8	-10	-10	-10	-5	-10	-5	0	5
9	0	0	0	40	0	0	-10	10
10	-5	-5	-10	20	-10	-5	-10	15
11	-10	-5	-5	5	5	0	-10	5
12	5	5	10	25	-5	-5	5	15
13	0	0	-10	15	-10	-10	-10	10
14	5	15	5	60	5	5	0	50
15	5	0	5	15	5	-5	0	25
16	-5	-5	5	30	5	5	5	25
17	0	-10	0	20	0	-10	-10	25
18	5	0	0	50	10	10	5	65
19	-10	0	0	15	-10	-5	5	15
20	-10	-10	-5	0	-10	-5	-5	5
21	-5	-5	-5	35	-5	-5	-10	20
22	5	15	5	20	5	5	5	25
23	-10	-10	-10	25	-5	-10	-10	25
24	-10	0	5	15	-10	-5	5	20
25	0	0	0	20	-5	-5	10	30
26	-10	-5	0	15	0	0	0	10
27	0	0	5	50	5	0	5	40
28	-5	-5	-5	55	-5	5	10	70
29	0	15	0	20	10	-5	0	10
30	-10	-5	0	15	-5	0	10	20
31	-10	-10	5	10	0	0	20	10
32	-5	5	10	25	-5	0	5	10
33	0	5	0	10	-10	0	0	0
34	-10	-10	-10	45	-10	-10	5	45
35	-5	10	20	45	-5	10	35	60
36	-5	-5	-5	30	-5	0	10	40
37	-10	-5	-5	45	-10	-5	-5	50
38	5	5	5	25	-5	-5	5	40
39	-10	-10	-10	0	-10	-10	-10	5
40	10	20	15	10	25	20	10	20
41	-10	-10	-10	20	-10	-10	0	5
42	5	5	-5	40	5	10	0	45
43	-10	0	10	20	-10	0	15	10
44	-10	-10	10	10	-10	-10	5	0
45	-5	-5	-10	35	-5	0	-10	55
46	5	5	10	25	10	5	5	20
47	5	0	10	70	-5	5	15	40
48	5	10	0	15	5	10	0	30
49	-5	-5	5	-10	-10	-5	0	20
50	-5	0	10	55	-10	0	5	50
51	-10	-10	-10	5	-10	-10	-5	0
52	5	10	20	25	0	5	15	0
53	-10	-10	50	25	-10	-10	-10	40
54	5	10	0	-10	0	5	-5	15
55	15	20	10	60	20	20	0	25
56	-10	-10	-10	5	-10	-10	-5	-10
57	-5	-5	-10	30	0	-5	-10	15
58	-5	-5	0	5	-5	-5	0	10
59	-5	5	5	40	0	0	0	10
60	5	10	30	20	5	5	20	60
61	5	5	0	10	-5	5	0	10
62	0	5	10	35	0	0	5	20
63	-10	-10	-10	0	-5	0	-5	0
64	-10	-5	-5	20	-10	-10	-5	5
65	5	10	0	25	5	5	0	15
66	-10	0	5	60	-10	-5	0	65
67	5	10	40	55	0	5	30	40
68	-5	-10	-10	20	-5	-10	-10	15
69	-5	-5	-5	20	-5	0	0	0
70	-5	-5	-5	5	-5	0	0	5

Table 14.1 Data in hear.dat (continued)

71	0	10	40	60	-5	0	25	50
72	-5	-5	-5	-5	-5	-5	-5	5
73	0	5	45	50	0	10	15	50
74	-5	-5	10	25	-10	-5	25	60
75	0	-10	0	60	15	0	5	50
76	-5	0	10	35	-10	0	0	15
77	5	0	0	15	0	5	5	25
78	15	15	5	35	10	15	-5	0
79	-10	-10	-10	5	-5	-5	-5	5
80	-10	-10	-5	15	-10	-10	-5	5
81	0	-5	5	35	-5	-5	5	15
82	-5	-5	-5	10	-5	-5	-5	5
83	-5	-5	-10	-10	0	-5	-10	0
84	5	10	10	20	-5	0	0	10
85	-10	-10	-10	5	-10	-5	-10	20
86	5	5	10	0	0	5	5	5
87	-10	0	-5	-10	-10	0	0	-10
88	-10	-10	10	15	0	0	5	15
89	-5	0	10	25	-5	0	5	10
90	5	0	-10	-10	10	0	0	0
91	0	0	5	15	5	0	0	5
92	-5	0	-5	0	-5	-5	-10	0
93	-5	5	-10	45	-5	0	-5	25
94	-10	-5	0	10	-10	5	-10	10
95	-10	-5	0	5	-10	-5	-5	5
96	5	0	5	0	5	0	5	15
97	-10	-10	5	40	-10	-5	-10	5
98	10	10	15	55	0	0	5	75
99	-5	5	5	20	-5	5	5	40
100	-5	-5	-10	-10	-5	0	15	10

14.2 Principal component analysis

Principal component analysis is one of the oldest but still most widely used techniques of multivariate analysis. Originally introduced by Pearson (1901) and independently by Hotelling (1933), the basic idea of the method is to try to describe the variation of the variables in a set of multivariate data as parsimoniously as possible using a set of derived uncorrelated variables, each of which is a particular linear combination of those in the original data. In other words, principal component analysis is a transformation from the observed variables, y_{1i}, \dots, y_{pi} to new variables z_{1i}, \dots, z_{pi} where

$$\begin{aligned}
 z_{1i} &= a_{11}y_{1i} + a_{12}y_{2i} + \dots + a_{1p}y_{pi} \\
 z_{2i} &= a_{21}y_{1i} + a_{22}y_{2i} + \dots + a_{2p}y_{pi} \\
 &\vdots = \vdots + \vdots + \vdots + \vdots \\
 z_{pi} &= a_{p1}y_{1i} + a_{p2}y_{2i} + \dots + a_{pp}y_{pi}.
 \end{aligned} \tag{14.1}$$

The new variables are derived in decreasing order of importance. The coefficients a_{11} to a_{1p} for the first principal component are derived so that the sample variance of y_{1i} is as large as possible. Since this variance could be increased indefinitely by simply increasing the co-

efficients, a restriction must be placed on them, generally that their sum of squares is one. The coefficients defining the second principal component y_{2i} are determined to maximize its sample variance subject to the constraint that the sum of squared coefficients equals 1 and that the sample correlation between y_{1i} and y_{2i} is 0. The other principal components are defined similarly by requiring that they are uncorrelated with all previous principal components. It can be shown that the required coefficients are given by the eigenvectors of the sample covariance matrix of y_{1i}, \dots, y_{pi} , and their variances are given by the corresponding eigenvalues. In practice components are often derived from the correlation matrix instead of the covariance, particularly if the variables have very different scales. The analysis is then equivalent to calculation of the components from the original variables after these have been standardized to unit variance.

The usual objective of this type of analysis is to assess whether the first few components account for a substantial proportion of the variation in the data. If they do, they can be used to summarize the data with little loss of information. This may be useful for obtaining graphical displays of the multivariate data or for simplifying subsequent analysis. The principal components can be interpreted by inspecting the eigenvectors defining them. Here it is often useful to multiply the elements by the square root of the corresponding eigenvalue in which case the coefficients represent correlations between an observed variable and a component. A detailed account of principal component analysis is given in Everitt and Dunn (2001).

14.3 Analysis using Stata

The data can be read in from an ASCII file *hear.dat* as follows:

```
infile id 1500 11000 12000 14000 r500 r1000 r2000 r4000 /*
      */ using hear.dat
summarize
```

(see Display 14.1).

Before undertaking a principal component analysis, some graphical exploration of the data may be useful. A scatterplot matrix, for example, with points labeled with a subject's identification number can be obtained using

```
graph matrix 1500-r4000, mlabel(id) msymbol(none) /*
      */ mlabposition(0) half
```

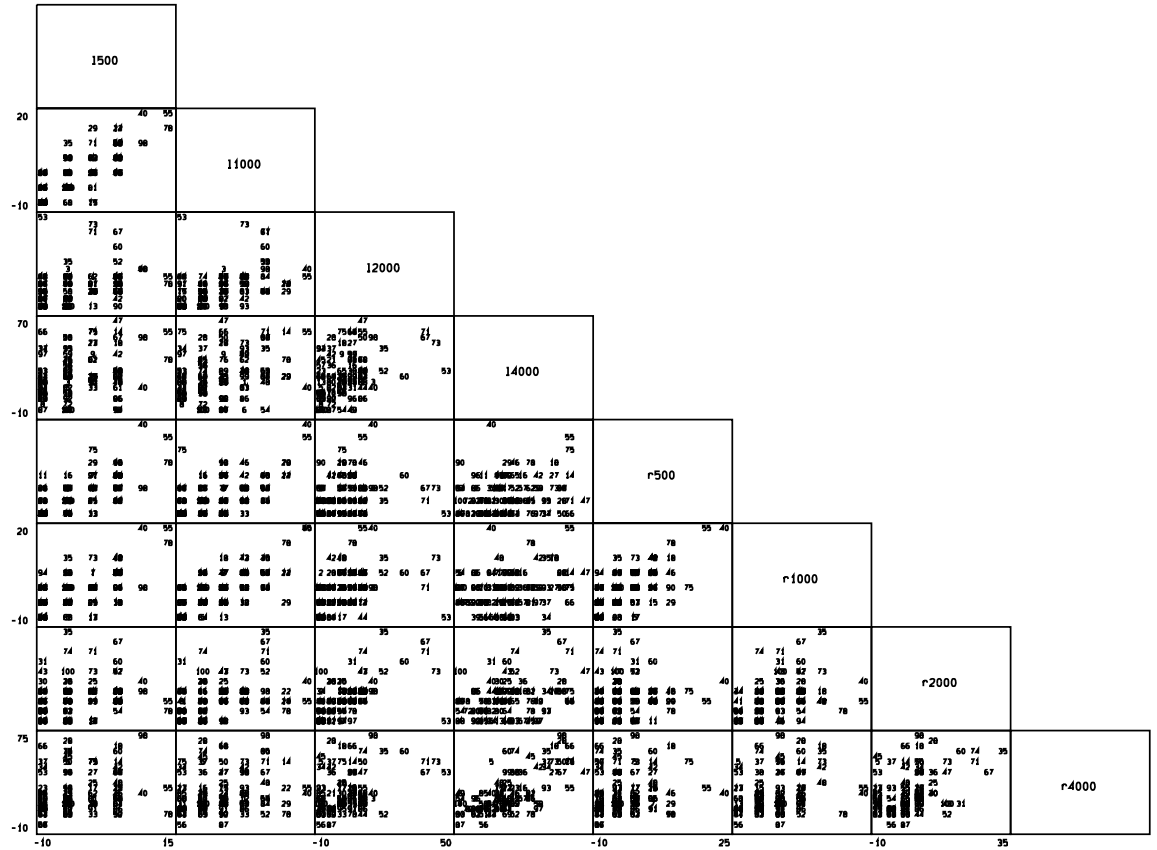


Figure 14.1: Scatterplot matrix of hearing loss at different frequencies for the left and right ear.

Variable	Obs	Mean	Std. Dev.	Min	Max
id	100	50.5	29.01149	1	100
1500	100	-2.8	6.408643	-10	15
11000	100	-.5	7.571211	-10	20
12000	100	2.45	11.94463	-10	50
14000	100	21.35	19.61569	-10	70
r500	100	-2.6	7.123726	-10	25
r1000	100	-.7	6.396811	-10	20
r2000	100	1.55	9.257675	-10	35
r4000	100	20.95	19.43254	-10	75

Display 14.1

The resulting diagram is shown in [Figure 14.1](#). The diagram looks a little ‘odd’ due to the largely ‘discrete’ nature of the observations. Some of the individual scatterplots suggest that some of the observations might perhaps be regarded as outliers; for example, individual 53 in the plot involving 12000, r2000. This subject has a score of 50 at this frequency in the left ear, but a score of -10 in the right ear. It might be appropriate to remove this subject’s observations before further analysis, but we shall not do this and will continue to use the data from all 100 individuals.

As mentioned in the previous section, principal components may be extracted from either the covariance matrix or the correlation matrix of the original variables. A choice needs to be made since there is not necessarily any simple relationship between the results in each case. The summary table shows that the variances of the observations at the highest frequencies are approximately nine times those at the lower frequencies; consequently, a principal component analysis using the covariance matrix would be dominated by the 4000 Hz frequency. But this frequency is not more clinically important than the others, and so, in this case, it seems more reasonable to use the correlation matrix as the basis of the principal component analysis.

To find the correlation matrix of the data requires the following instruction:

```
correlate 1500-r4000
```

and the result is given in Display 14.2. Note that the highest correlations occur between adjacent frequencies on the same ear and between corresponding frequencies on different ears.

The `pca` command can be used to obtain the principal components of this correlation matrix:

(obs=100)									
		1500	11000	12000	14000	r500	r1000	r2000	r4000
		<hr/>							
1500		1.0000							
11000		0.7775	1.0000						
12000		0.3247	0.4437	1.0000					
14000		0.2554	0.2749	0.3964	1.0000				
r500		0.6963	0.5515	0.1795	0.1790	1.0000			
r1000		0.6416	0.7070	0.3532	0.2632	0.6634	1.0000		
r2000		0.2399	0.3606	0.5910	0.3193	0.1575	0.4151	1.0000	
r4000		0.2264	0.2109	0.3598	0.6783	0.1421	0.2248	0.4044	1.0000

Display 14.2

`pca 1500-r4000`

which gives the results shown in Display 14.3.

An informal rule for choosing the number of components to represent a set of correlations is to use only those components with eigenvalues greater than one, i.e., those with variances greater than the average. Here, this leads to retaining only the first two components. Another informal indicator of the appropriate number of components is the *scree plot*, a plot of the eigenvalues against their rank. A scree plot may be obtained using

`greigen`

with the result shown in [Figure 14.2](#). The number of eigenvalues above a distinct ‘elbow’ in the scree plot is usually taken as the number of principal components to select. From Figure 14.2, this would again appear to be two. The first two components account for 68% of the variance in the data.

Examining the eigenvectors defining the first two principal components, we see that the first accounting for 48% of the variance has coefficients that are all positive and all approximately the same size. This principal component essentially represents the overall hearing loss of a subject and implies that individuals suffering hearing loss at certain frequencies will be more likely to suffer this loss at other frequencies as well. The second component, accounting for 20% of the variance, contrasts high frequencies (2000 Hz and 4000 Hz) and low frequencies (500 Hz and 1000 Hz). It is well known in the case of normal hearing that hearing loss as a function of age is first noticeable in the higher frequencies.

Scores for each individual on the first two principal components might be used as a convenient way of summarizing the original eight-

(obs=100)

(principal components; 8 components retained)

Component	Eigenvalue	Difference	Proportion	Cumulative
1	3.82375	2.18915	0.4780	0.4780
2	1.63459	0.72555	0.2043	0.6823
3	0.90904	0.40953	0.1136	0.7959
4	0.49951	0.12208	0.0624	0.8584
5	0.37743	0.03833	0.0472	0.9055
6	0.33910	0.07809	0.0424	0.9479
7	0.26101	0.10545	0.0326	0.9806
8	0.15556	.	0.0194	1.0000

Eigenvectors

Variable	1	2	3	4	5	6
1500	0.40915	-0.31257	0.13593	-0.27217	-0.16650	0.41679
11000	0.42415	-0.23011	-0.09332	-0.35284	-0.49977	-0.08474
12000	0.32707	0.30065	-0.47772	-0.48723	0.50331	0.04038
14000	0.28495	0.44875	0.47110	-0.17955	0.09901	-0.51286
r500	0.35112	-0.38744	0.23944	0.30453	0.62830	0.17764
r1000	0.41602	-0.23673	-0.05684	0.36453	-0.08611	-0.54457
r2000	0.30896	0.32280	-0.53841	0.51686	-0.16229	0.12553
r4000	0.26964	0.49723	0.41499	0.19757	-0.17570	0.45889

Eigenvectors

Variable	7	8
1500	0.28281	-0.60077
11000	-0.02919	0.61330
12000	-0.27925	-0.06396
14000	0.43536	-0.02978
r500	0.12745	0.36603
r1000	-0.46180	-0.34285
r2000	0.44761	0.02927
r4000	-0.47094	0.07469

Display 14.3

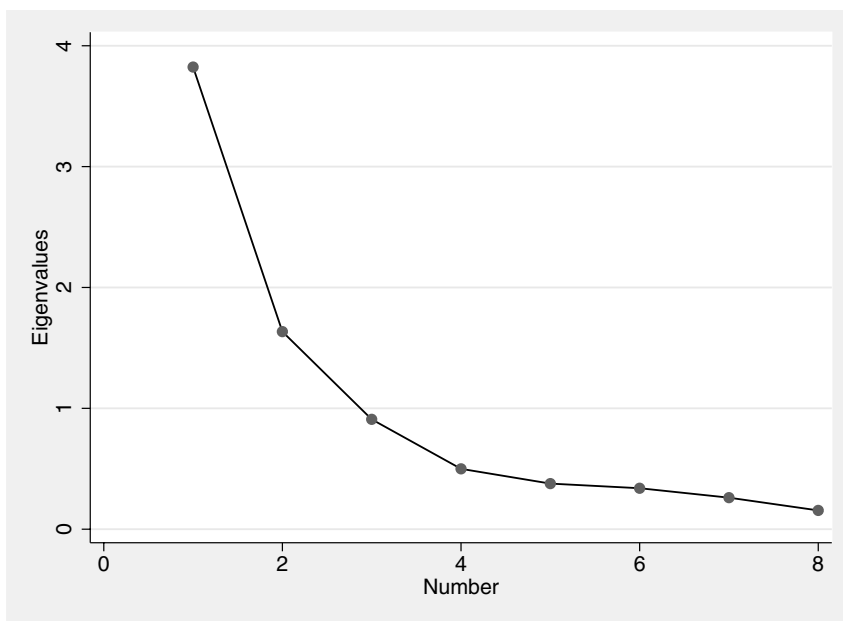


Figure 14.2: Scree plot.

dimensional data. Such scores are obtained by applying the elements of the corresponding eigenvector to the standardized values of the original observations for an individual. The necessary calculations can be carried out with the `score` procedure:

```
score pc1 pc2
```

(see Display 14.4).

The new variables `pc1` and `pc2` contain the scores for the first two principal components, and the output lists the coefficients used to form these scores. For principal component analysis, these coefficients are just the elements of the eigenvectors in Display 14.3. The principal component scores can be used to produce a useful graphical display of the data in a single scatterplot, which may then be used to search for structure or patterns in the data, particularly the presence of clusters of observations (see Everitt *et al.*, 2001). Note that the distances between observations in this graph approximate the Euclidean distances between the (standardized) variables, i.e., the graph is a *multidimensional scaling* solution. In fact, the graph is the classical scaling (or principal coordinate) scaling solution to the Euclidean distances (see Everitt and Dunn, 2001, or Everitt and Rabe-Hesketh, 1997).

The principal component plot is obtained using

(based on unrotated principal components)		
(6 scorings not used)		
	Scoring Coefficients	
Variable	1	2
1500	0.40915	-0.31257
11000	0.42415	-0.23011
12000	0.32707	0.30065
14000	0.28495	0.44875
r500	0.35112	-0.38744
r1000	0.41602	-0.23673
r2000	0.30896	0.32280
r4000	0.26964	0.49723

Display 14.4

```
twoway scatter pc2 pc1, mlabel(id)
```

The resulting diagram is shown in [Figure 14.3](#). Here, the variability in differential hearing loss for high versus low frequencies (**pc2**) is greater among subjects with higher overall hearing loss, as would be expected. It would be interesting to investigate the relationship between the principal components and other variables related to hearing loss such as age (see Exercise 6).

14.4 Exercises

1. Rerun the principal component analysis described in this chapter using the covariance matrix of the observations. Compare the results with those based on the correlation matrix.
2. Interpret components 3 through 8 in the principal components analysis based on the correlation matrix.
3. Create a scatterplot matrix of the first five principal component scores.
4. Investigate other methods of factor analysis available in **factor** applied to the hearing data.
5. Apply principal component analysis to the air pollution data analyzed in [Chapter 3](#), excluding the variable **so2**, and plot the first two principal components (i.e., the two-dimensional classical scaling solution for Euclidean distances between standardized variables).
6. Regress **so2** on the first two principal components and add a line corresponding to this regression (the direction of steepest increase

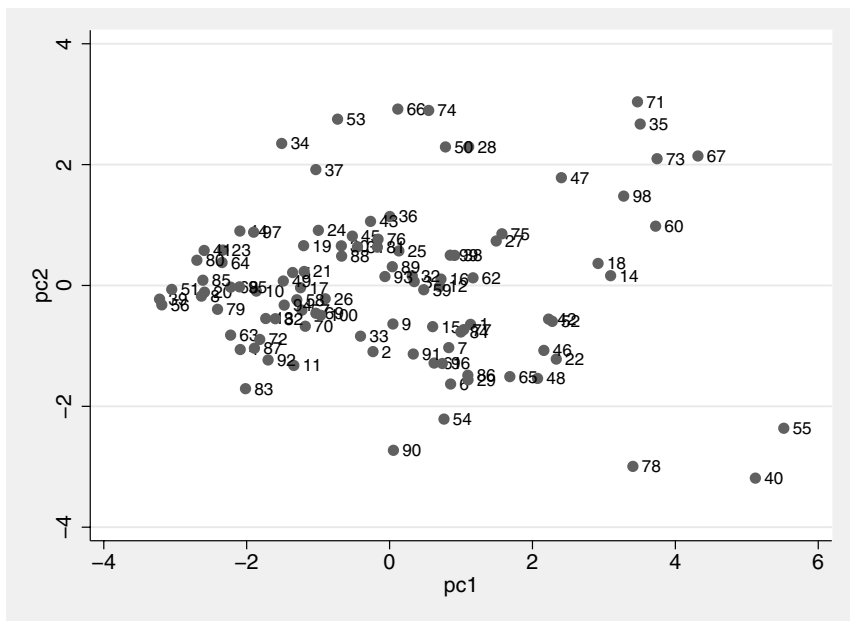


Figure 14.3: Principal component plot.

in `so2` predicted by the regression plane) into the multidimensional scaling solution.

Chapter 15

Cluster Analysis: Tibetan Skulls and Air Pollution in the USA

15.1 Description of data

The first set of data to be used in this chapter is shown in [Table 15.1](#). These data, collected by Colonel L.A. Waddell, were first reported in Morant (1923) and are also given in Hand *et al.* (1994). The data consists of five measurements on each of 32 skulls found in the southwestern and eastern districts of Tibet. The five measurements (all in millimeters) are as follows:

- y1: greatest length of skull
- y2: greatest horizontal breadth of skull
- y3: height of skull
- y4: upper face length
- y5: face breadth, between outermost points of cheek bones

The main question of interest about these data is whether there is any evidence of different types or classes of skull.

The second set of data that we shall analyze in this chapter is the air pollution data introduced previously in Chapter 3 (see [Table 3.1](#)). Here we shall investigate whether the clusters of cities found are predictive of air pollution.

Table 15.1 Tibetan skull data

y1	y2	y3	y4	y5
190.5	152.5	145.0	73.5	136.5
172.5	132.0	125.5	63.0	121.0
167.0	130.0	125.5	69.5	119.5
169.5	150.5	133.5	64.5	128.0
175.0	138.5	126.0	77.5	135.5
177.5	142.5	142.5	71.5	131.0
179.5	142.5	127.5	70.5	134.5
179.5	138.0	133.5	73.5	132.5
173.5	135.5	130.5	70.0	133.5
162.5	139.0	131.0	62.0	126.0
178.5	135.0	136.0	71.0	124.0
171.5	148.5	132.5	65.0	146.5
180.5	139.0	132.0	74.5	134.5
183.0	149.0	121.5	76.5	142.0
169.5	130.0	131.0	68.0	119.0
172.0	140.0	136.0	70.5	133.5
170.0	126.5	134.5	66.0	118.5
182.5	136.0	138.5	76.0	134.0
179.5	135.0	128.5	74.0	132.0
191.0	140.5	140.5	72.5	131.5
184.5	141.5	134.5	76.5	141.5
181.0	142.0	132.5	79.0	136.5
173.5	136.5	126.0	71.5	136.5
188.5	130.0	143.0	79.5	136.0
175.0	153.0	130.0	76.5	142.0
196.0	142.5	123.5	76.0	134.0
200.0	139.5	143.5	82.5	146.0
185.0	134.5	140.0	81.5	137.0
174.5	143.5	132.5	74.0	136.5
195.5	144.0	138.5	78.5	144.0
197.0	131.5	135.0	80.5	139.0
182.5	131.0	135.0	68.5	136.0

15.2 Cluster analysis

Cluster analysis is a generic term for a set of (largely) exploratory data analysis techniques that seek to uncover groups or clusters in data. The term exploratory is important since it explains the largely absent ‘*p*-value’, ubiquitous in many other areas of statistics. Clustering methods are primarily intended for generating rather than testing hypotheses. A detailed account of what is now a very large area is given in Everitt *et al.*, (2001).

The most commonly used class of clustering methods contains those methods that lead to a series of nested or hierarchical classifications of the observations, beginning at the stage where each observation is regarded as forming a single-member ‘cluster’ and ending at the stage where all the observations are in a single group. The complete hierarchy of solutions can be displayed as a tree diagram known as a dendrogram. In practice, most users will be interested not in the whole dendrogram, but in selecting a particular number of clusters that is optimal in some sense for the data. This entails ‘cutting’ the dendrogram at some particular level.

Most hierarchical methods operate not on the raw data, but on an inter-individual distance matrix calculated from the raw data. The most commonly used distance measure is Euclidean and is defined as:

$$d_{ij} = \sqrt{(y_{1i} - y_{1j})^2 + (y_{2i} - y_{2j})^2 + \cdots + (y_{pi} - y_{pj})^2}, \quad (15.1)$$

where y_{1i} to y_{pi} are the variables for individual i .

A variety of hierarchical clustering techniques arise because of the different ways in which the distance between a cluster containing several observations and a single observation, or between two clusters, can be defined. The inter-cluster distances used by three commonly applied hierarchical clustering techniques are:

- Single linkage clustering: distance between the closest pair of observations, where one member of the pair is in the first cluster and the other in the second cluster, and
- Complete linkage clustering: distance between the most remote pair of observations where one member of the pair is in the first cluster and the other in the second cluster.
- Average linkage: average of distances between all pairs of observations where one member of the pair is in the first cluster and the other in the second cluster.

An alternative approach to clustering to that provided by the hierarchical methods described above is *k*-means clustering. Here the data are partitioned into a specified number of groups set by the user

by an iterative process in which, starting from an initial set of cluster means, each observation is placed into the group to whose mean vector it is closest (generally in the Euclidean sense). After each iteration, new group means are calculated and the procedure repeated until no observations change groups. The initial group means can be chosen in a variety of ways. In general, the method is applied to the data for different numbers of groups and then an attempt is made to select the number of groups that provides the best fit for the data.

Important issues that need to be considered when using clustering in practice include how to scale the variables before calculating the chosen distance matrix, which particular method of cluster analysis to use, and how to decide on the appropriate number of groups in the data. These and many other practical problems of clustering are discussed in Everitt *et al.* (2001).

15.3 Analysis using Stata

15.3.1 *Tibetan skulls*

Assuming the data in [Table 15.1](#) are contained in a file `tibetan.dat`, they can be read into Stata using the instruction

```
infile y1 y2 y3 y4 y5 using tibetan.dat, clear  
gen id = _n
```

Here we have also generated an identifier variable `id` for the skulls. To begin it is good practice to examine some graphical displays of the data. With multivariate data such as the measurements on skulls in [Table 15.1](#) a scatterplot matrix is often helpful and can be generated as follows:

```
graph matrix y1-y5
```

The resulting plot is shown in [Figure 15.1](#). A few of the individual scatterplots in [Figure 15.1](#) are perhaps suggestive of a division of the observations into distinct groups, for example that for `y4` (upper face height) versus `y5` (face breadth).

We shall now apply each of single linkage, complete linkage, and average linkage clustering to the data using Euclidean distance as the basis of each analysis. Here the five measurements are all on the same scale, so that standardization before calculating the distance matrix is probably not needed (but see the analysis of the air pollution data described later). The necessary Stata commands are

```
cluster singlelinkage y1-y5, name(s1)  
cluster dendrogram
```

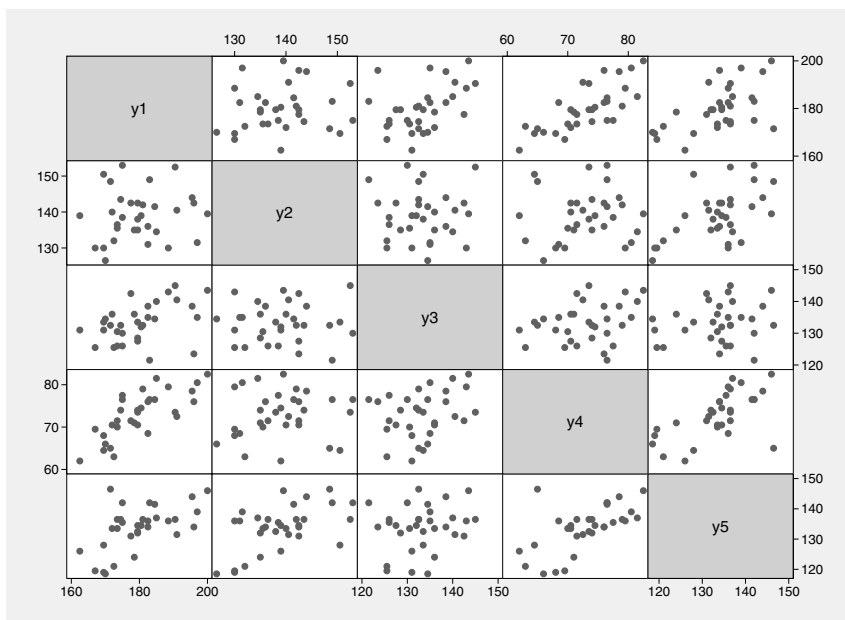



Figure 15.1: Scatterplot matrix of Tibetan skull data

```

cluster completelinkage y1-y5, name(c1)
cluster dendrogram
cluster averagelinkage y1-y5, name(a1)
cluster dendrogram

```

Here the `name()` option is used to attach a name to the results from each cluster analysis. The resulting three dendrograms are shown in Figures 15.2, 15.3, and 15.4.

The single linkage dendrogram illustrates one of the common problems with this technique, namely its tendency to incorporate observations into existing clusters rather than begin new ones, a property generally referred to as chaining (see Everitt *et al.*, 2001, for full details). The complete linkage and average linkage dendrograms show more evidence of cluster structure in the data, although this structure appears to be different for each method, a point we shall investigate later.

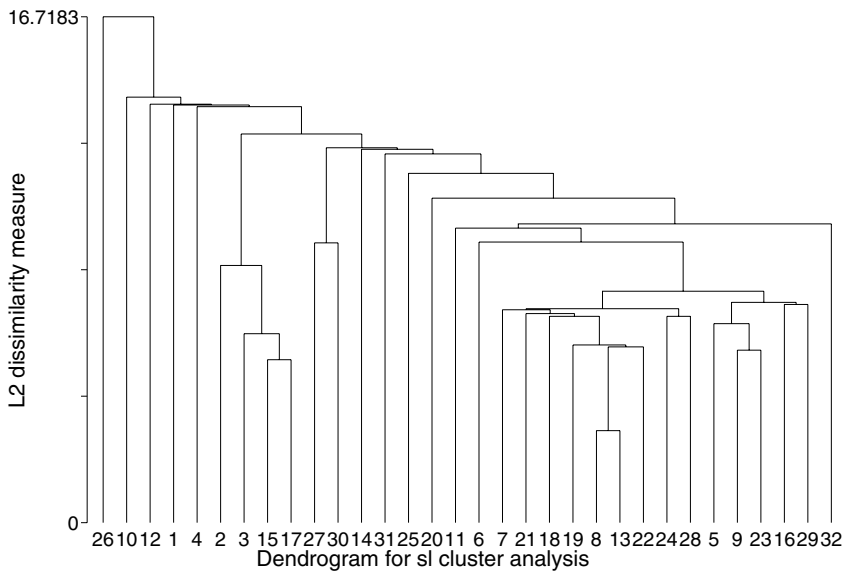


Figure 15.2: Dendrogram using single linkage.

In most applications of cluster analysis the researcher will try to determine the solution with the optimal number of groups, i.e., the number of groups that ‘best’ fits the data. Estimating the number of groups in a cluster analysis is a difficult problem without a completely

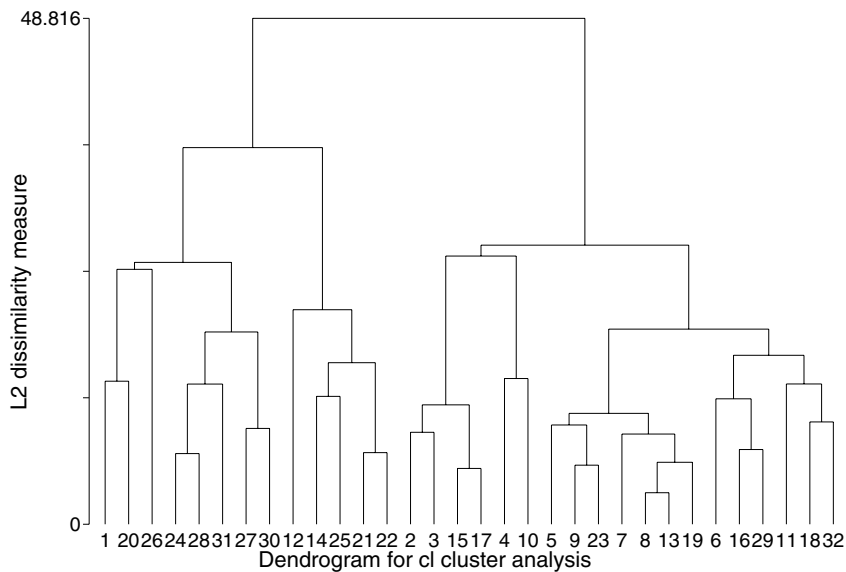


Figure 15.3: Dendrogram using complete linkage.

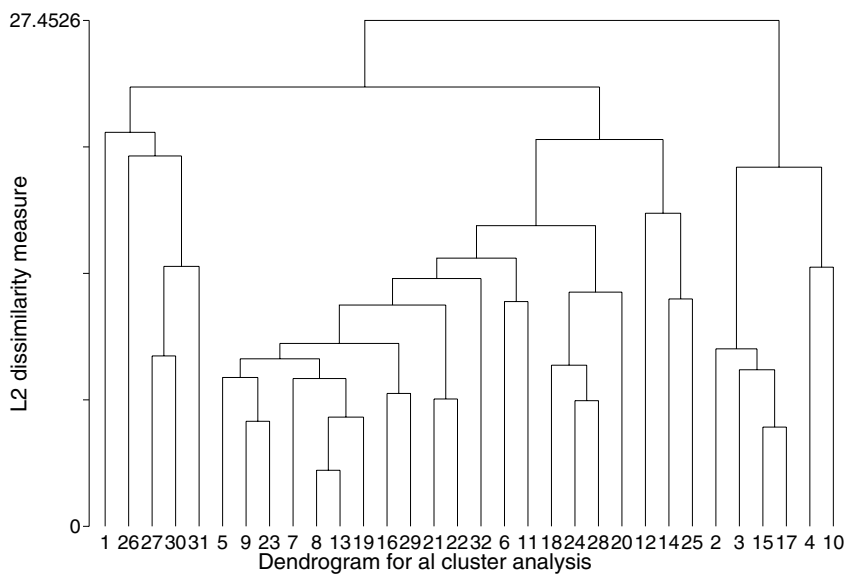


Figure 15.4: Dendrogram using average linkage.

satisfactory solution – see Everitt *et al.* (2001). Two stopping rules are provided in Stata, the Calinski and Harabasz pseudo F -statistic (Calinski and Harabasz, 1974) and the Duda and Hart index (Duda and Hart, 1973); see [CL] **cluster stop** for details. For both these rules, larger values indicate more distinct clustering.

Here we shall illustrate the use of the Duda and Hart index in association with the three clustering techniques applied above. The Stata commands are

```
cluster stop sl, rule(duda) groups(1/5)
```

(see Display 15.1),

Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.9512	1.54
2	0.9357	1.99
3	0.9430	1.69
4	0.9327	1.95
5	0.9380	1.72

Display 15.1

```
cluster stop cl, rule(duda) groups(1/5)
```

(see Display 15.2), and

Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.6685	14.88
2	0.6073	7.11
3	0.5603	13.34
4	0.3356	7.92
5	0.7006	2.56

Display 15.2

`cluster stop al, rule(duda) groups(1/5)`
 (see Display 15.3).

Number of clusters	Duda/Hart	
	Je(2)/Je(1)	pseudo T-squared
1	0.6722	14.63
2	0.7192	9.37
3	0.5959	2.03
4	0.7200	7.39
5	0.3731	3.36

Display 15.3

Distinct clustering is generally considered to be indicated by large values of the Duda and Hart index and small values of the Duda and Hart pseudo *T*-squared. Adopting this approach, the results from single linkage clustering do not suggest any distinct cluster structure largely because of the chaining phenomenon. The results associated with complete linkage clustering suggest a five-group solution and those from the average linkage method suggest perhaps three or four groups.

To see which skulls are placed in which groups we can use the `cluster gen` command. For example, to examine the five group solution given by complete linkage we use

```
cluster gen g5cl=groups(5), name(c1)
sort g5cl id
forvalues i=1/5 {
    disp " "
    disp "cluster `i'"
    list id if g5cl==`i', noobs noheader separator(0)
}
```

Here we use a `forvalues` loop to list `id` for each cluster. The `noobs` option suppresses line-numbers; the `noheader` option suppresses variable names; and `separator(0)` suppresses separator lines. The resulting output is shown in Display 15.4. The numbers of observations in each group can be tabulated using

```
tab g5cl
```

giving the table in Display 15.5.

cluster 1

1
20
24
26
27
28
30
31

cluster 2

12
14
21
22
25

cluster 3

2
3
15
17

cluster 4

4
10

cluster 5

5
6
7
8
9
11
13
16
18
19
23
29
32

Display 15.4

g5c1	Freq.	Percent	Cum.
1	8	25.00	25.00
2	5	15.63	40.63
3	4	12.50	53.13
4	2	6.25	59.38
5	13	40.63	100.00
Total	32	100.00	

Display 15.5

It is often helpful to compare the mean vectors of each of the clusters. The necessary code to find these is:

```
table g5c1, contents(mean y1 mean y2 mean y3 mean y4 /*
*/ mean y5) format(%4.1f)
```

g5c1	mean(y1)	mean(y2)	mean(y3)	mean(y4)	mean(y5)
1	192.9	139.4	138.6	78.1	138.0
2	179.0	146.8	130.2	74.7	141.7
3	169.8	129.6	129.1	66.6	119.5
4	166.0	144.8	132.3	63.3	127.0
5	177.6	137.9	132.7	72.5	133.4

Display 15.6

The skulls in cluster 1 are characterized by being relatively long and narrow. Those in cluster 2 are, on average, shorter and broader. Cluster 3 skulls appear to be particularly narrow, and those in cluster 4 have short upper face length. Skulls in cluster 5 might perhaps be considered ‘average’.

The scatterplot matrix of the data used earlier to allow a ‘look’ at the raw data is also useful for examining the results of clustering the data. For example, we can produce a scatterplot matrix with observations identified by their cluster number from the three group solution from average linkage:

```
cluster gen g3al=groups(3), name(c1)
graph matrix y1-y5, mlabel(g3al) mlabpos(0) msymbol(i)
```

(see Figure 15.5). The separation between the three groups is most distinct in the panel for greatest length of skull y1 versus face breadth y5.

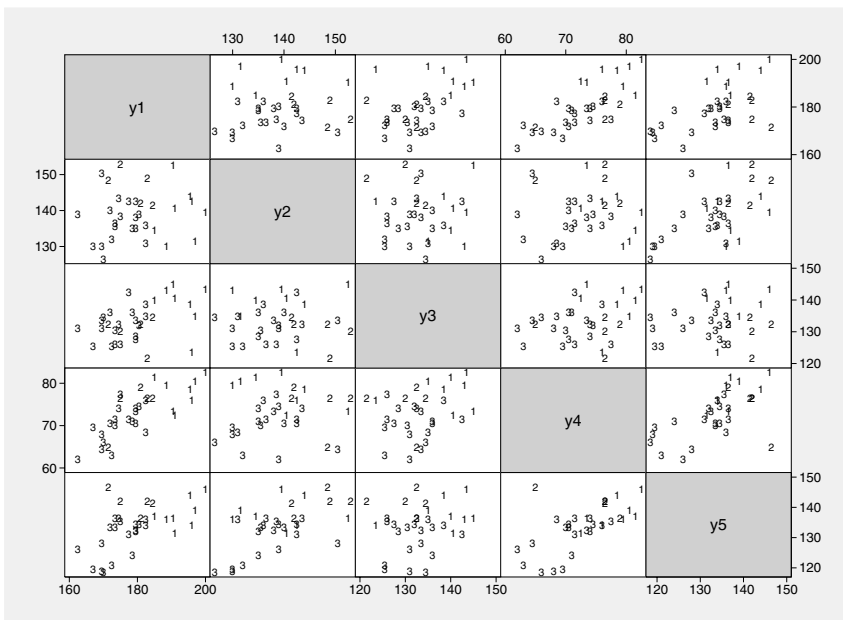


Figure 15.5: Scatterplot matrix with observations identified by their cluster number.

The data as originally collected by Colonel Waddell were thought to consist of two types of skulls; the first type, skulls 1-17, came from graves in Sikkim and the neighboring areas of Tibet. The remaining 15 skulls were picked up on a battlefield in the Lhasa district and were believed to be those of native soldiers from the eastern province of Khams. These skulls were of particular interest because it was thought at the time that Tibetans from Khams might be survivors of a particular fundamental human type, unrelated to the Mongolian and Indian types which surrounded them. We can compare this classification with the two group solutions given by each of the three clustering methods by cross-tabulating the corresponding categorical variables containing group membership. The Stata code for this is:

```
gen cl2 = cond(id<=17,1,2)
```



```

cluster gen g2s1=groups(2), name(s1)
cluster gen g2c1=groups(2), name(c1)
cluster gen g2a1=groups(2), name(a1)
tab c12 g2s1, row

```

(see Display 15.7),

Key			
<i>frequency</i>			
<i>row percentage</i>			

c12	g2s1		Total
	1	2	
1	0	17	17
	0.00	100.00	100.00
2	1	14	15
	6.67	93.33	100.00
Total	1	31	32
	3.13	96.88	100.00

Display 15.7

```

tab c12 g2c1, row

```

(see Display 15.8), and

```

tab c12 g2a1, row

```

(see Display 15.9). The two group solution from single linkage consists of 31 observations in one group and only a single observation in the second group, again illustrating the chaining problem associated with this method. The complete linkage solution provides the closest match to the division originally proposed for the skulls.

15.3.2 Air pollution

In this section we shall apply *k*-means clustering to the air pollution data from [Chapter 3](#). We will use the variables `temp`, `manuf`, `pop`, `wind`, `precip`, and `days` for the cluster analysis. Since these variables have very different metrics we shall begin by standardizing them

```

infile str10 town so2 temp manuf pop wind precip days /*
*/ using usair.dat, clear

```

Key
<i>frequency</i> <i>row percentage</i>

c12	g2c1		Total
	1	2	
1	3	14	17
	17.65	82.35	100.00
2	10	5	15
	66.67	33.33	100.00
Total	13	19	32
	40.63	59.38	100.00

Display 15.8

Key
<i>frequency</i> <i>row percentage</i>

c12	g2a1		Total
	1	2	
1	11	6	17
	64.71	35.29	100.00
2	15	0	15
	100.00	0.00	100.00
Total	26	6	32
	81.25	18.75	100.00

Display 15.9

```

foreach var of varlist temp manuf pop wind precip days {
    egen s`var' = std(`var')
}

```

We will now use the k-means algorithm to divide the data into 2, 3, 4, and 5 groups using the default for choosing initial cluster centers, namely the random selection of k unique observations from among those to be clustered. The necessary code is

```

cluster kmeans stemp smanuf spop swind sprecip sdays, k(2) /*
    */ name(cluster2)
cluster kmeans stemp smanuf spop swind sprecip sdays, k(3) /*
    */ name(cluster3)
cluster kmeans stemp smanuf spop swind sprecip sdays, k(4) /*
    */ name(cluster4)
cluster kmeans stemp smanuf spop swind sprecip sdays, k(5) /*
    */ name(cluster5)

```

We can now use the Calinski and Harabasz approach to selecting the optimal number of groups:

```
cluster stop cluster2
```

Number of clusters	Calinski/ Harabasz pseudo-F
2	6.44

```
cluster stop cluster3
```

Number of clusters	Calinski/ Harabasz pseudo-F
3	9.04

```
cluster stop cluster4
```

Number of clusters	Calinski/ Harabasz pseudo-F
4	12.38

```
cluster stop cluster5
```

Number of clusters	Calinski/ Harabasz pseudo-F
5	16.34

The largest value of the Calinski and Harabasz index corresponds to the five group solution. Details of this solution can be found from

```
sort cluster5 town
forvalues i=1/5 {
    disp " "
    disp "cluster 'i'"
    list town if cluster5=='i', noobs noheader separator(0)
}
```

The output is shown in Display 15.10. We will use the `tabstat` command to tabulate the means, as the `table` command can only tabulate up to five statistics:

```
tabstat temp manuf pop wind precip days, by(cluster5) /*
    */ nototal format(%4.1f)
```

(see Display 15.11).

We will now compare pollution levels (the annual mean concentration of sulphur dioxide `so2`) between these five clusters of towns. The means and standard deviations can be tabulated using

```
table cluster5, contents(mean so2 sd so2) format(%4.1f)
```

(see Display 15.12). Clusters 4 and 5 appear to have extremely high pollution levels and cluster 3 a relatively low level. A more formal analysis of differences in pollution levels among the clusters can be undertaken using a one-way analysis of variance. (Note that the variable `so2` did not contribute to the cluster analysis. If it had, it would be circular and invalid to carry out the analysis of variance.) We will first log-transform `so2` since the standard deviations appear to increase with the mean.

```
gen lso2 = ln(so2)
anova so2 cluster5
```

(see Display 15.13). The analysis shows that the clusters differ significantly in their average pollution levels, $F_{4,36} = 5.16$, $p = 0.002$.

cluster 1

Alburq
Denver
Phoenix
SLC
Sfran

cluster 2

Albany
Baltim
Charlest
Cincinn
Colum
Hartford
Indian
Louisv
Nashville
Pittsb
Richmond
Seattle
StLouis
Washing
Wilming

cluster 3

Atlanta
Dallas
Houston
Jackson
Lock
Memphis
Miami
NewO
Norfolk

cluster 4

Buffalo
Cleve
DesM
Kansas
Milwak
Minn
Omaha
Provid
Wichita

cluster 5

Chicago
Detroit
Philad

Summary statistics: mean

by categories of: cluster5

cluster5	temp	manuf	pop	wind	precip	days
1	57.3	260.6	446.6	8.3	12.7	67.2
2	57.4	346.5	501.8	10.1	39.3	101.9
3	70.3	317.0	614.5	9.3	54.8	115.0
4	48.7	1252.3	1358.1	10.8	33.1	135.3
5	52.9	268.6	364.2	8.6	40.2	131.1

Display 15.11

cluster5	mean(so2)	sd(so2)
1	15.6	7.4
2	22.7	15.5
3	10.8	2.2
4	47.9	35.3
5	38.8	20.9

Display 15.12

	Number of obs =	41	R-squared =	0.3646	
	Root MSE =	19.7221	Adj R-squared =	0.2940	
Source	Partial SS	df	MS	F	Prob > F
Model	8035.24113	4	2008.81028	5.16	0.0022
cluster5	8035.24113	4	2008.81028	5.16	0.0022
Residual	14002.6613	36	388.962814		
Total	22037.9024	40	550.947561		

Display 15.13

15.4 Exercises

1. Repeat the analyses of the Tibetan skull data described in the chapter using the Manhattan distance measure rather than the Euclidean. Compare the two sets of results.
2. Investigate the use of other options for determining an initial partition when applying k -means to the air pollution data.
3. Compare the results from k -medians cluster analysis applied to the air pollution data with those from k -means.

Chapter

Appendix: Answers to Selected Exercises

Chapter 1

2. Assuming that the data are stored in the directory `c:\user`,

```
cd c:\user
insheet using test.dat, clear
```

4. `label define s 1 male 2 female`
`label values sex s`

5. `gen id = _n`

6. `rename v1 time1`
`rename v2 time2`
`rename v3 time3`

or

```
forval i = 1/3 {
    rename var`i' time`i'
}
```

7. `reshape long time, i(id) j(occ)`

8. `egen d = mean(time), by(id)`
`replace d = (time-d)^2`

9. `drop if occ==3&id==2`

Chapter 2

1. `table depress, contents(mean weight)`
2.

```
foreach var in iq age weight {  
    table life, contents(mean `var' sd `var')  
}
```
3. `search mann`
`help signrank`
4. `ranksum weight, by(life)`
5.

```
twoway (scatter iq age if life==1, msymbol(circle) /*  
      */ mcolor(black) jitter(2)) /*  
      */ (scatter iq age if life==2, msymbol(x) /*  
      */ mcolor(black) jitter(2)), /*  
      */ legend(order(1 "no" 2 "yes"))
```
6. Save the commands in the Review window and edit the file using the Do-file Editor or any text editor, e.g., Notepad. Add the commands given in the do-file template in Section 1.10, and save the file with the extension `.do`. Run the file by typing the command `do filename`.

Chapter 4

1. `infile bp11 bp12 bp13 bp01 bp02 bp03 using bp.raw`
Now follow the commands on pages 80 to 81. There is no need to redefine labels, but if you wish to do so, first issue the command `label drop _all`.
2.

```
graph box bp, over(drug)  
graph box bp, over(diet)  
graph box bp, over(biofeed)
```
4.

```
sort id  
save bp  
infile id age using age.dat, clear  
sort id  
merge id using bp  
anova bp drug diet biofeed age, continuous(age)
```

Chapter 5

1. `anova resp cond*status status cond, sequential`

2. `gen dcond = cond - 1`
`gen dstat = status - 1`
`gen dinter = dcond*dstat`
`regress resp dcond dstat dinter`
3. `xi: regress resp i.cond*i.status`
4. `char cond[omit] 2`
`char status[omit] 2`
`xi: regress resp i.cond*i.status`

Chapter 6

1. `ologit outc therapy sex [fweight=fr], table`
2. a. `ologit depress life`
b. `replace life = life - 1`
`logistic life depress`
3. Even if we use very lenient inclusion and exclusion criteria,
`sw logistic life depress anxiety iq sex sleep, /*`
`*/ pr(0.2) pe(0.1) forward`
only `depress` is selected. If we exclude `depress` from the list of candidate variables, `anxiety` and `sleep` are selected.
4. `bprobit pres tot ck`
`predict predp`
`twoway (line predp ck) (scatter prop ck), /*`
`*/ ytitle("Probability")`

`twoway (function y=norm(_b[_cons]+_b[ck]*x), /*`
`*/ range(0 480)) (scatter prop ck), /*`
`*/ ytitle("Probability")`

Chapter 7

1. `xi: glm resp i.cond i.status, fam(gauss) link(id)`
`local dev1 = e(deviance)`
`xi: glm resp i.cond, fam(gauss) link(id)`
`local dev2 = e(deviance)`
`local ddev = `dev2' - `dev1'`
`/* F-test equivalent to anova cond status, sequential */`
`local f = (`ddev'/1)/(`dev1'/31)`

- ```

disp `f`
disp fprob(1,31,`f`)
/* difference in deviance */
disp `ddev`
disp chiprob(1, `ddev`)
2. reg resp status, robust
 ttest resp, by(status) unequal
3. gen cleth = class*ethnic
 glm days class ethnic cleth, fam(poisson) link(log)
4. glm days class ethnic if stres<4, fam(poisson) link(log)
 or, assuming the sort order of the data has not changed,
 glm days class ethnic if _n!=72, fam(poisson) link(log)
5. gen abs = cond(days>=14,1,0)
 glm abs class ethnic, fam(binomial) link(logit)
 glm abs class ethnic, fam(binomial) link(probit)
6. logit abs class ethnic, robust
 probit abs class ethnic, robust
 bs "logit abs class ethnic" "_b[class] _b[ethnic]", /*
 */ reps(500)
 bs "probit abs class ethnic" "_b[class] _b[ethnic]", /*
 */ reps(500)

```

## Chapter 8

1. graph box dep1-dep6, by(group)
2. We can obtain the mean over visits for subjects with complete data using the simple command (data in 'wide' form)
 

```
gen av2 = (dep1+dep2+dep3+dep4+dep5+dep6)/6
```

For subjects with missing data, av2 will be missing whereas the `rmean()` `egen` function would return the mean of all available data. The *t*-tests are obtained using

```
ttest av2, by(group)
ttest av2, by(group) unequal
```
3. `egen max = rmax(dep1-dep6)`

```
ttest max, by(group)
```
4. a. `gen diff = avg-pre`

```
ttest diff, by(group)
```

 b. `anova avg group pre, continuous(pre)`

## Chapter 9

2. 

```
use madras, clear
reshape long y, i(id) j(month)
label variable month /*
 */ "Number of months since hospitalization"
gen month_early = month*early
label define e 0 "Late onset" 1 "Early onset"
label values early e

gllamm y month early month_early, i(id) /*
 */ link(logit) family(binom) eform adapt
gllapred prob1, mu
sort id month
twoway (line prob1 month, connect(ascending)), /*
 */ by(early) ytitle(Predicted probability)

gen cons = 1
eq slope: month
eq inter: cons
gllamm y month early month_early, i(id) nrf(2) /*
 / eqs(inter slope) link(logit) family(binom) /
 */ eform adapt
gllapred prob2, mu
sort id month
twoway (line prob2 month, connect(ascending)), /*
 */ by(early) ytitle(Predicted probability)
```
3. 

```
use quine, clear
encode eth, gen(ethnic)
drop eth
encode sex, gen(gender)
drop sex
encode age, gen(class)
drop age
encode lrn, gen(slow)
drop lrn

gen id=_n
gllamm days class ethnic, i(id) adapt family(pois)
```
4. 

```
use epil, clear
gen y0 = baseline
reshape long y, i(subj) j(visit)
```

```

gen post = visit>0
gen tr_post = treat*post
gen lnoobs = cond(visit==0,ln(8),ln(2))
gllamm y age treat post tr_post, i(subj) /*
 */ family(pois) offset(lnoobs) adapt

```

## Chapter 10

2. a. 

```
infile subj group dep0 dep1 dep2 dep3 dep4 dep5 dep6 /*
 */ using depress.dat
mvdecode _all, mv(-9)
reshape long dep, i(subj) j(visit)
gen gr_vis = group*visit
reg dep group visit gr_vis, robust cluster(subj)
```

b. 

```
bs "reg dep group visit gr_vis" /*
 / "_b[group] _b[visit] _b[gr_vis]", /
 */ cluster(subj) reps(500)
```
4. 

```
use madras, clear
reshape long y, i(id) j(month)
label variable month "Number of months since hospitalization"
gen month_early = month*early

xtgee y month early month_early, i(id) link(logit) /*
 */ family(binom) corr(exch)

twoway (function y=1/(1+exp(-_b[_cons]-_b[month]*x /*
 */ -_b[early] - _b[month_early]*x)), range(0 10)), /*
 */ ytitle(Predicted probability) xtitle(Month)
```

## Chapter 11

1. 

```
infile v1-v2 using estrogen.dat, clear
gen str8 = cond(_n==1,"ncases1","ncases0")
xpose,clear
gen conestr = 2-_n
reshape long ncases, i(conestr) j(casestr)
expand ncases
sort casestr conestr
gen caseid = _n
expand 2
bysort caseid: gen control = _n-1 /* 1=cont., 0=case */
```

```

gen estr = 0
replace estr = 1 if control==0&casestr==1
replace estr = 1 if control==1&conestr>0
gen cancer = cond(control==0,1,0)

preserve
collapse (sum) estr (mean) casestr , by(caseid)
gen conestr = estr - casestr
tab casestr conestr
restore

clogit cancer estr, group(caseid) or
2. infile str5 age num1 py1 num0 py0 using ihd.dat,clear
gen agegr = _n
reshape long num py, i(agegr) j(exposed)

table exposed, contents(sum num sum py)
iri 28 17 1857.5 2768.9
3. Keeping the data from the previous exercise:
xi: poisson num i.age*exposed, e(py) irr
testparm _laxeX*

The interaction is not statistically significant at the 5% level.
4. infile subj y1 y2 y3 y4 treat baseline age /*
 */ using chemo.dat
reshape long y, i(subj) j(week)
expand 2 if week==1
sort subj week
qui by subj: replace week = 0 if _n==1
replace y = baseline if week==0
gen post = week!=1
gen ltime = log(cond(week==0,8,2))
xi: xtgee y i.treat*i.post age , i(subj) t(week) /*
 */ corr(exc) family(pois) scale(x2) offset(ltime)

```

## Chapter 12

1. We consider anyone still at risk after 450 days as being censored at 450 days and therefore need to make the appropriate changes to status and time before running Cox regression.

```
use heroin, clear
```

```

replace status = 0 if time>450
replace time = 450 if time>450
egen zdose = std(dose)
stset time status
stcox zdose prison clinic

```

2. The model is fitted using

```

gen dosecat = 0 if dose<.
replace dosecat = 1 if dose>=60 & dose<.
replace dosecat = 2 if dose>=80 & dose<.
xi: stcox i.dosecat i.prison i.clinic, bases(s)

```

The survival curves for no prison record, clinic 1 are obtained and plotted using

```

gen s0 = s if dosecat==0
gen s1 = s^(exp(_b[_Idosecat_1])) if dosecat==1
gen s2 = s^(exp(_b[_Idosecat_2])) if dosecat==2
sort time
graph twoway (line s0 time, connect(stairstep)) /*
 / (line s1 time, connect(stairstep) clpat(dash)) /
 / (line s2 time, connect(stairstep) clpat(dot)), /
 */ legend(order(1 "<60" 2 "60-79" 3 ">=80"))

```

Note that the baseline survival curve is the survival curve for someone whose covariates are all zero. If we had used `clinic` instead of `i.clinic` above, this would have been meaningless; we would have had to exponentiate `s0`, `s1`, and `s2` by `_b[clinic]` to calculate the survival curves for clinic 1.

3. Treating dose as continuous:

```

gen clindose = clinic*zdose
stcox zdose clinic clindose prison

```

Treating dose as categorical:

```

xi: stcox i.dosecat*i.clinic i.prison
testparm _IdosX*

```

4. xi: stcox i.dosecat i.prison i.clinic  
xi: stcox i.dosecat i.prison i.clinic, efron  
xi: stcox i.dosecat i.prison i.clinic, exactm

It makes almost no difference which method is used.

5. stset time, failure(status) id(id)  
stsplot, at(failures) strata(clinic)  
gen tpris = prison\*(t-504)/365.25  
stcox dose prison tpris, strata(clinic)

```
lrtest, saving(0)
stcox dose prison, strata(clinic)
lrtest
```

## Chapter 13

2. 

```
nlcom (sd1: exp([lsd1][_cons])) /*
 / (sd2: exp([lsd2][_cons])) /
 */ (p: invlogit([lo1][_cons]))
```

giving estimates (standard errors) 6.54 (0.82) and 7.06 (1.81) for the standard deviations and 0.74 (0.07) for the probability.

4. The only thing that is different from fitting a normal distribution with constant mean is that the mean is now a linear function of `status` so that the `ml model` command changes as shown below:

```
infile cond status resp using slim.dat, clear
ml model lf mixing1 (xb: resp = status) /lsd
ml maximize, noheader
```

In linear regression, the mean square error is equal to the sum of squares divided by the degrees of freedom,  $n - 2$ . The maximum likelihood estimate is equal to the sum of squares divided by  $n$ . We can therefore get the root mean square error for linear regression using

```
disp exp([lsd][_cons])*sqrt(e(N)/(e(N)-2))
```

Note that the standard error of the regression coefficients need to be corrected by the same factor, i.e.,

```
disp [xb]_se[status]*sqrt(e(N)/(e(N)-2))
```

Compare this with the result of

```
regress resp status
```

5. Repeat the procedure above but replace the `ml model` command by
- ```
ml model lf mixing1 (resp = status) (lsd: status)
```

The effect of `status` on the standard deviation is significant ($p = 0.003$) which is not too different from the result of

```
sdtest resp, by(status)
```

6.

```
capture program drop mixing3
program mixing3
    version 8.1
```



```

args lj xb1 xb2 xb3 lo1 lo2 ls1 ls2 ls3

tempvar f1 f2 f3 p1 p2 p3 s1 s2 s3 d

quietly {
    gen double `s1' = exp(`ls1')
    gen double `s2' = exp(`ls2')
    gen double `s3' = exp(`ls3')
    gen double `d' = 1 + exp(`lo1') + exp(`lo2')
    gen double `p1' = 1/`d'
    gen double `p2' = exp(`lo1')/`d'
    gen double `p3' = exp(`lo2')/`d'

    gen double `f1' = normden($ML_y1,`xb1',`s1')
    gen double `f2' = normden($ML_y1,`xb2',`s2')
    gen double `f2' = normden($ML_y1,`xb3',`s3')

    replace `lj' = ln(`p1'*`f1' /*
        */ + `p2'*`f2' + `p3'*`f3')
}
end

clear
set obs 300
set seed 12345678
gen z = uniform()
gen y = invnorm(uniform())
replace y = y + 5 if z<1/3
replace y = y + 10 if z<2/3&z>=1/3
ml model lf mixing3 (xb1: y=) /*
    */ /xb2 /xb3 /lo1 /lo2 /lsd1 /lsd2 /lsd3
ml init 0 5 10 0 0 0 0 0, copy
ml maximize, noheader trace

```

Chapter 14

1. `pca 1500-r4000, cov`
`score npc1 npc2`
`twoway scatter npc2 npc1, mlabel(id)`
3. `capture drop pc*`
`pca 1500-r4000`

- ```

score pc1-pc5
graph matrix pc1-pc5
5. infile str10 town so2 temp manuf pop wind precip days /*
 */ using usair.dat
 pca temp manuf pop wind precip days
 score pc1 pc2
 scatter pc2 pc1, mlabel(town) mlabpos(0) ms(i)
6. regress so2 pc1 pc2
 gen regline = pc1*_b[pc2]/_b[pc1]
 twoway (line regline pc1) /*
 */ (scatter pc2 pc1, mlabel(town) mlabpos(0) ms(i))

```

## Chapter 15

- infile y1 y2 y3 y4 y5 using tibetan.dat, clear  
cluster singlelinkage y1-y5, name(s1) manhattan  
cluster completelinkage y1-y5, name(c1) manhattan  
cluster averagelinkage y1-y5, name(a1) manhattan  
Then plot dendrograms, etc.
- infile str10 town so2 temp manuf pop wind precip /\*  
 \*/ days using usair.dat, clear  
foreach var of varlist temp manuf pop wind precip days {  
 egen s`var` = std(`var`)  
}  
cluster kmeans stemp smanuf spop swind sprecip sdays, /\*  
 \*/ k(5) name(cluster5) start(segments)  
The start(segments) option splits the sample into  $k$  (here 5) equal groups and uses the means as starting values.
- cluster kmedians stemp smanuf spop swind sprecip sdays, /\*  
 \*/ k(5) name(cluster5)

---

# References

---

- Agresti, A. 1996. *Introduction to Categorical Data Analysis*. New York: Wiley.
- Agresti, A. 2002. *Categorical Data Analysis*. Hoboken, NJ: Wiley.
- Aitkin, M. 1978. The analysis of unbalanced cross-classifications (with discussion). *Journal of the Royal Statistical Society, A*, **41**, 195–223.
- Allison, P. D. 1984. *Event History Analysis. Regression for Longitudinal Event Data*. Newbury Park, CA: Sage.
- Altman, D. G. 1991. *Practical Statistics for Medical Research*. London: Chapman & Hall.
- Binder, D. A. 1983. On the variances of asymptotically normal estimators from complex surveys. *International Statistical Review*, **51**, 279–292.
- Boniface, D. R. 1995. *Experimental Design and Statistical Methods*. London: Chapman & Hall.
- Breslow, N. E., & Clayton, D. G. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, **88**, 9–25.
- Calinski, T., & Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics – Theory and Methods A*, **3**, 1–27.
- Caplehorn, J., & Bell, J. 1991. Methadone dosage and the retention of patients in maintenance treatment. *The Medical Journal of Australia*, **154**, 195–199.
- Chatterjee, S., Hadi, A. S., & Price, B. 1999. *Regression Analysis by Example (3rd Edition)*. New York: Wiley.

- Clayton, D., & Hills, M. 1993. *Statistical Models in Epidemiology*. Oxford: Oxford University Press.
- Cleves, M., Gould, W. W., & Gutierrez, R. 2004. *An Introduction to Survival Analysis using Stata (Revised Edition)*. College Station, TX: Stata Press.
- Collett, D. 2002. *Modelling Binary Data (Second Edition)*. London: Chapman & Hall/CRC.
- Collett, D. 2003. *Modelling Survival Data in Medical Research (Second Edition)*. Boca Raton, FL: Chapman & Hall/CRC.
- Cook, R. D. 1977. Detection of influential observations in linear regression. *Technometrics*, **19**, 15–18.
- Cook, R. D. 1979. Influential observations in linear regression. *Journal of the American Statistical Association*, **74**, 164–174.
- Cook, R. D., & Weisberg, S. 1982. *Residuals and Influence in Regression*. London: Chapman & Hall.
- Cox, N. J. 2002a. Speaking Stata: How to face lists with fortitude. *The Stata Journal*, **2**, 202–222.
- Cox, N. J. 2002b. Speaking Stata: How to move step by: step. *The Stata Journal*, **2**, 86–102.
- Cox, N. J. 2003. Speaking Stata: Problems with lists. *The Stata Journal*, **3**, 185–202.
- Crouchley, R., & Davies, R. B. 1999. A comparison of population average and random effects models for the analysis of longitudinal count data with base-line information. *Journal of the Royal Statistical Association, Series A*, **162**, 331–347.
- Diggle, P. J., Heagerty, P., Liang, K.-Y., & Zeger, S. L. 2002. *Analysis of Longitudinal Data*. Oxford: Oxford University Press.
- Duda, R. O., & Hart, P. E. 1973. *Pattern Classification and Scene Analysis*. Chichester: Wiley.
- Efron, B., & Tibshirani, R. 1993. *An Introduction to the Bootstrap*. London: Chapman & Hall.
- Everitt, B. S. 1992. *The Analysis of Contingency Tables*. London: Chapman & Hall.
- Everitt, B. S. 1994. *Statistical Methods for Medical Investigations*. London: Edward Arnold.
- Everitt, B. S. 1995. The analysis of repeated measures: A practical review with examples. *The Statistician*, **44**, 113–135.
- Everitt, B. S. 1996. An introduction to finite mixture distributions. *Statistical Methods in Medical Research*, **5**, 107–127.

- Everitt, B. S. 2001. *Statistics for Psychologists: An Intermediate Course*. Mahwah, NJ: Lawrence Erlbaum.
- Everitt, B. S., & Der, G. 2001. *A Handbook of Statistical Analyses using SAS (Second Edition)*. Boca Raton, FL: Chapman & Hall/CRC.
- Everitt, B. S., & Dunn, G. 2001. *Applied Multivariate Data Analysis (Second Edition)*. London: Edward Arnold.
- Everitt, B. S., & Rabe-Hesketh, S. 1997. *The Analysis of Proximity Data*. London: Edward Arnold.
- Everitt, B. S., Landau, S., & Leese, M. 2001. *Cluster Analysis (Fourth Edition)*. London: Edward Arnold.
- Goldstein, H. 2003. *Multilevel Statistical Models*. London: Arnold.
- Gould, W., Pitblado, J., & Sribney, W. 2003. *Maximum Likelihood Estimation with Stata*. College Station, TX: Stata Press.
- Grambsch, P., & Therneau, T. 1994. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, **8**, 515–526.
- Gregoire, A. J. P., Kumar, R., Everitt, B. S., Henderson, A. F., & Studd, J. W. W. 1996. Transdermal oestrogen for the treatment of severe post-natal depression. *The Lancet*, **347**, 930–934.
- Hamilton, L. C. 2004. *Statistics with Stata 8*. Belmont, CA: Duxbury Press.
- Hand, D. J., & Crowder, M. 1996. *Practical Longitudinal Data Analysis*. London: Chapman & Hall.
- Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., & Ostrowski, E. 1994. *A Handbook of Small Data Sets*. London: Chapman & Hall.
- Hardin, J., & Hilbe, J. 2001. *Generalized Linear Models and Extensions*. College Station, TX: Stata Press.
- Hardin, J., & Hilbe, J. 2002. *Generalized Estimating Equations*. Boca Raton, FL: Chapman & Hall/CRC.
- Harrell, F. E. 2001. *Regression Modeling Strategies. With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York: Springer.
- Hills, M., & De Stavola, B. 2002. *A Short Introduction to Stata for Biostatistics*. London: Timberlake Consultants Press.
- Holtbrugge, W., & Schumacher, M. 1991. A comparison of regression models for the analysis of ordered categorical data. *Applied Statistics*, **40**, 249–259.
- Hotelling, H. 1933. Analysis of complex statistical variables into principal components. *Journal of Educational Psychology*, **24**, 498–520.

- Jackson, J. E. 1991. *A User's Guide to Principal Components*. New York: Wiley.
- Jennrich, R. I., & Schluchter, M. D. 1986. Unbalanced repeated measures models with unstructured covariance matrices. *Biometrics*, **42**, 805–820.
- Klein, J. P., & Moeschberger, M. L. 2003. *Survival Analysis: Techniques for Censored and Truncated Data (Second Edition)*. New York: Springer.
- Kohler, U., & Kreuter, F. 2004. *Data Analysis Using Stata*. College Station, TX: Stata Press.
- Lachenbruch, P., & Mickey, R. M. 1986. Estimation of error rates in discriminant analysis. *Technometrics*, **10**, 1–11.
- Lewine, R. R. J. 1981. Sex differences in schizophrenia: timing or subtypes? *Psychological Bulletin*, **90**, 432–444.
- Liang, K.-Y., & Zeger, S. L. 1986. Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13–22.
- Lindsey, J. K. 1999. *Models for Repeated Measurements*. Oxford, UK: Oxford University Press.
- Lindsey, J. K., & Lambert, P. 1998. On the appropriateness of marginal models for repeated measurements in clinical trials. *Statistics in Medicine*, **17**, 447–469.
- Long, J. S., & Freese, J. 2003. *Regression Models for Categorical Dependent Variables using Stata (Revised edition)*. College Station, TX: Stata Press.
- Mallows, C. L. 1973. Some comments on  $C_p$ . *Technometrics*, **15**, 661–667.
- Manly, B. F. J. 1997. *Randomisation, Bootstrap and Monte Carlo Methods in Biology*. London: Chapman & Hall.
- Matthews, J. N. S., Altman, D. G., Campbell, M. J., & Royston, P. 1990. Analysis of serial measurements in medical research. *British Medical Journal*, **300**, 230–235.
- Maxwell, S. E., & Delaney, H. D. 1990. *Designing Experiments and Analysing Data*. Belmont, CA: Wadsworth.
- McCullagh, P., & Nelder, J. A. 1989. *Generalized Linear Models*. London: Chapman & Hall.
- McLachlan, G., & Peel, D. A. 2000. *Finite Mixture Models*. New York, NY: Wiley.
- Morant, G. M. 1923. A first study of the Tibetan skull. *Biometrika*, **14**, 193–260.

- Nelder, J. A. 1977. A reformulation of linear models. *Journal of the Royal Statistical Society (Series A)*, **140**, 48–63.
- Pearson, K. 1901. On lines and planes of closest fit to points in space. *Philosophical Magazine, Series 6*, **2**, 559–572.
- Rabe-Hesketh, S., Skrondal, A., & Pickles, A. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal*, **2**, 1–21.
- Rabe-Hesketh, S., Pickles, A., & Skrondal, A. 2003. Correcting for covariate measurement error in logistic regression using nonparametric maximum likelihood estimation. *Statistical Modelling*, **3**, 215–232.
- Rabe-Hesketh, S., Skrondal, A., & Pickles, A. 2004a. Generalized multilevel structural equation modeling. *Psychometrika*, in press.
- Rabe-Hesketh, S., Pickles, A., & Skrondal, A. 2004b. *Multilevel and Structural Equation Modeling of Continuous, Categorical and Event Data*. College Station, TX: Stata Press.
- Rawlings, J. O., Pantula, S. G., & Dickey, D. A. 1998. *Applied Regression Analysis: A Research Tool (2nd edition)*. New York: Springer.
- Sackett, D. L., Haynes, R. B., Guyatt, G. H., & Tugwell, P. 1991. *Clinical Epidemiology*. Boston, MA: Little Brown & Company.
- Skrondal, A., & Rabe-Hesketh, S. 2003. Multilevel logistic regression for polytomous data and rankings. *Psychometrika*, **68**, 267–287.
- Skrondal, A., & Rabe-Hesketh, S. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.
- Snijders, T. A. B., & Bosker, R. J. 1999. *Multilevel Analysis*. London: Sage Publications.
- Sokal, R. R., & Rohlf, F. J. 1981. *Biometry*. San Francisco: W. H. Freeman.
- Sprent, P., & Smeeton, N. C. 2001. *Applied Nonparametric Statistical Methods (Third Edition)*. Boca Raton, FL: Chapman & Hall/CRC.
- Sribney, W. 1998. *What are the advantages of using robust variance estimators over standard maximum-likelihood estimators in logistic regression*. Stata FAQ  
[http://www.stata.com/support/faqs/stat/robust\\_var.html](http://www.stata.com/support/faqs/stat/robust_var.html).
- Stata Base Reference Manuals. 2003. *Stata Base Reference Manuals*. College Station, TX: Stata Press.
- Stata Cross-Sectional Time-Series. 2003. *Cross-Sectional Time-Series Reference Manual*. College Station, TX: Stata Press.

- Stata Getting Started. 2003. *Getting Started with Stata for Windows Manual*. College Station, TX: Stata Press.
- Stata Graphics Manual. 2003. *Graphics Reference Manual*. College Station, TX: Stata Press.
- Stata Programming Manual. 2003. *Programming Reference Manual*. College Station, TX: Stata Press.
- Stata User's Guide. 2003. *Stata User's Guide*. College Station, TX: Stata Press.
- Thall, P. F., & Vail, S. C. 1990. Some covariance models for longitudinal count data with overdispersion. *Biometrics*, **46**, 657–671.
- Thara, R., Henrietta, M., Joseph, A., Rajkumar, S., & Eaton, W. 1994. Ten year course of schizophrenia – the Madras Longitudinal study. *Acta Psychiatrica Scandinavica*, **90**, 329–336.
- Verbeke, G., & Molenberghs, G. 2000. *Linear Mixed Models for Longitudinal Data*. New York, NY: Springer.
- Wedderburn, R. W. M. 1974. Quasilikelihood functions, generalised linear models and the Gauss-Newton method. *Biometrika*, **61**, 439–447.
- Williams, R. L. 2000. A note on robust variance estimation for cluster-correlated data. *Biometrics*, **56**, 645–646.
- Winer, B. J. 1971. *Statistical Principles in Experimental Design*. New York: McGraw-Hill.
- Wolfe, J. H. 1971. *A Monte Carlo study of the sampling distribution of the likelihood ratio for mixtures of multinormal distributions*. Technical Bulletin, vol. STB72-2. San Diego: Naval Personnel and Training Research Laboratory.
- Wooldridge, J. M. 2002. *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: The MIT Press.