

Chapter 2

Using RExcel and R Commander

Abstract We review the complete set of Rcmdr menu items, including both the action menu items and the active Dataset and model items. We illustrate the output graphs and tables associated with a least-squares fit. We show the R Commander window and the RGUI Console window.

2.1 Appearance

Users will normally have only one version of Excel installed on their computer. Most illustrations in this book use the most recent, Excel 2007. The RExcel behavior is identical in both Excel 2003 and Excel 2007. The Rcmdr behavior is identical in both Excel versions and without Excel.

The RExcel and Rcmdr menus in Excel 2007 (Fig. 2.1) and Excel 2003 (Fig. 2.2) and on the R Commander window (Fig. 2.3) have slightly different appearances.

The RExcel menu is an item on the Menu Commands box of the Add-Ins tab of Excel 2007. The RExcel menu is an item on the main menu of Excel 2003.

The Rcmdr menu is a toolbar on the Ribbon of the Add-Ins tab of Excel 2007 and is a toolbar on the main toolbar of Excel 2003. The Rcmdr menu is a toolbar on the Rcmdr window when Excel is not running.

The content and behavior of the RExcel menu in Fig. 2.4 are identical in both versions of Excel. The content and appearance of the Rcmdr menu are almost identical in all three settings. The Rcmdr ► Edit menu item does not appear in current releases of RExcel because it is not needed in the Excel setting. (It is shown on some of our earlier screenshots.) We display each item on the menus in the upcoming figures.

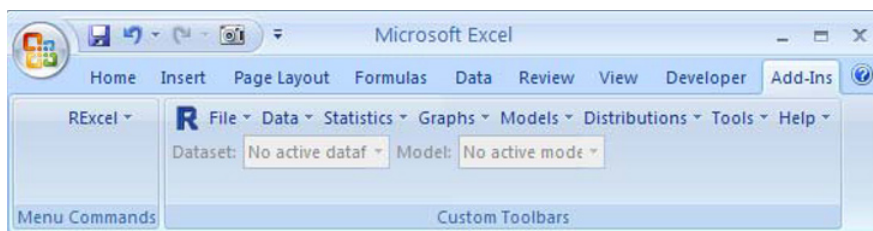


Fig. 2.1 RExcel and Rcmdr menus on the Add-Ins tab in Excel 2007.

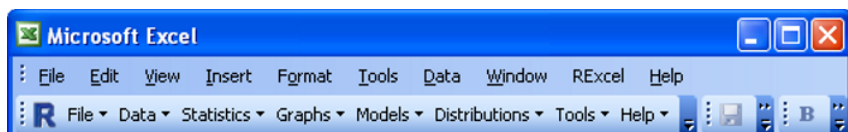


Fig. 2.2 RExcel and Rcmdr menus in Excel 2003.

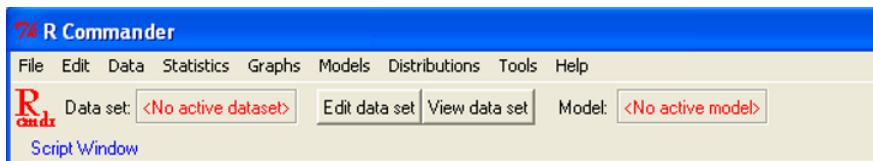


Fig. 2.3 Rcmdr menu on the R Commander window when Rcmdr has been started from RExcel by clicking the with separate menus menu item (see Fig. 1.11) or directly from R (see Fig. 1.15).

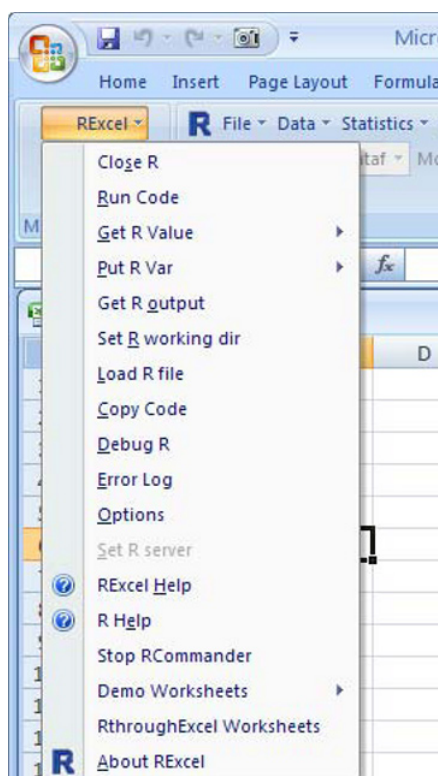


Fig. 2.4 RExcel menu. This is the main menu for starting and stopping the interface between Excel and R. This menu can also be used for communicating between R and Excel. We will usually use the Context menu (right-click menu) in Excel (Fig. 2.14) for communication between the two programs.

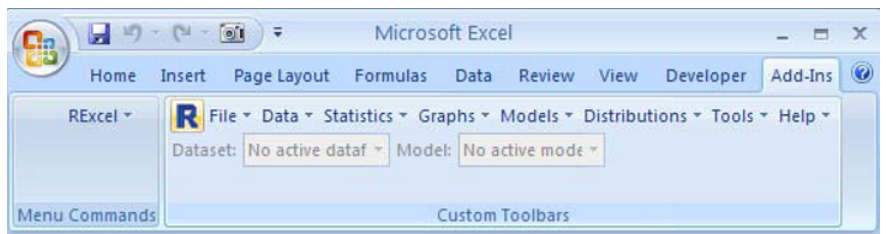


Fig. 2.5 **R** button. When you type commands to R directly (in either the Rcmdr script window or the R Console), as distinct from clicking in the Rcmdr menu in Excel, then Excel, R, and the Rcmdr menu can get out of phase. Clicking the **R** button resynchronizes them.

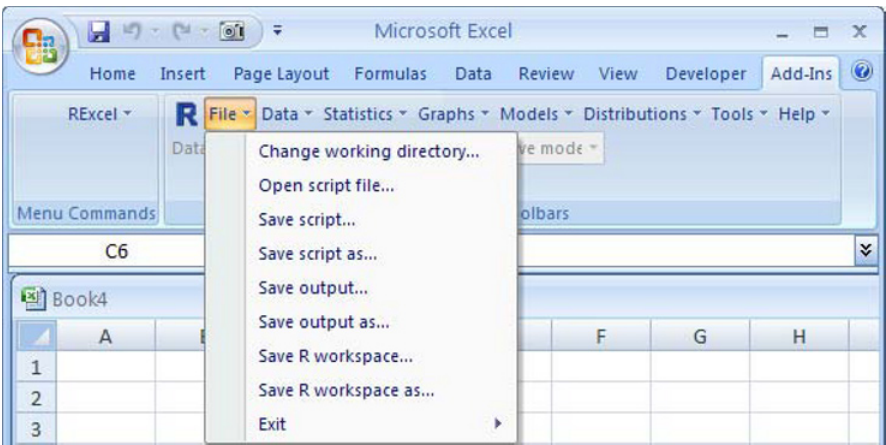


Fig. 2.6 Rcmdr File menu. We normally do not use this menu.

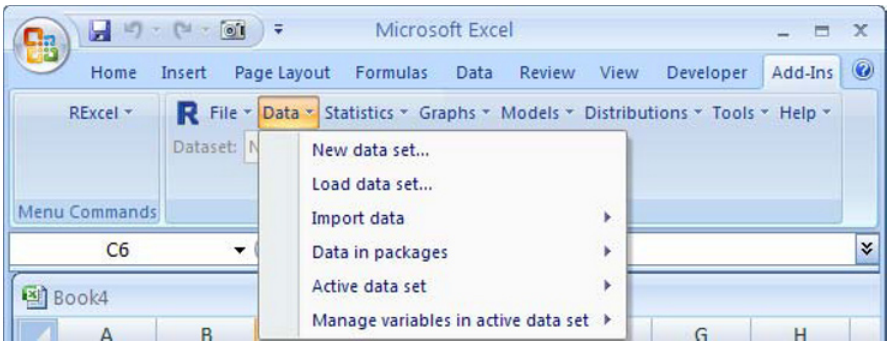


Fig. 2.7 Rcmdr Data menu. This menu is very helpful for bringing data into R and for restructuring the data after it is already in R.

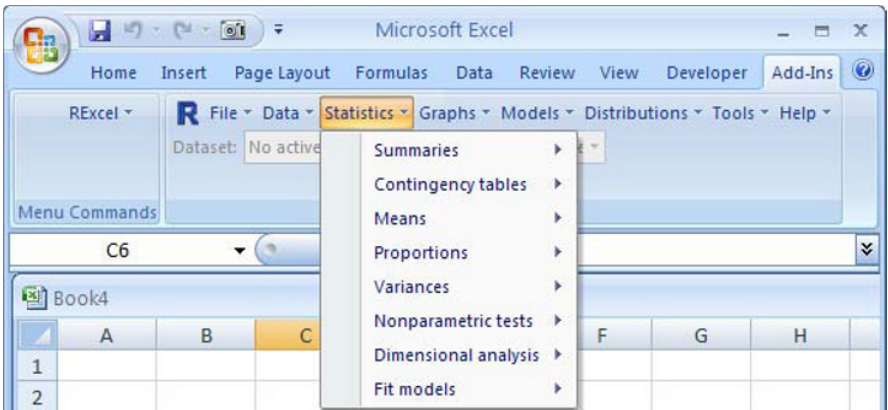


Fig. 2.8 Rcmdr Statistics menu. This is the workhorse menu for computations and analysis.

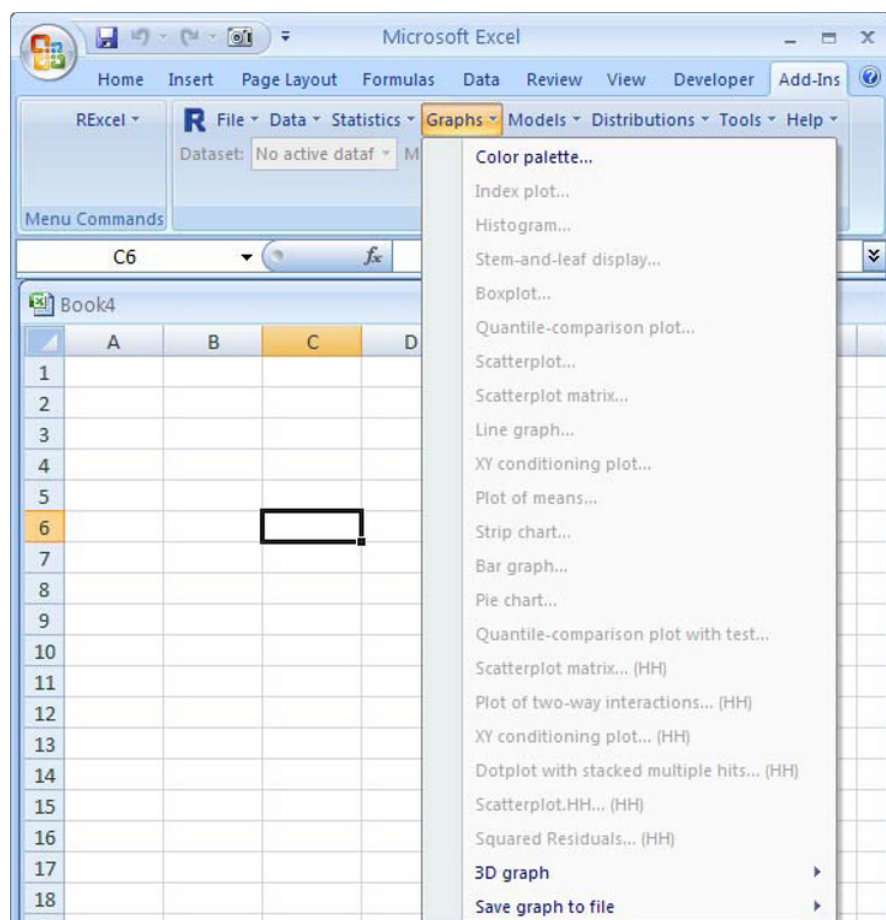


Fig. 2.9 Rcmdr Graphs menu. This is the workhorse menu for graphs.

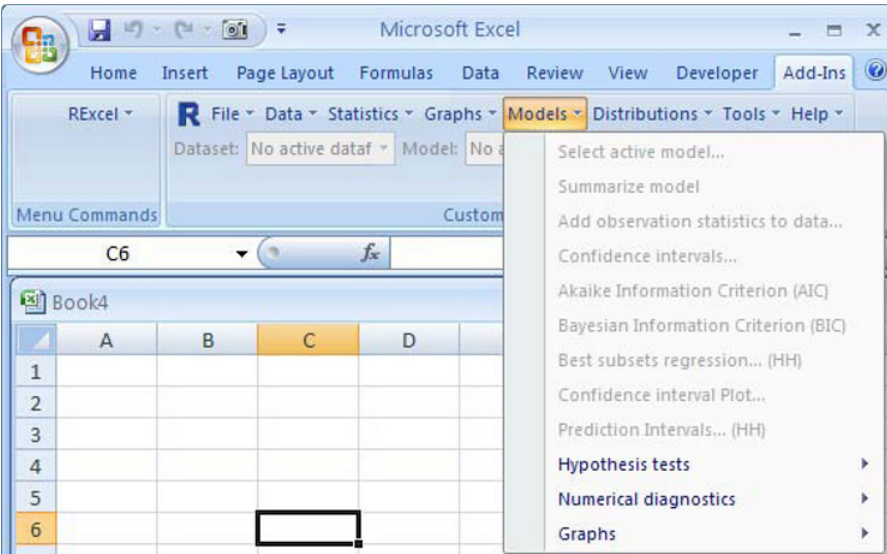


Fig. 2.10 Rcmdr Models menu. This menu allows follow-up display of results from analyses calculated in the Statistics menu.

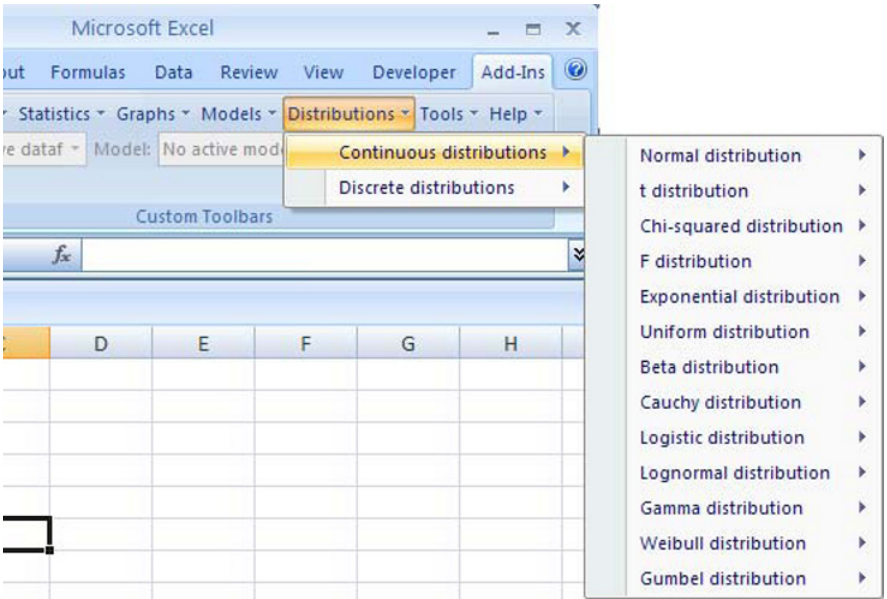


Fig. 2.11 Rcmdr Distributions menu. The normal, t , F , chi-squared, and other tables are accessible from the Continuous distributions menu shown here. The binomial and other tables are accessible from the Discrete distributions menu.

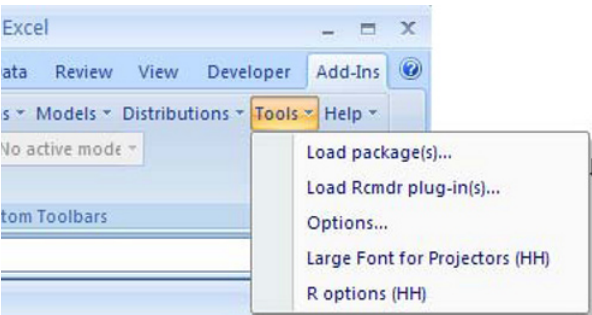


Fig. 2.12 Rcmdr Tools menu. The Options... item provides access to display options (font size, for example) for the Rcmdr window. Rcmdr plug-ins are a mechanism that permits people other than the author of Rcmdr to provide additional menu items on the Rcmdr menu bar.

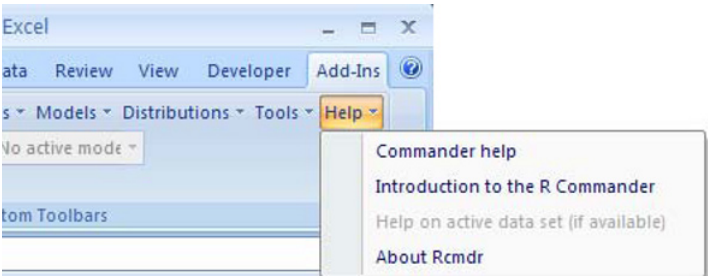


Fig. 2.13 Rcmdr Help menu. The Introduction to the R Commander is the best reference.

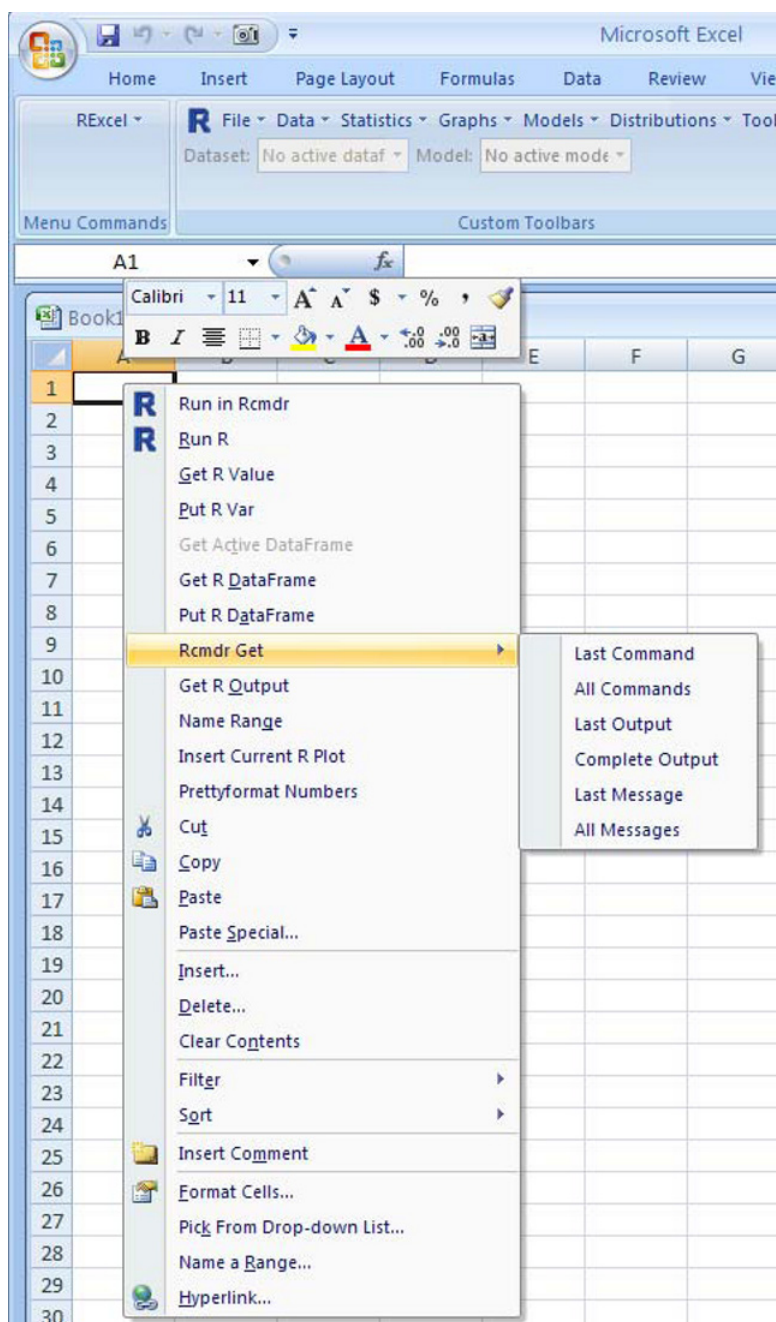
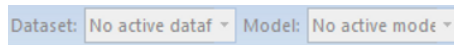


Fig. 2.14 Excel Context menu (right-click menu) displayed when REExcel is active and Rcmdr is loaded. This menu is the primary tool used to communicate between Excel and R.

2.2 The Dataset and Model Menus

In all previous figures (see Fig. 2.14, for example) the toolbar shows



This portion of the toolbar shows the Rcmdr active dataset and active model. Both initially show the value “not active”.

When we work with a dataset, the active dataset is the one to which the menu commands are applied. In this section, we look at a dataset and model it with a simple least-squares fit.

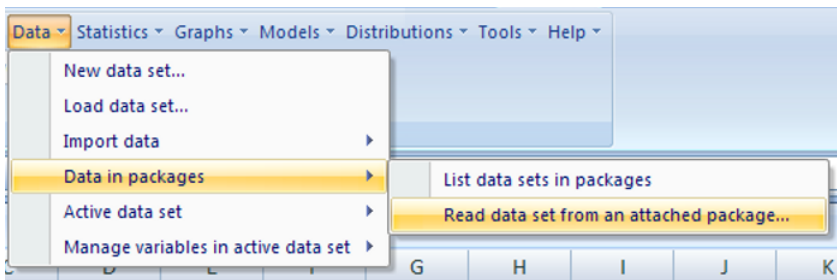


Fig. 2.15 R includes several datasets. We will make one of them active by clicking in the Rcmdr Data ► Data in packages ► Read data set from an attached package... menu. This opens the dialog box in Fig. 2.16.

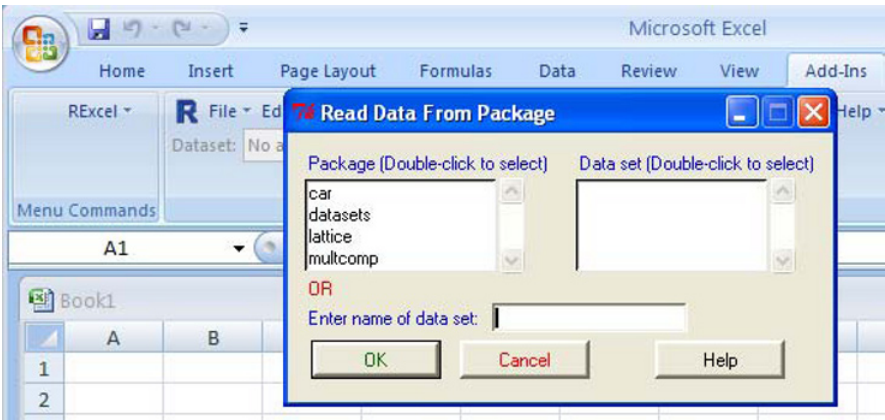


Fig. 2.16 R consists of a “base package” and many additional packages. The Read Data From Package menu shows those currently attached packages that include datasets.

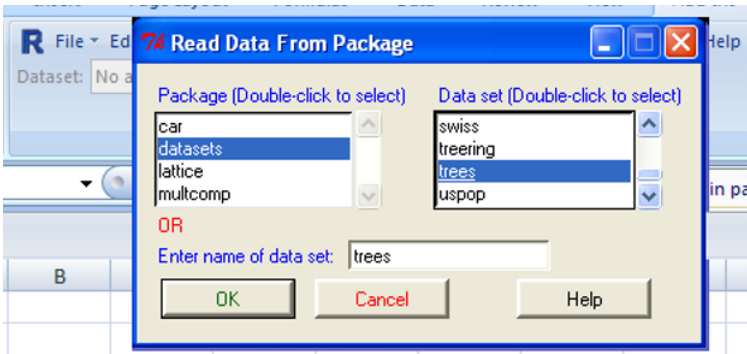


Fig. 2.17 Double-click a package name to put the list of datasets in the right-hand menu, double-click the trees dataset, and click OK.

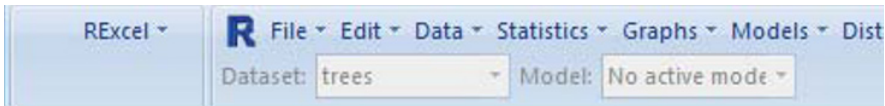



Fig. 2.18 The toolbar now shows trees as the active dataset.

We query R on the dataset by typing `?trees` in the Script Window and clicking the  button. R replies by displaying a help file. The screenshots of the query and the help file are shown in Fig. 2.29 and 2.30. The help file says the data is the

Girth, Height and Volume for Black Cherry Trees

Description:

This data set provides measurements of the girth, height and volume of timber in 31 felled black cherry trees. Note that girth is the diameter of the tree (in inches) measured at 4 ft 6 in above the ground. A data frame with 31 observations on 3 variables.

```
[ ,1] Girth numeric Tree diameter in inches  
[ ,2] Height numeric Height in ft  
[ ,3] Volume numeric Volume of timber in cubic ft
```

We need to look at the numbers in the dataset. The easiest way is to bring the numbers into Excel, using the technique described in Section 3.6.

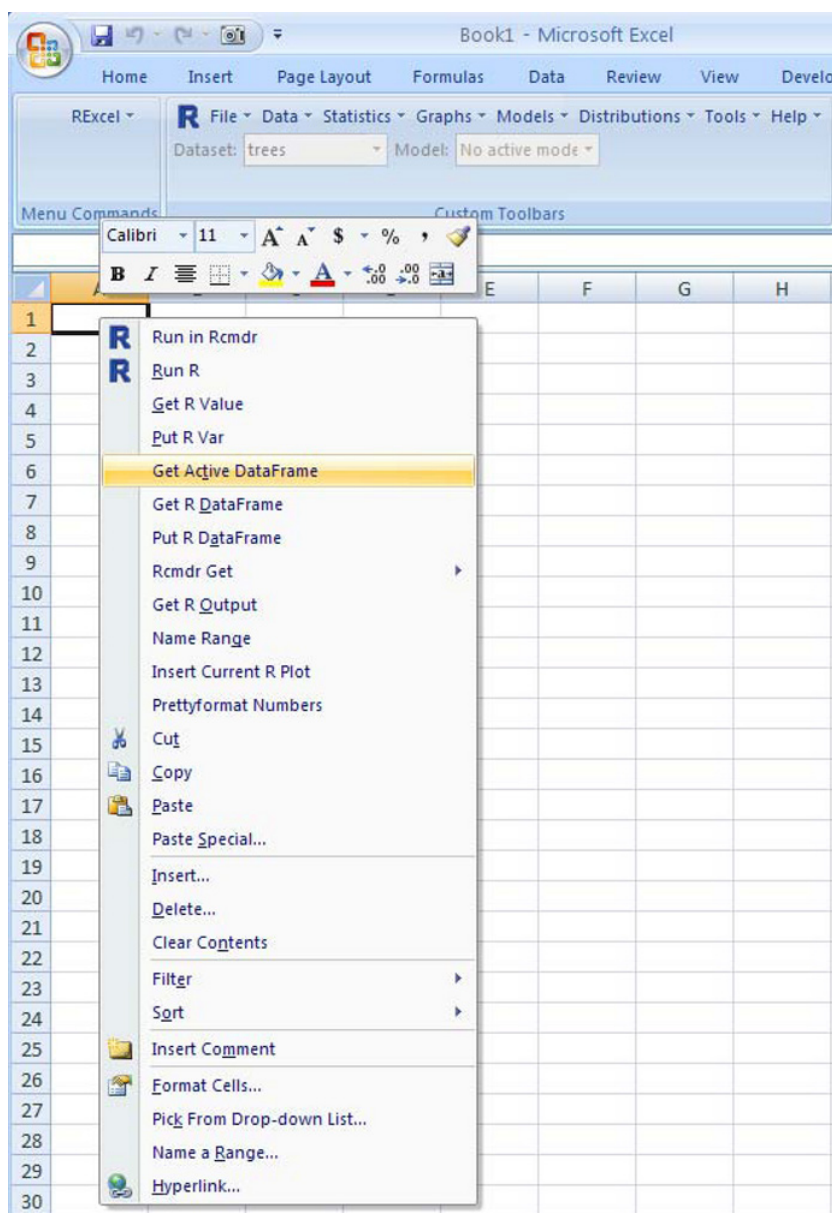


Fig. 2.19 Use the Context menu (right-click menu) to copy the active dataset named in the Dataset toolbar item into the Excel worksheet. We do so by activating an empty worksheet, clicking in cell A1 and then right-clicking Get Active DataFrame. The dataset will appear beginning with the highlighted cell. If you choose to bring data into an existing workbook, be careful about the choice of starting cell. There is no undo for this transfer.

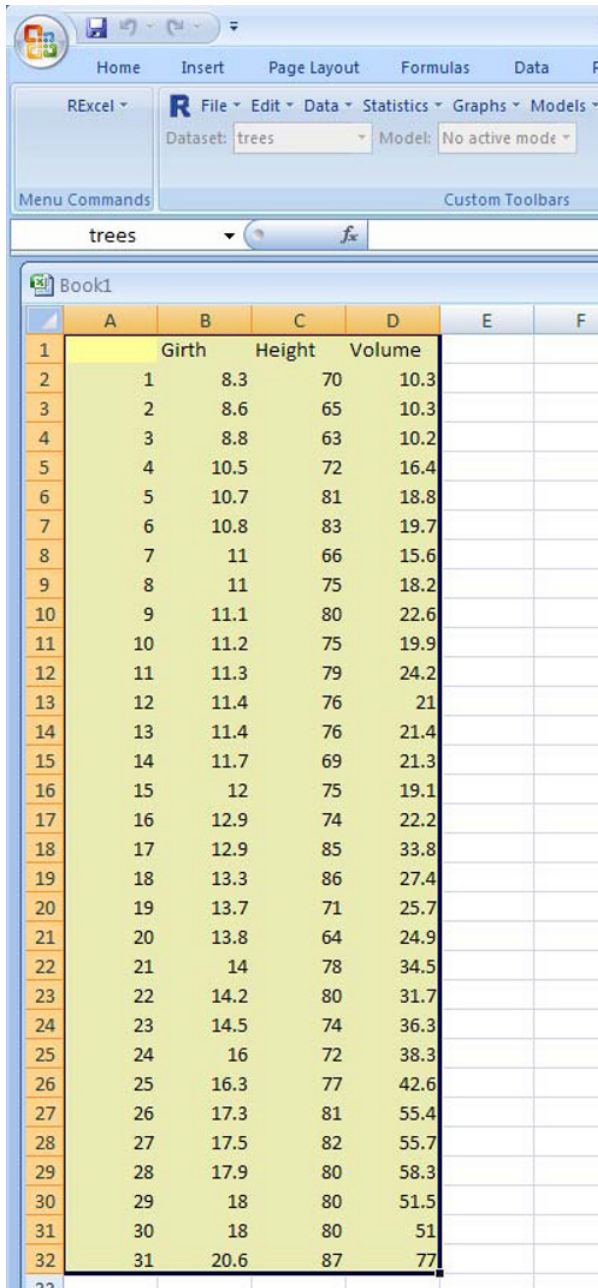


Fig. 2.20 The dataset is displayed in the worksheet. This dataset has four columns: the row name (in this example, a numerical index) and three variables. The region of the worksheet containing the dataset is colored and is given a name in the cell identifier box on the Excel toolbar.

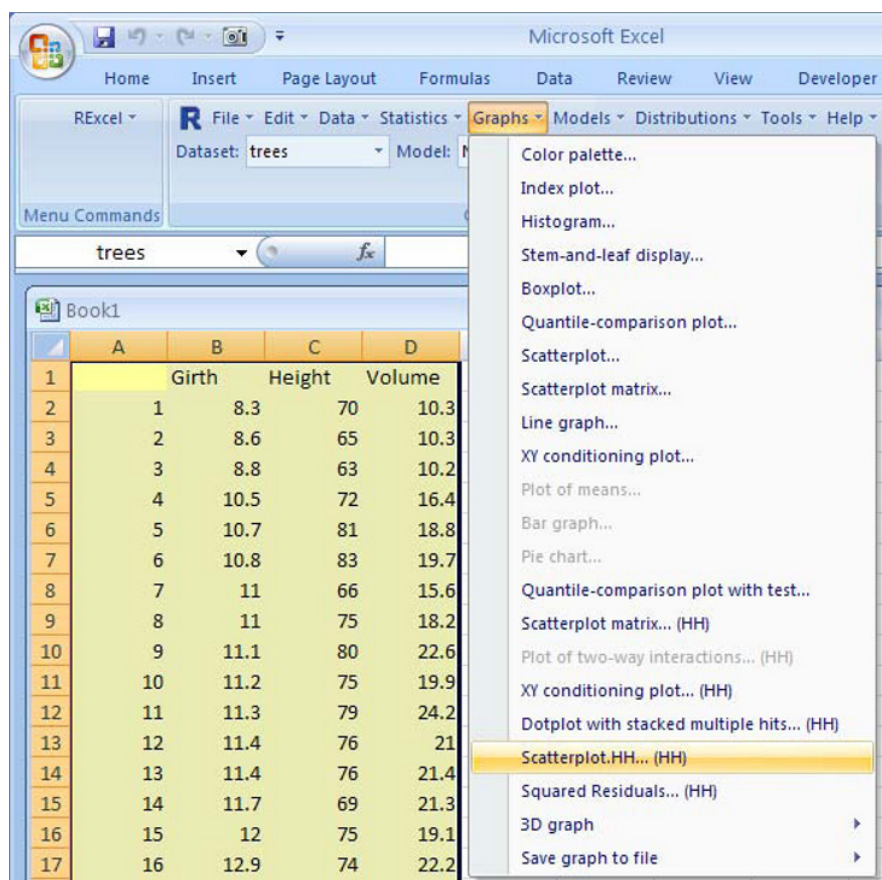


Fig. 2.21 Now that there is an active dataset, we can use the Graphs ► Scatterplot matrix... (HH) menu item. Menu items with ... indicate that the dialog box in Fig. 2.22 will ask for further information. The boldfaced menu items are the ones that make sense for the active dataset. The other items are grayed out.

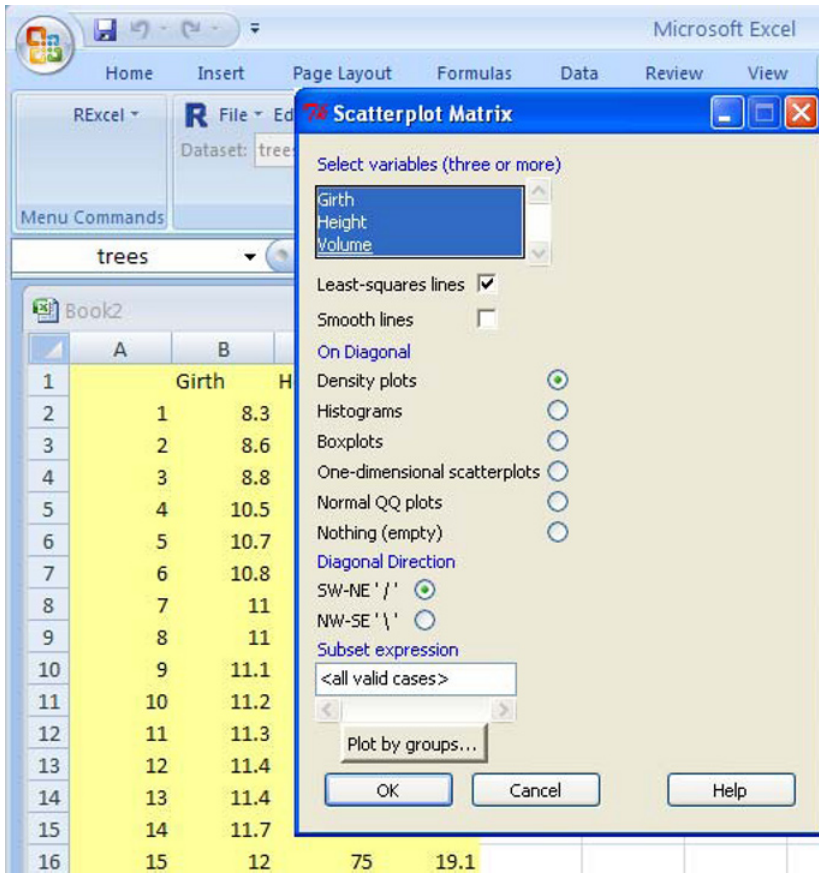


Fig. 2.22 Many of the menus in the Rcmdr dialog boxes include variable-selection dropdown boxes. When there is only one variable in the active dataset that meets the criterion, that variable is shown highlighted when the dialog box opens. When there are multiple variables, or when there is an optional variable selection, then no variables are shown highlighted when the dialog box opens. In this example, select all three variables, accept the defaults for the other items, and click OK.

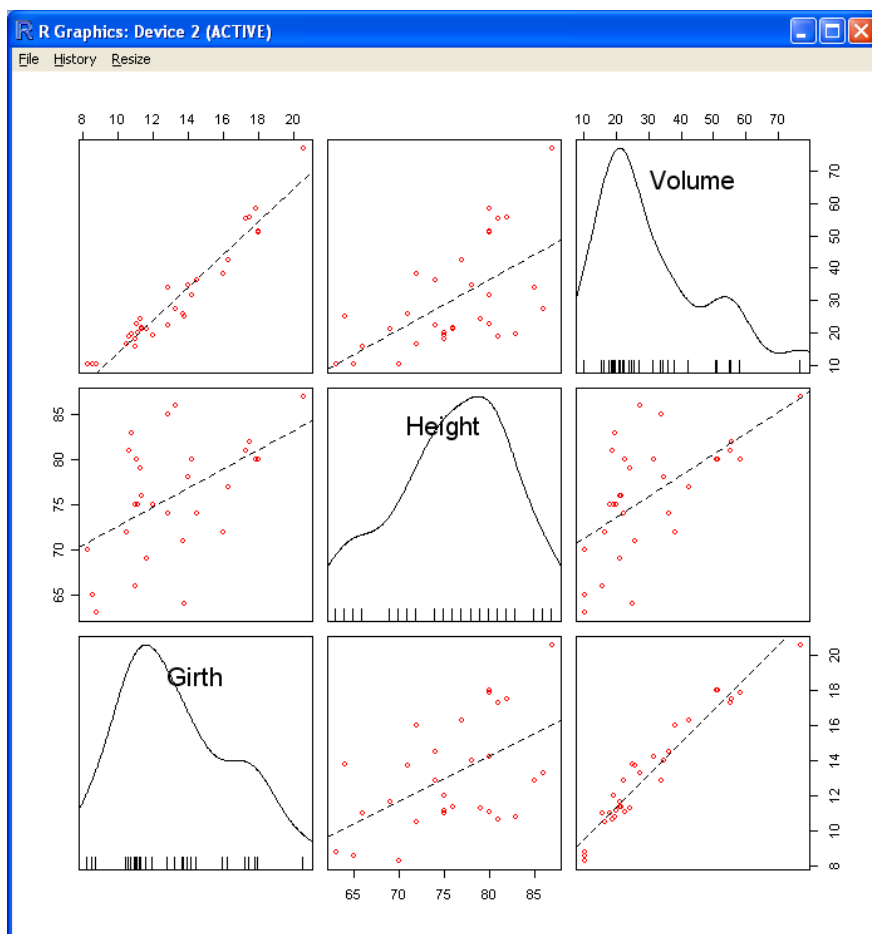


Fig. 2.23 The scatterplot matrix is a matrix of scatterplots of each variable plotted against the others. The sequence of variables normally starts at the lower left. Look at the plot of $\text{Volume} \sim \text{Girth}$ in the upper left panel and observe the linearity of the plot. This suggests a least-squares fit (see Chapter 8 for more information on least squares) might be appropriate. The scatterplot matrix displays on the main diagonal the estimated univariate densities for each of the variables. The notation $y \sim x$ means that the left variable is on the vertical axis and the right variable is on the horizontal axis. This notation is used throughout R to specify statistical models.

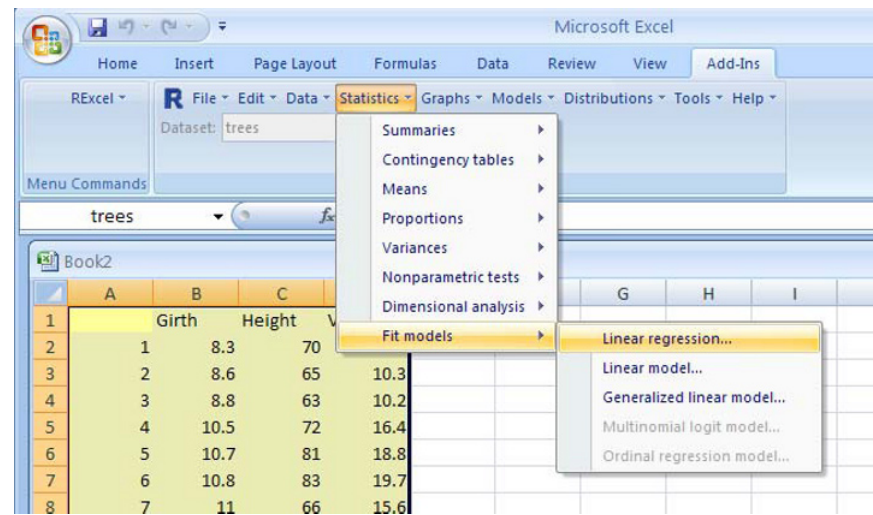


Fig. 2.24 Specify a least-squares fit on the active dataset by selecting the Statistics ► Fit models ► Linear regression... menu item.

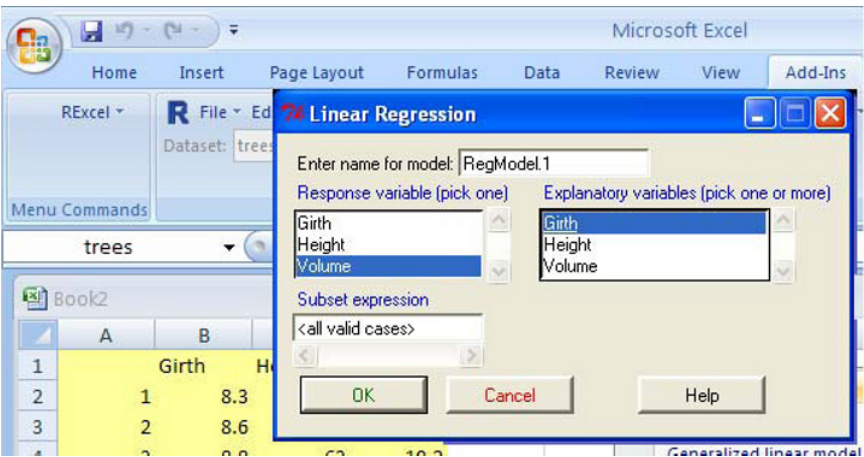


Fig. 2.25 The Linear Regression dialog box asks for a response variable, and we select the y-variable Volume from the Volume ~ Girth plot in Fig. 2.23. We select as Explanatory variable the x-variable Girth. Output from this dialog box is in Fig. 2.26. The dialog box also changes the active model as shown in Fig. 2.27.

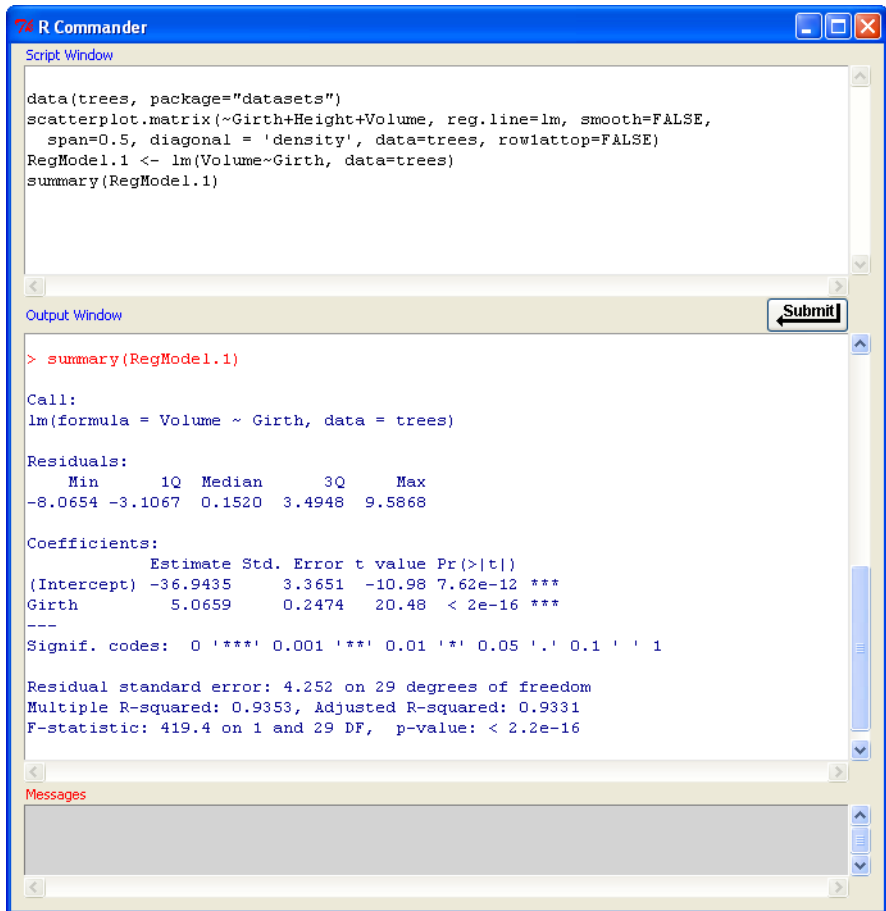


Fig. 2.26 The R Commander window shows the printed output from the dialog box in Fig. 2.25. The generated R code is in the top Script Window. The analysis object is given the name `RegModel.1`. The summary of the regression analysis in model `RegModel.1` is in the bottom Output Window. The coefficients of the least-squares line are displayed in the `coefficients` section of the output. The least-squares line is

$$\hat{y} = -36.9435 + 5.0659x$$

The least-squares line is displayed as the dotted line in the `Volume ~ Girth` panel on the plot in Fig. 2.23.



Fig. 2.27 The Model item in the Rcmdr menu now shows the name of the active regression analysis object `RegModel.1`. Compare to Fig. 2.18, where only the active dataset is displayed. The summary of the model object is automatically displayed in Fig. 2.26. Additional graphs and tables can be constructed from the active model. All menu items in the Models menu use the active model object.

2.3 R Console

R was originally designed as a command language; commands were typed into a text-based input area on the computer screen and the program responded with a written response to each command. The written response is usually a table. In this book, we normally do not use the command language directly. Occasionally, we need it; therefore, we give a small introduction here.

Start R as in Figs. 1.13 and 1.14. The R Console in Fig. 1.14 is essentially a typewriter window. The R Console opens with information and then a prompt mark, usually `>`, indicating that it is ready for us to type. We type a line, ending with the `↵` Enter key. Then R types one or more lines in return, ending with the prompt `>`.

In Fig. 2.28, we repeat in the R Console the regression example first shown in the Rcmdr Output Window in the bottom section of Fig. 2.26.

```

>
> data(trees, package="datasets")
> scatterplot.matrix(~Girth+Height+Volume, reg.line=lm, smooth=FALSE,
+   span=0.5, diagonal = 'density', data=trees, rowlattop=FALSE)
> RegModel.1 <- lm(Volume~Girth, data=trees)
> summary(RegModel.1)

Call:
lm(formula = Volume ~ Girth, data = trees)

Residuals:
    Min       1Q   Median       3Q      Max
-8.0654 -3.1067  0.1520  3.4948  9.5868

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
Girth         5.0659     0.2474   20.48 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.252 on 29 degrees of freedom
Multiple R-squared:  0.9353,    Adjusted R-squared:  0.9331
F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16

> |

```

Fig. 2.28 Repeat of the regression analysis in Fig. 2.26. This figure is primarily a demonstration of the R Console window and only incidentally an illustration of the regression analysis. The R Console displays a prompt mark `>` indicating that it is ready for us to type. We type each line into the R Console window and end each line with an `↵` Enter keypress. After each *complete* line is typed, the R Console responds with a prompt `>` saying it is ready for a new line. After *incomplete* lines—in this example, the first line of the `scatterplot.matrix()` function call—the R Console responds with a continuation prompt `+`. The `data()` command produces no printed output. The `scatterplot.matrix()` command tells R to construct and display the plot in Fig. 2.23. The `summary()` command produces an output display, which is shown immediately after the command line and before the next prompt.

2.4 R Commander Window

The R Commander window is so named because it generates R commands from clicks on the menu. Let's look again at Fig. 2.26, this time focusing on the structure of the window.

In Figs. 2.15 and 2.18, we clicked menus and dialog boxes. The R Commander translated those clicks into the `data()` command in the Script Window in the top half of the R Commander window in Fig. 2.26. It also put the command in the Output Window in the bottom half of the R Commander window in Fig. 2.26 (it scrolled offscreen in this illustration).

In Figs. 2.21 and 2.22 we clicked menus and dialog boxes. The R Commander translated those clicks into the `scatterplot.matrix()` command in the Script Window. It also put the command in the Output Window. This command also scrolled offscreen in this illustration.

In Figs. 2.24–2.25 we clicked menus and dialog boxes. The R Commander translated those clicks into the `lm()` and `summary()` commands in the Script Window. It also put the command and its printed output in the Output Window.

The Script Window simulates a program file of commands that could be typed into the R Console. The Output Window simulates the R Console window.

It is possible, and sometimes useful, to type commands into the R Commander Script Window and submit them to the R Commander Output Window for execution. Just enter a complete command in the Script Window and click the Submit button. In Section 2.2, we showed the information in the help file for the `trees` dataset. In Section 2.5, as an example of typing into the R Commander window, we show screenshots of the query in Fig. 2.29 and the help file in Fig. 2.30.

2.5 R Help Files

All functions and datasets in R have a help file. Help files can be accessed by typing a query, for example, `?trees`, in the R Commander Script Window and then clicking the Submit button as shown in Fig. 2.29. In addition, all Rcmdr dialog boxes include a Help button that will open the appropriate help file.

Fig. 2.30 shows the help file for the `trees` dataset in the standard MS help format. R can present help files in other formats. See `?help` for more information. If the help files appear in some other format, you can force R to use the standard MS help format by entering `options(chmhelp=TRUE)` in the R Commander Script Window and then clicking the Submit button. You can also change the help file format from the RExcel ► Options dialog box.

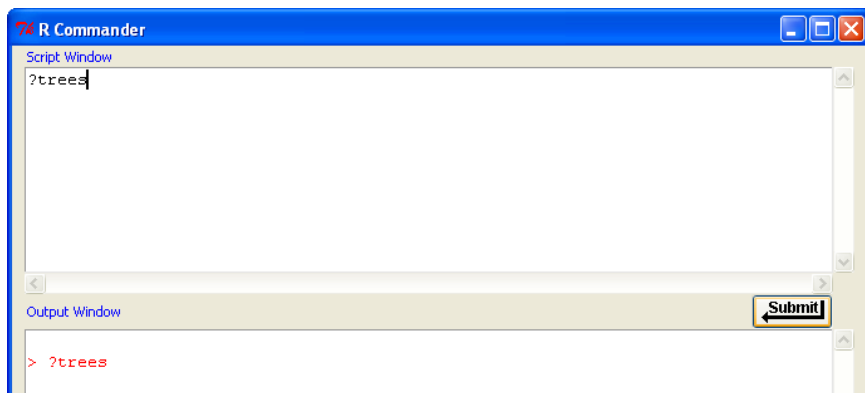



Fig. 2.29 Type the help query `?trees` in the R Commander Script Window and leave the cursor on that line. Then click the  button. The R Commander copies the line to the Output Window and executes the line by opening the help file in Fig. 2.30 to a discussion of the `trees` dataset.

2.6 Messages from R, Rcmdr, or Excel

R and Excel are two different processes running on your computer. The RExcel interface coordinates communication between them. It is possible for them to get out of phase, particularly during startup. See Appendix B for remedies for some of the more likely problems.

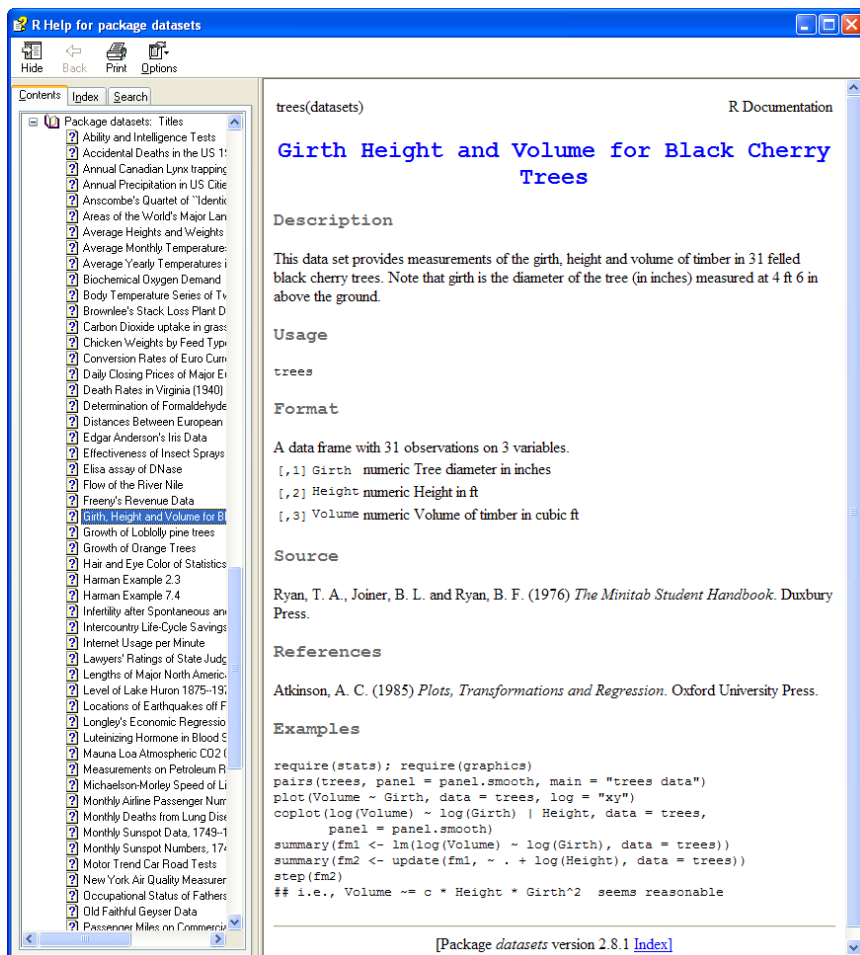


Fig. 2.30 This is an illustration of a help window in the standard MS help format. The right panel describes the dataset. The left panel opens to the table of contents for that help file. In this example, the help file contains descriptions of all the datasets included as part of base R.

<http://www.springer.com/978-1-4419-0051-7>

R Through Excel

A Spreadsheet Interface for Statistics, Data Analysis,
and Graphics

Heiberger, R.M.; Neuwirth, E.

2009, XXIV, 344 p. 300 illus. in color., Softcover

ISBN: 978-1-4419-0051-7