New R Commander Features

John Fox

2020-09-01

This document describes new significant features introduced subsequent to the publication of Fox (2017), *Using the R Commander* (called "the book" or "the text" below) in July 2016. The book is current as of version 2.2-4 of the **Rcmdr** package. Typically a new minor version of the **Rcmdr** package is released each summer. To see all changes to the R Commander, read the `NEWS` file, for example, by entering the command `news(package="Rcmdr")` at the `>` command prompt in the R console.

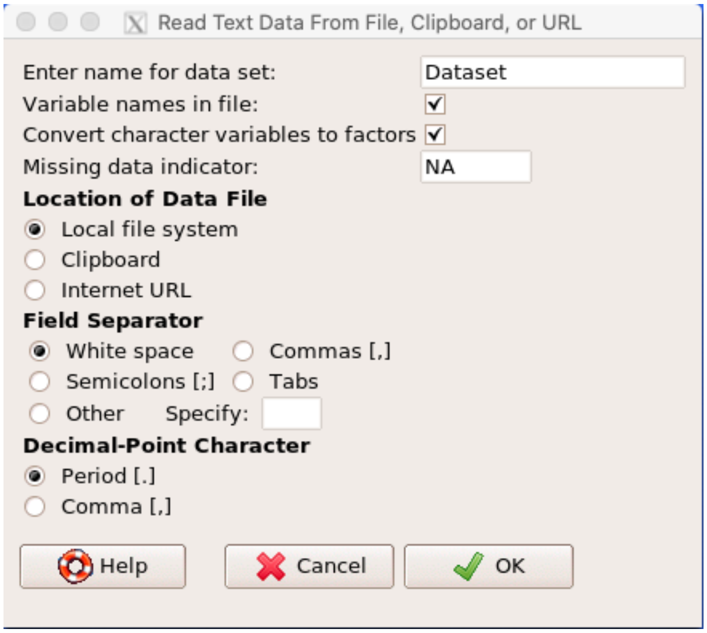
1 Rcmdr Version 2.7-0 (August 2020)

1.1 Adaptations to New `"stringsAsFactors"` Default Setting in R

As of version 4.0.0, the `"stringsAsFactors"` option in R is set to `FALSE` by default, a setting affecting some standard R functions, such as `read.table()`, that are used in the R Commander. For the most part, the R Commander treats character variables as if they were factors, as is true of statistical-modeling functions such as `lm()`.

In a few cases, however, the new default caused problems for R Commander users. There are several changes in **Rcmdr** version 2.7-0 to accommodate the new default for `"stringsAsFactors"`. Some of these are invisible to the user, but (as illustrated in Figure 1), the *Read Text Data* dialog has acquired a *Convert character variables to factors* check box, which is selected by default, producing the default behavior in R (and the R Commander) prior to version 4.0.0.

There is also a new *Convert Character Variables to Factors* dialog, accessible via the *Data > Manage variables in active data set > Convert character variables to numeric* menu item; this dialog is entirely straightforward and so I don't show it here. Finally, the menu item *Active data set > Convert all character variables to factors* does what its name implies.



Read Text Data From File, Clipboard, or URL

Enter name for data set: Dataset

Variable names in file: ☑

Convert character variables to factors ☑

Missing data indicator: NA

Location of Data File

- ● Local file system
- ○ Clipboard
- ○ Internet URL

Field Separator

- ● White space ○ Commas [,]
- ○ Semicolons [;] ○ Tabs
- ○ Other Specify:

Decimal-Point Character

- ● Period [.]
- ○ Comma [,]

Figure 1: The modified *Read Text Data* dialog box, with a *Convert character variables to factors* check box, which is checked by default. This and other screen-shots in this section were produced under macOS.

1.2 New Repeated-Measures ANOVA/ANCOVA Dialogs and Data Management

Version 2.7-0 of the R Commander introduces two new dialogs under the *Statistics > Means* menu for analysis of variance and analysis of covariance with one or two repeated-measures (i.e., within-subjects) factors. There may be an arbitrary number of between-subjects factors and covariates. These dialogs are a traditional alternative to mixed-effects models when the repeated-measures part of the model is balanced.

To complement these repeated-measures dialogs, there are two new data-management dialogs under the *Data > Active data set* menu to convert data from "long" to "wide" format, where there is one row per subject, with repeated measures appearing as separate columns (as is required by the new repeated-measures dialogs), and from wide to long format, where there is a separate row for each combination of repeated-measures factors within a subject and where the response appears in a single column (as is required by the mixed-models dialogs).

1.2.1 One Repeated-Measures Factor

I'll employ] the `OBrienKaiser` data set (O'Brien and Kaiser, 1985, Table 7) in the **carData** package to illustrate repeated-measures ANOVA with one and two repeated-measures factors. I use the *Read Data from Package* dialog (not shown) to input the data set and make it the active data set in the R Commander. The data are in wide format, with 16 rows represent 16 subjects, and include the following variables:

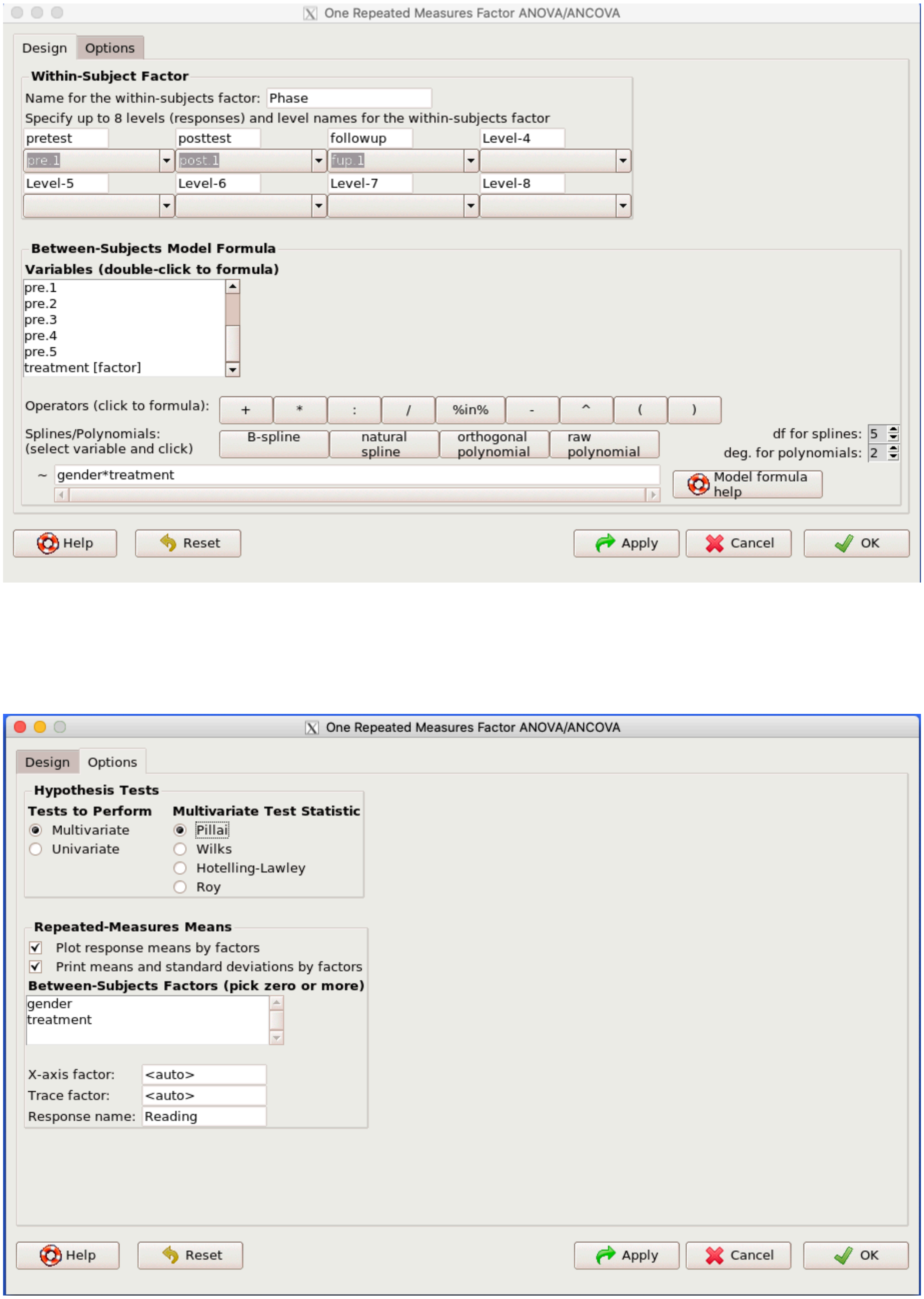
`treatment` A between-subjects factor, with levels `"control"`, `"A"`, and `"B"`.

`gender` A between-subjects factor, with levels `"F"` and `"M"`.

`pre.1` ... `fup.5` Fifteen variables, labeled `pre.1` through `pre.5`, `post.1` through `post.5`, and `fup.1` through `fup.1`, representing the response under three levels of a repeated-measures *phase* factor, representing pretest scores, posttest scores, and followup scores, crossed with five levels of an *hour*-within-phase factor.

The data were contrived, but for concreteness, we can imagine that 16 learning-disabled students were taught reading by the standard (`"control"`) method or by one of two novel methods (`"A"` or `"B"`), and that the response represents each student's reading performance at different points in the study. The data are naturally analyzed using a design with two between-subjects factors (`treatment` and `gender`) and two within-subjects factors (`phase` and `hour`), but for a preliminary example, I'll consider one within-subjects factor, using the first-hour scores for the three phases.

To this end, I select *Statistics > Means > One-factor repeated-measures ANOVA/ANCOVA* from the R Commander menus, and complete the resulting dialog box as in Figure 2. The top panel of the *Design* tab defines an optional name for the within-subjects factor, here `Phase`; up to eight variables corresponding to the levels of the within-subjects factor, here the three variables `pre.1`, `post.1`, and `fup.1`; and optional names for the levels of the within-subjects factor, `pretest`, `posttest`, and `followup`. The within-subjects part of the model is specified just like the right-hand side of a linear model, in this case by double-clicking on `gender` in the variable list, clicking the `*` operator, and double-clicking on `treatment`. In the *Options* tab, I check the two boxes to plot means and print tables of means and standard deviations by factors, and select the two between-subjects factors, leaving other options at their defaults. Clicking *OK* produces the printed output in Figure 3 and the graph in Figure 4.



One Repeated Measures Factor ANOVA/ANCOVA

Design | Options

Within-Subject Factor

Name for the within-subjects factor: Phase

Specify up to 8 levels (responses) and level names for the within-subjects factor

pretest	posttest	followup	Level-4
pre.1	post.1	fup.1	
Level-5	Level-6	Level-7	Level-8

Between-Subjects Model Formula

Variables (double-click to formula)

pre.1
pre.2
pre.3
pre.4
pre.5
treatment [factor]

Operators (click to formula): + * : / %in% - ^ ( )

Splines/Polynomials: (select variable and click) B-spline | natural spline | orthogonal polynomial | raw polynomial

df for splines: 5

deg. for polynomials: 2

~ gender*treatment

Model formula help

Help | Reset | Apply | Cancel | OK

One Repeated Measures Factor ANOVA/ANCOVA

Design | Options

Hypothesis Tests

Tests to Perform

● Multivariate
○ Univariate

Multivariate Test Statistic

● Pillai
○ Wilks
○ Hotelling-Lawley
○ Roy

Repeated-Measures Means

☑ Plot response means by factors
☑ Print means and standard deviations by factors

Between-Subjects Factors (pick zero or more)

gender
treatment

X-axis factor: <auto>
Trace factor: <auto>
Response name: Reading

Help | Reset | Apply | Cancel | OK

Figure 2: The *One Repeated-Measures Factor ANOVA/ANCOVA* dialog box, showing the *Design* tab (top) and *Options* tab (bottom).

```
> Anova(lm(cbind(pre.1, post.1, fup.1) ~ gender*treatment, data=OBrienKaiser),
+   idata=data.frame(Phase=factor(c('pretest', 'posttest', 'followup'))), idesign = ~Phase,
+   test.statistic="Pillai")

Type II Repeated Measures MANOVA Tests: Pillai test statistic
                       Df test stat approx F num Df den Df        Pr(>F)
(Intercept)             1   0.96482  274.269      1     10 0.00000001346 ***
gender                  1   0.28313    3.950      1     10     0.0749501 .
treatment               2   0.47972    4.610      2     10     0.0381215 *
gender:treatment        2   0.41852    3.599      2     10     0.0664775 .
Phase                   1   0.85793   27.174      2      9     0.0001536 ***
gender:Phase            1   0.03796    0.178      2      9     0.8401910
treatment:Phase         2   0.61660    2.229      4     20     0.1023679
gender:treatment:Phase  2   0.28058    0.816      4     20     0.5299762
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> repeatedMeasuresPlot(OBrienKaiser, within=c("pre.1", "post.1", "fup.1"),
+   within.names="Phase", within.levels=list(Phase=c("pretest", "posttest", "followup")),
+   print.tables=TRUE, plot.means=TRUE, between.names=c("gender", "treatment"),
+   response.name="Reading")

Means of Reading
                treatment  control        A        B
Phase    gender
pretest  F                4.000000 2.500000 3.250000
         M                3.333333 6.000000 4.333333
posttest F                4.000000 3.000000 5.500000
         M                3.000000 8.000000 6.666667
followup F                4.000000 5.500000 6.750000
         M                4.333333 8.500000 7.000000


Standard deviations of Reading
                treatment   control         A         B
Phase    gender
pretest  F                1.4142136 0.7071068 1.5000000
         M                2.0816660 1.4142136 1.5275252
posttest F                2.8284271 1.4142136 1.2909944
         M                1.0000000 1.4142136 2.0816660
followup F                0.0000000 0.7071068 0.5000000
         M                2.5166115 0.7071068 1.7320508
```

Figure 3: Repeated-measures ANOVA with one within-subjects factor for the O'Brien-Kaiser data, reporting the default MANOVA tests.

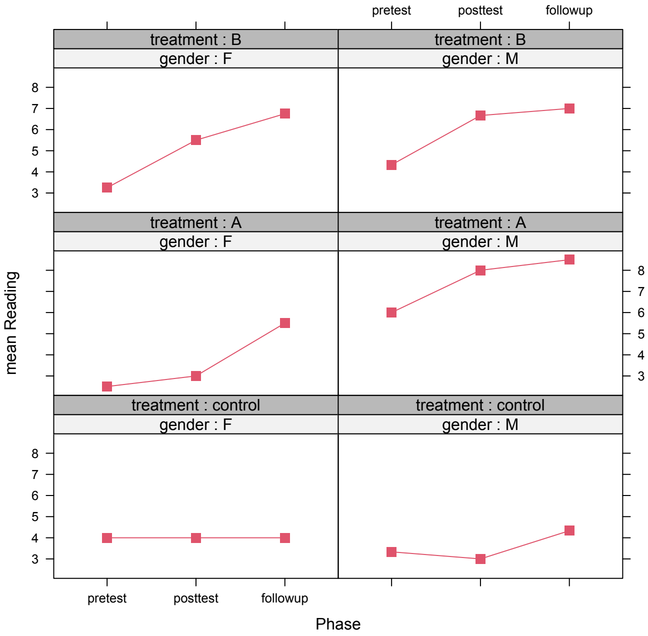
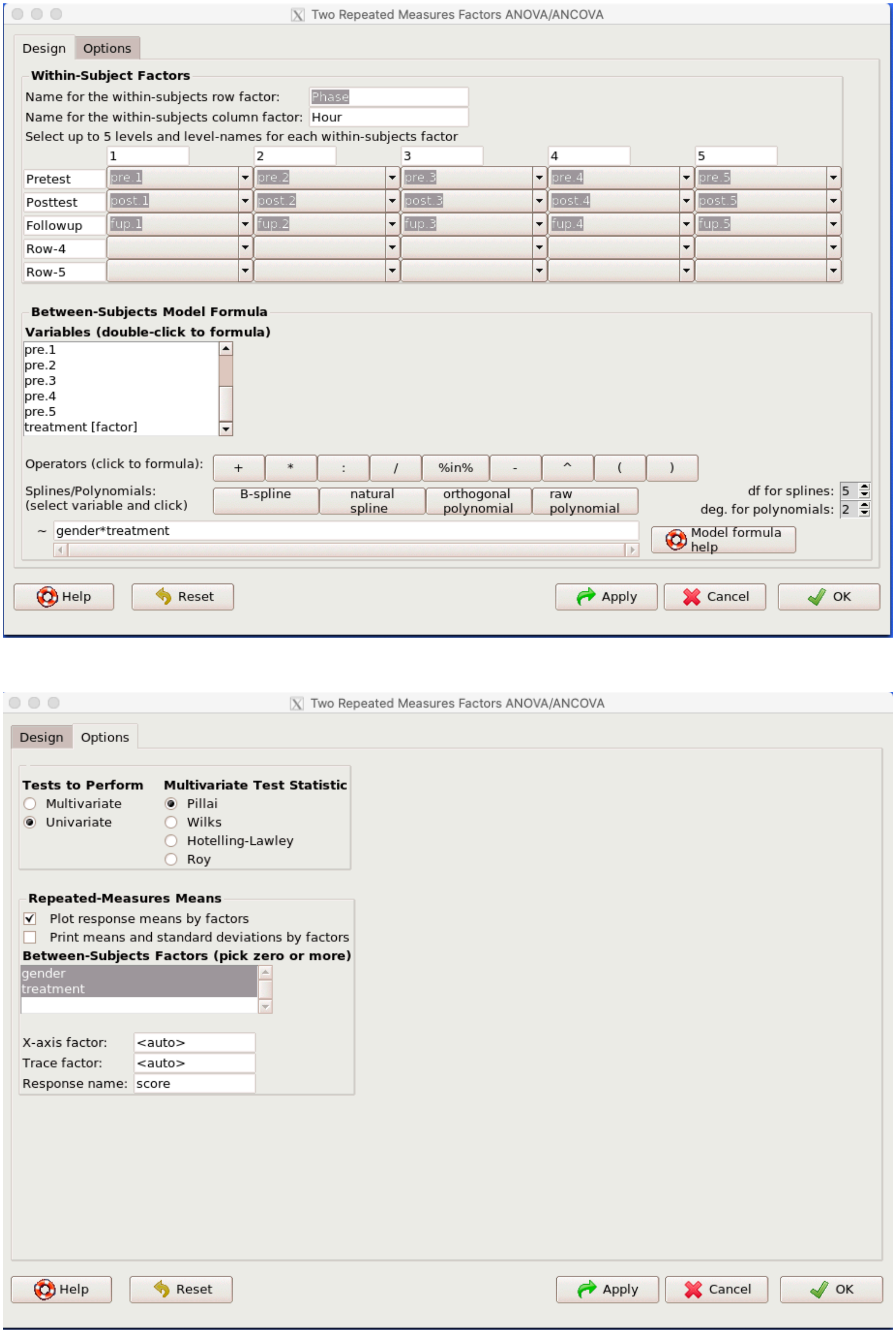


Figure 4: Graph of cell means for the repeated-measures ANOVA of the O'Brien-Kaiser data, with one within-subjects factor and two between-subjects factors.

1.2.2 Two Repeated-Measures Factors

As mentioned, the natural analysis of the O'Brien–Kaiser data uses two repeated measures factors, for phase and hour. Selecting *Statistics > Means > Two-factor repeated-measures ANOVA/ANCOVA* from the R Commander menus brings up the dialog box shown in Figure 5. The dialog box is similar to that for one repeated-measures factor, except that the responses in the *Design* tab are arrayed in a two-way table, with rows corresponding to one within-subjects factor (here, `Phase`) and columns to the other (`Hour`); the dialog accommodates up to five levels of each within-subjects factor. In the options tab, I select univariate tests for variety (rather than the default multivariate tests), check the box for plotting cell means, but leave the box for printing means and standard deviations unchecked. The resulting output for the repeated-measures ANOVA is displayed in Figures 6 and 7, and the graph of means is displayed in Figure 8.



Within-Subject Factors

Name for the within-subjects row factor: Phase

Name for the within-subjects column factor: Hour

Select up to 5 levels and level-names for each within-subjects factor

	1	2	3	4	5
Pretest	pre.1	pre.2	pre.3	pre.4	pre.5
Posttest	post.1	post.2	post.3	post.4	post.5
Followup	fup.1	fup.2	fup.3	fup.4	fup.5
Row-4					
Row-5					

Between-Subjects Model Formula

Variables (double-click to formula)

pre.1
pre.2
pre.3
pre.4
pre.5
treatment [factor]

Operators (click to formula):

Splines/Polynomials:
(select variable and click)

df for splines: 5

deg. for polynomials: 2

~ gender*treatment

Tests to Perform

○ Multivariate
● Univariate

Multivariate Test Statistic

● Pillai
○ Wilks
○ Hotelling-Lawley
○ Roy

Repeated-Measures Means

☑ Plot response means by factors
☐ Print means and standard deviations by factors

Between-Subjects Factors (pick zero or more)

gender
treatment

X-axis factor: <auto>

Trace factor: <auto>

Response name: score

Figure 5: The *Two Repeated-Measures Factor ANOVA/ANCOVA* dialog box, showing the *Design* tab (top) and *Options* tab (bottom).

```
Univariate Type II Repeated-Measures ANOVA Assuming Sphericity

                           Sum Sq num Df Error SS den Df  F value          Pr(>F)    
(Intercept)                7260.0      1  228.056     10 318.3435 0.000000006532 ***
gender                       58.3      1  228.056     10   2.5558       0.140974    
treatment                   211.3      2  228.056     10   4.6323       0.037687 *  
gender:treatment            130.2      2  228.056     10   2.8555       0.104469    
Phase                       167.5      2   80.278     20  20.8651 0.000012744708 ***
gender:Phase                  1.7      2   80.278     20   0.2078       0.814130    
treatment:Phase              78.7      4   80.278     20   4.8997       0.006426 ** 
gender:treatment:Phase       10.2      4   80.278     20   0.6366       0.642369    
Hour                        106.3      4   62.500     40  17.0067 0.000000031911 ***
gender:Hour                   2.6      4   62.500     40   0.4094       0.800772    
treatment:Hour                1.2      8   62.500     40   0.0929       0.999257    
gender:treatment:Hour         7.8      8   62.500     40   0.6204       0.755484    
Phase:Hour                   11.1      8   96.167     80   1.1525       0.338317    
gender:Phase:Hour             6.6      8   96.167     80   0.6900       0.699124    
treatment:Phase:Hour          6.3     16   96.167     80   0.3256       0.992814    
gender:treatment:Phase:Hour  14.2     16   96.167     80   0.7359       0.749562    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 


Mauchly Tests for Sphericity

                           Test statistic p-value
Phase                             0.74927 0.27282
gender:Phase                      0.74927 0.27282
treatment:Phase                   0.74927 0.27282
gender:treatment:Phase            0.74927 0.27282
Hour                              0.06607 0.00760
gender:Hour                       0.06607 0.00760
treatment:Hour                    0.06607 0.00760
gender:treatment:Hour             0.06607 0.00760
Phase:Hour                        0.00478 0.44939
gender:Phase:Hour                 0.00478 0.44939
treatment:Phase:Hour              0.00478 0.44939
gender:treatment:Phase:Hour       0.00478 0.44939
```

Figure 6: Repeated-measures ANOVA with two within-subjects factor for the O'Brien-Kaiser data: Univariate ANOVA table and sphericity tests.

```
Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                             GG eps Pr(>F[GG])
Phase                       0.79953 0.00007323 ***
gender:Phase                0.79953    0.76616
treatment:Phase             0.79953    0.01223 *
gender:treatment:Phase      0.79953    0.61162
Hour                        0.46028 0.00008741 ***
gender:Hour                 0.46028    0.65346
treatment:Hour              0.46028    0.97879
gender:treatment:Hour       0.46028    0.64136
Phase:Hour                  0.44950    0.34573
gender:Phase:Hour           0.44950    0.58903
treatment:Phase:Hour        0.44950    0.94019
gender:treatment:Phase:Hour 0.44950    0.64634
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                               HF eps    Pr(>F[HF])
Phase                       0.9278594 0.00002387543
gender:Phase                0.9278594 0.79844953605
treatment:Phase             0.9278594 0.00808976545
gender:treatment:Phase      0.9278594 0.63199753131
Hour                        0.5592802 0.00002014357
gender:Hour                 0.5592802 0.69115211546
treatment:Hour              0.5592802 0.98877159749
gender:treatment:Hour       0.5592802 0.66929761447
Phase:Hour                  0.7330608 0.34404597799
gender:Phase:Hour           0.7330608 0.65523821077
treatment:Phase:Hour        0.7330608 0.98047311423
gender:treatment:Phase:Hour 0.7330608 0.70801216162
```

Figure 7: Repeated-measures ANOVA with two within-subjects factor for the O'Brien-Kaiser data: Tests corrected for departures from sphericity.

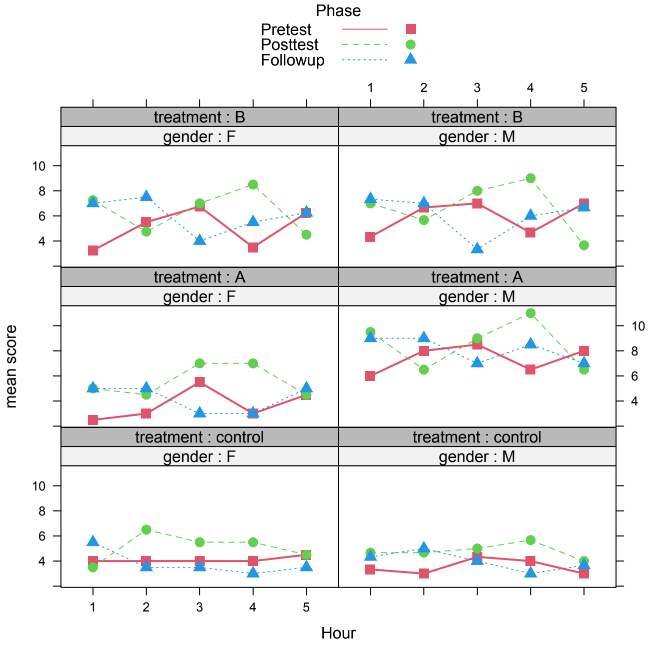
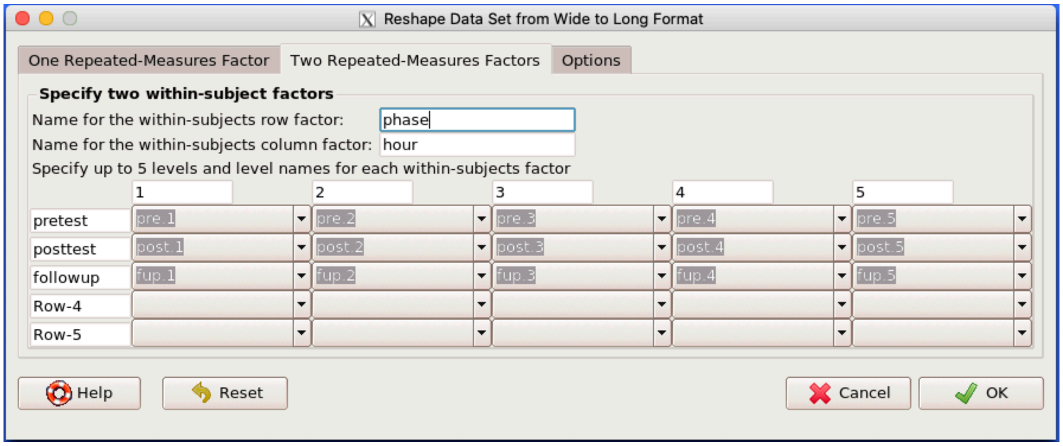


Figure 8: Graph of cell means for the repeated-measures ANOVA of the O'Brien-Kaiser data, with two within-subjects factor and two between-subjects factors.

1.2.3 Dialogs for Reshaping Data

As I explained, the new repeated-measures ANOVA dialogs require data in wide form, with all repeated measures for a subject in a single row of the data set, while the mixed-effects models dialogs require data in long form, with multiple rows for each subject. Two new menu items in the R Commander *Data > Active data set* menu lead to dialogs for *reshaping* data between these two forms: *Reshape data set from long to wide format* and *Reshape data set from wide to long format.*

Suppose, for example, that we wish to reshape the O'Brien-Kaiser data set from wide to long form so that we can analyze the data with a linear mixed model. Figure 9 shows the completed *Reshape Data Set from Wide to Long Format* dialog. The dialog has alternative tabs for one and two repeated-measures factors (the latter is shown at the top of the figure), along with an *Options* tab (at the bottom of the figure). The variables representing repeated measures are specified much as in the repeated-measures ANOVA dialogs. The *Options* tab allows us to specify a name for the long data set, a name for the response variable, and so on. Figure 10 shows the top of the resulting long form of the data set, as it appears in the R Commander data viewer.


Reshape Data Set from Wide to Long Format
One Repeated-Measures Factor
Two Repeated-Measures Factors
Options
Help
Reset
Cancel
OK

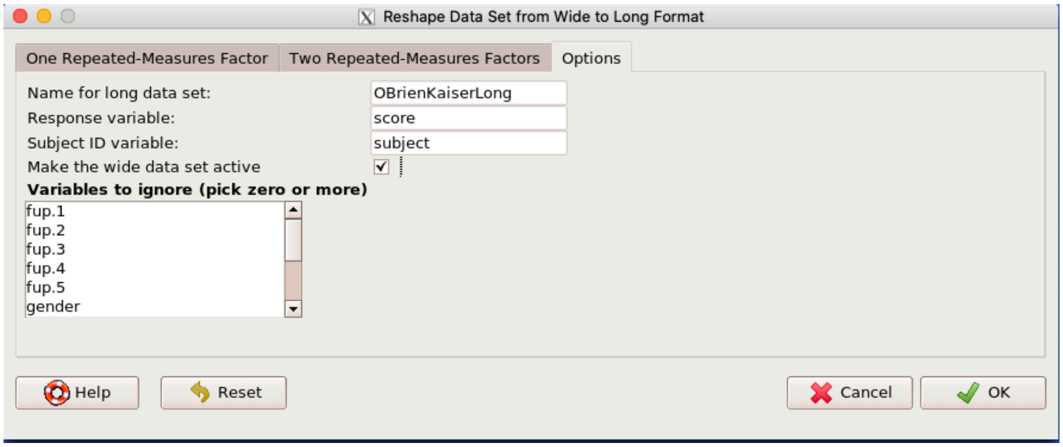
Specify two within-subject factors

Name for the within-subjects row factor: phase

Name for the within-subjects column factor: hour

Specify up to 5 levels and level names for each within-subjects factor

	1	2	3	4	5
pretest	pre.1	pre.2	pre.3	pre.4	pre.5
posttest	post.1	post.2	post.3	post.4	post.5
followup	fup.1	fup.2	fup.3	fup.4	fup.5
Row-4					
Row-5					


Reshape Data Set from Wide to Long Format
One Repeated-Measures Factor
Two Repeated-Measures Factors
Options
Help
Reset
Cancel
OK

Name for long data set: OBrienKaiserLong

Response variable: score

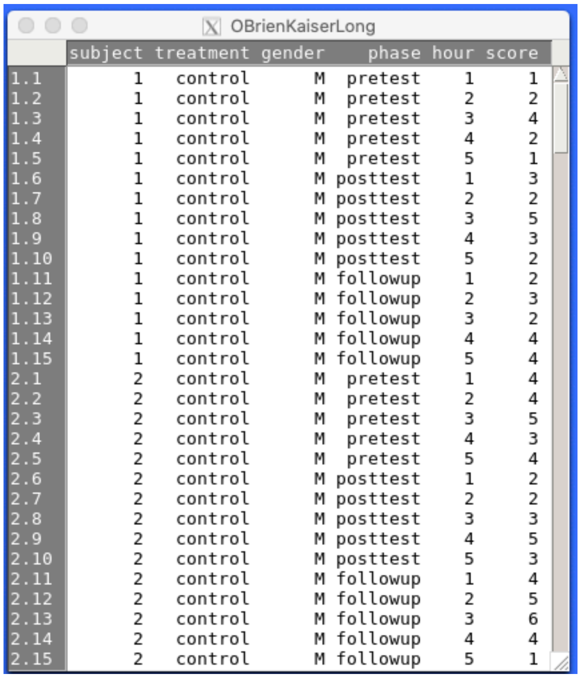
Subject ID variable: subject

Make the wide data set active ☑

Variables to ignore (pick zero or more)

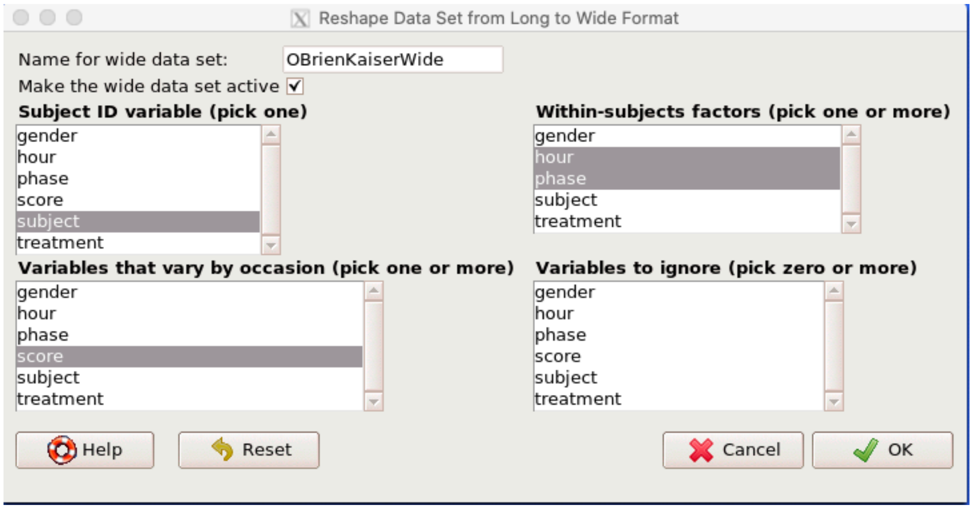
fup.1
fup.2
fup.3
fup.4
fup.5
gender

Figure 9: The dialog for reshaping the O'Brien–Kaiser data from wide to long format.



	subject	treatment	gender	phase	hour	score
1.1	1	control	M	pretest	1	1
1.2	1	control	M	pretest	2	2
1.3	1	control	M	pretest	3	4
1.4	1	control	M	pretest	4	2
1.5	1	control	M	pretest	5	1
1.6	1	control	M	posttest	1	3
1.7	1	control	M	posttest	2	2
1.8	1	control	M	posttest	3	5
1.9	1	control	M	posttest	4	3
1.10	1	control	M	posttest	5	2
1.11	1	control	M	followup	1	2
1.12	1	control	M	followup	2	3
1.13	1	control	M	followup	3	2
1.14	1	control	M	followup	4	4
1.15	1	control	M	followup	5	4
2.1	2	control	M	pretest	1	4
2.2	2	control	M	pretest	2	4
2.3	2	control	M	pretest	3	5
2.4	2	control	M	pretest	4	3
2.5	2	control	M	pretest	5	4
2.6	2	control	M	posttest	1	2
2.7	2	control	M	posttest	2	2
2.8	2	control	M	posttest	3	3
2.9	2	control	M	posttest	4	5
2.10	2	control	M	posttest	5	3
2.11	2	control	M	followup	1	4
2.12	2	control	M	followup	2	5
2.13	2	control	M	followup	3	6
2.14	2	control	M	followup	4	4
2.15	2	control	M	followup	5	1

Figure 10: The first few rows of the resulting long form of the O'Brien–Kaiser data set.



Reshape Data Set from Long to Wide Format

Name for wide data set: OBrienKaiserWide

Make the wide data set active ☑

Subject ID variable (pick one)

gender
hour
phase
score
subject
treatment

Within-subjects factors (pick one or more)

gender
hour
phase
subject
treatment

Variables that vary by occasion (pick one or more)

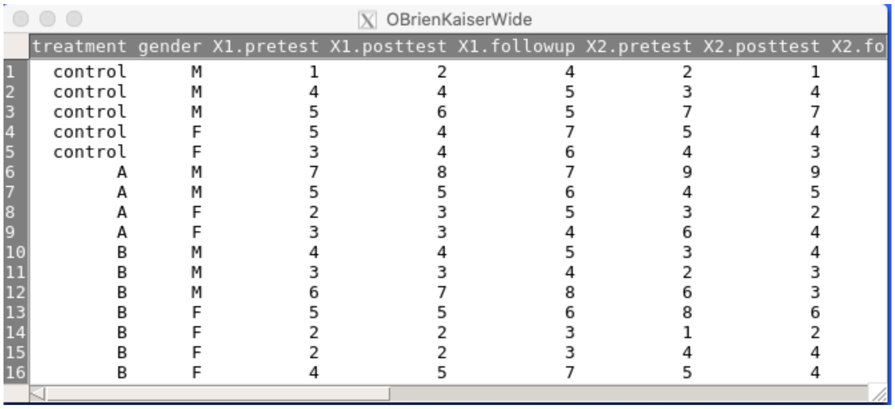
gender
hour
phase
score
subject
treatment

Variables to ignore (pick zero or more)

gender
hour
phase
score
subject
treatment

Figure 11: The dialog for reshaping the O'Brien–Kaiser data from long back to wide format.

The dialog for reshaping from long (back) to wide format, shown in Figure 11, is a bit simpler. We select the ID variable (here, `subject`), the variable or variables that vary by occasion (here just the response `score`), and the within-subjects factors (`hour` and `phase`). The resulting data set recovers the original wide form of the O'Brien–Kaiser data; all 16 rows, corresponding to the 16 subjects, and the first few columns are shown in Figure 12. Notice that "`X`"s are appended to the levels of the long-format within-subjects factor `hour`, which are numerals (`"1"`, `"2"`, . . . , `"5"`), to create legal variable names in the wide form of the data (`X1.pretest`, etc).



	treatment	gender	X1.pretest	X1.posttest	X1.followup	X2.pretest	X2.posttest	X2.fo
1	control	M	1	2	4	2	1	
2	control	M	4	4	5	3	4	
3	control	M	5	6	5	7	7	
4	control	F	5	4	7	5	4	
5	control	F	3	4	6	4	3	
6	A	M	7	8	7	9	9	
7	A	M	5	5	6	4	5	
8	A	F	2	3	5	3	2	
9	A	F	3	3	4	6	4	
10	B	M	4	4	5	3	4	
11	B	M	3	3	4	2	3	
12	B	M	6	7	8	6	3	
13	B	F	5	5	6	8	6	
14	B	F	2	2	3	1	2	
15	B	F	2	2	3	4	4	
16	B	F	4	5	7	5	4	

Figure 12: The first few columns of the restored wide form of the O'Brien–Kaiser data set.

2 Rcmdr Version 2.6-0 (September 2019)

2.1 The Official R Commander Hex Sticker

The **Rcmdr** package now has a hex sticker, shown in Figure 13! It can be displayed from the R Commander menus via *Help* > R Commander hex sticker.

2.2 New Dialogs in the *Statistics* > *Fit models* Menu for Linear and Generalized Linear Mixed Models

Figure 15 shows the *Linear Mixed Model* dialog completed to fit a LMM to the `Blackmore` data. The contents of the box for the left-hand side of the model aren't entirely visible, and the box contains `log2(exercise + 5/60)`. Preliminary examination of the data, which I invite the reader to repeat, shows that the `exercise` values are positively skewed and that some values are 0. I added 5 minutes to the exercise values and then took logs (to the base 2) to correct the skew. An alternative would be to use a modified Box-Cox transformation capable of handling 0 values (as in Fox and Weisberg, 2019).

The right-hand side of the model also requires a bit of explanation: The formula fits fixed effects for `group`, `age`, and their interaction, and random `age` slopes and (implied) intercepts that vary by subjects. Subtracting 8 from `age` locates the regression intercept at the start of the study; the identity (or "inhibit") function `I()` is required so that the minus operator (`-`) is interpreted as subtraction. The resulting output is shown in Figure 16.

Selecting *Models* > *Hypothesis tests* > *ANOVA table* brings up the same dialog box as in Figure 7.20 in the text, and selecting the default type-II tests produces Wald chisquare tests for the fixed effects, as shown in Figure 17. There is, therefore, very strong evidence for an interaction of `group` and `age`.

The coefficients of the model are interpretable: The intercept is the average value of log base 2 `exercise` at `age` 8 for the baseline `"control"` group; the coefficient `group[T.patient]` is the difference in average log base 2 `exercise` at `age` 8 between the `"patient"` and `"control"` groups; the coefficient `I(age - 8)` is the average `age` slope in the baseline `"control"` group; and the `group[T.patient]:I(age - 8)` coefficient is the difference in average slopes between the `"patient"` and `"control"` groups. Thus, average `exercise` is similar at the start of the study for the two groups but rises more rapidly with `age` for the `"patient"` group.

We can visualize the fixed effects by selecting *Models* > *Graphs* > *Predictor effect plots*, producing the dialog box in Figure 18. I completed the dialog as shown and clicked *OK*, producing the graph in Figure 19. Because the vertical axis is in the log base 2 scale, -1 represents $2^{-1} = 1/2$ hour, 0 represents $2^0 = 1$ hour, 1 represents $2^1 = 2$ hours, and 2 represents $2^2 = 4$ hours of exercise per week (disregarding the 5 minutes added to the response variable).

I invite the reader to explore the other active items in the *Models* menu, including those for numerical and graphical diagnostics.

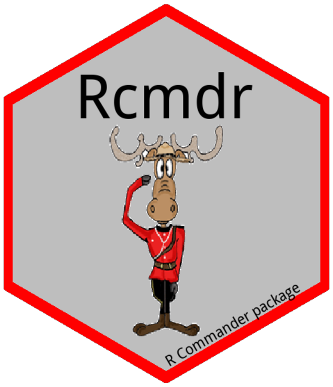
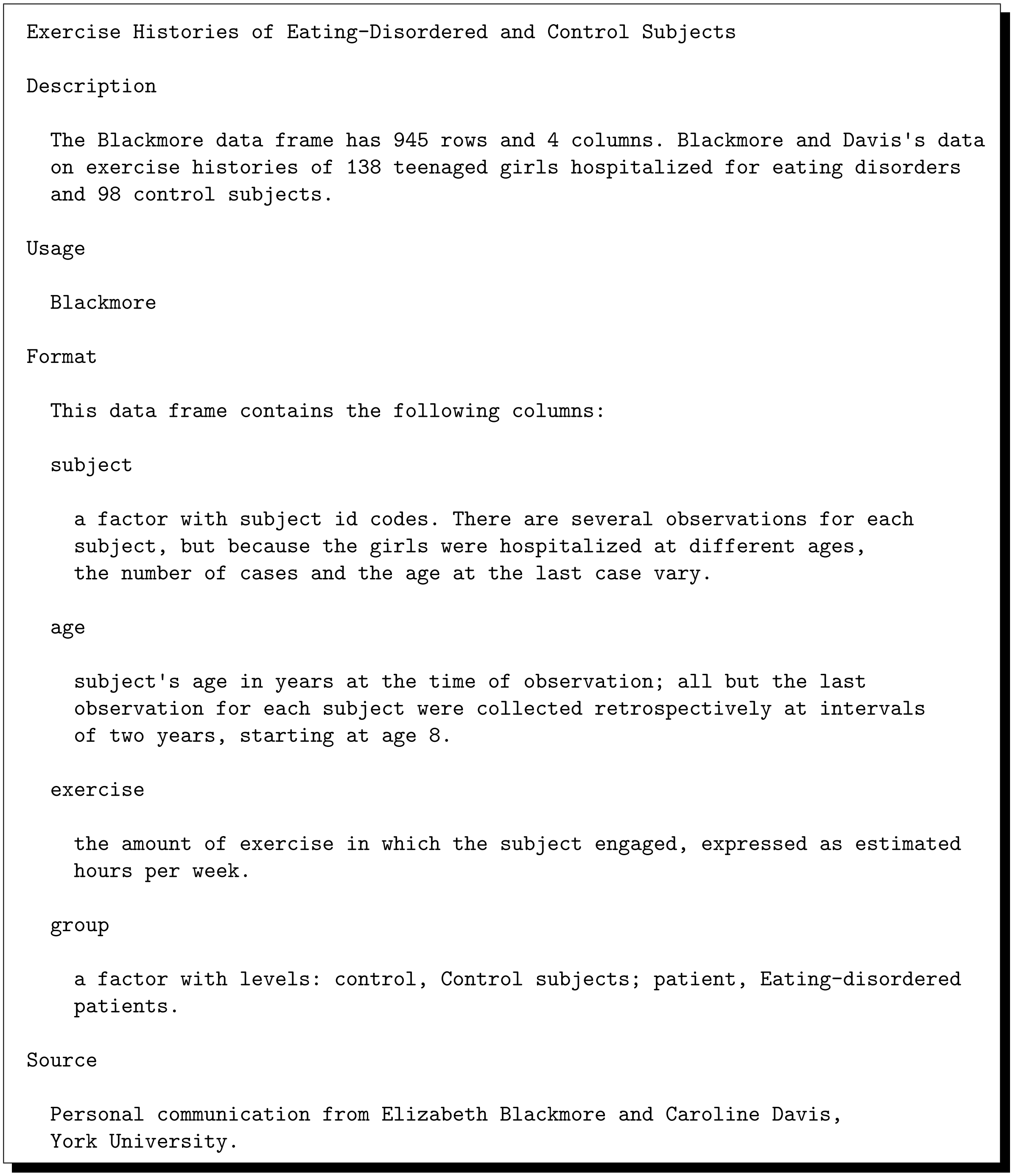
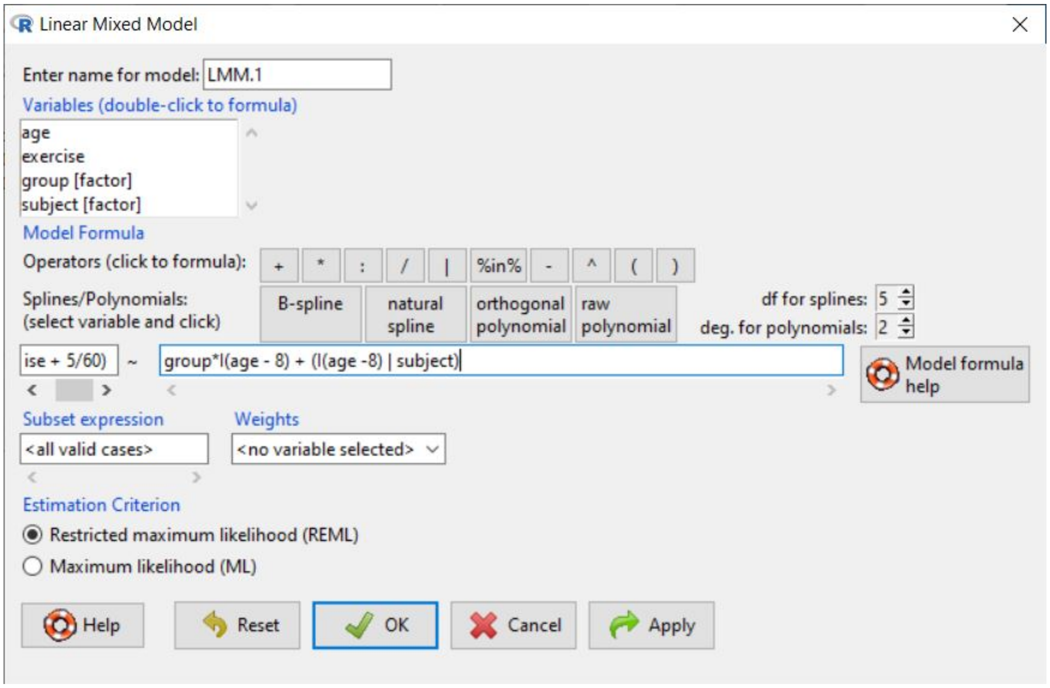


Figure 13: The official R Commander hex sticker.



```
Exercise Histories of Eating-Disordered and Control Subjects

Description

  The Blackmore data frame has 945 rows and 4 columns. Blackmore and Davis's data
  on exercise histories of 138 teenaged girls hospitalized for eating disorders
  and 98 control subjects.

Usage

  Blackmore

Format

  This data frame contains the following columns:

  subject

    a factor with subject id codes. There are several observations for each
    subject, but because the girls were hospitalized at different ages,
    the number of cases and the age at the last case vary.

  age

    subject's age in years at the time of observation; all but the last
    observation for each subject were collected retrospectively at intervals
    of two years, starting at age 8.

  exercise

    the amount of exercise in which the subject engaged, expressed as estimated
    hours per week.

  group

    a factor with levels: control, Control subjects; patient, Eating-disordered
    patients.

Source

  Personal communication from Elizabeth Blackmore and Caroline Davis,
  York University.
```

Figure 14: Codebook for the Blackmore data set.



Enter name for model: LMM.1

df for splines: 5

deg. for polynomials: 2

ise + 5/60) ~ group*I(age - 8) + (I(age -8) | subject)

Subset expression

<all valid cases>

Weights

<no variable selected>

● Restricted maximum likelihood (REML)

○ Maximum likelihood (ML)

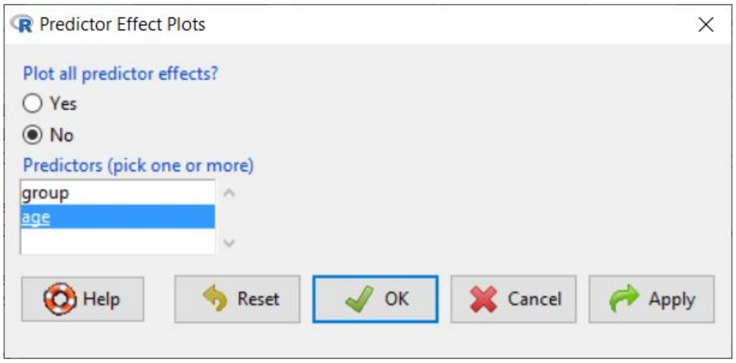
Figure 15: The *Linear Mixed Model* dialog.

```
> LMM.1 <- lmer(log2(exercise + 5/60) ~ group * I(age - 8) + (I(age - 8) | subject),
+     data = Blackmore, REML = TRUE)

> summary(LMM.1)
Linear mixed model fit by REML ['lmerMod']
Formula: log2(exercise + 5/60) ~ group * I(age - 8) + (I(age - 8) | subject)
   Data: Blackmore

REML criterion at convergence: 3614.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7349 -0.4245  0.1228  0.5280  2.6362

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subject  (Intercept) 2.08384  1.4436
          I(age - 8)  0.02716  0.1648   -0.28
 Residual             1.54775  1.2441
Number of obs: 945, groups:  subject, 231

Fixed effects:
                            Estimate Std. Error t value
(Intercept)                 -0.27602    0.18237  -1.514
group[T.patient]            -0.35400    0.23529  -1.505
I(age - 8)                   0.06402    0.03136   2.041
group[T.patient]:I(age - 8)  0.23986    0.03941   6.087

Correlation of Fixed Effects:
            (Intr) gr[T.] I(g-8)
grp[T.ptnt] -0.775
I(age - 8)  -0.489  0.379
g[T.]:I(-8)  0.389 -0.489 -0.796
```

Figure 16: Output from the linear mixed model fit to the `Blackmore` data.

```
> Anova(LMM.1, type = "II")
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: log2(exercise + 5/60)
                     Chisq Df     Pr(>Chisq)
group               2.8564  1        0.09101 .
I(age - 8)        129.2811  1      < 2.2e-16 ***
group:I(age - 8)   37.0474  1 0.000000001153 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 17: Type-II tests for the fixed effects in the linear mixed model fit to the `Blackmore` data.



Plot all predictor effects?

○ Yes

● No

Predictors (pick one or more)

group

age

Figure 18: The *Predictor Effect Plots* dialog for the mixed model fit to the `Blackmore` data.

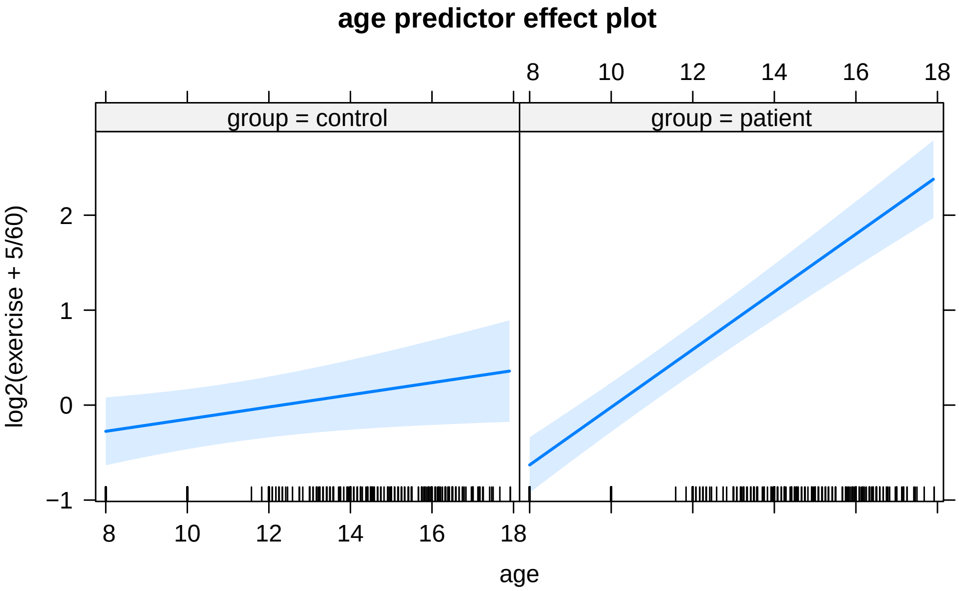


Figure 19: The predictor effect plot for `age` for the mixed model fit to the `Blackmore` data, showing the `age` by `group` interaction.

3 Rcmdr Version 2.5-0 (August 2018)

3.1 New Dialogs in the *Models* Menu

There are two new regression-diagnostics dialogs: *Models > Numerical diagnostics > Response transformation* and *Models > Graphs > Influence index plot.*

For an example of the former, I fit a regression model to the `UN` data in the **carData** package via the *Statistics > Fit models > Linear model* dialog, with the formula `LinearModel.1 <- lm(infantMortality ~ ppgdp + group, data=UN)`. The data are for the nations of the world, the response variable is the infant-mortality rate of each nation, and the explanatory variables are the nation's per-capita GDP (a numeric variable) and group of nations (a factor).

Then selecting *Models > Numerical diagnostics > Response transformation* brings up the dialog box in Figure 20, for Box-Cox maximum-likelihood-like selection of a power transformation of the response variable in a regression model. The default *Transformation Family* selection, *Box-Cox*, is appropriate for a strictly positive response such as `infantMortality`. The other two families are generalizations of the Box-Cox family that are appropriate when the response takes on zero or negative values. The output (including the output from the original regression, where `"oecd"` is the reference level for the factor `group`), shown in Figure 21, suggests the log transformation of `infantMortality` (the "zeroth" power); for more information on response transformations, see Fox and Weisberg (2019, Sec. 8.4.1).

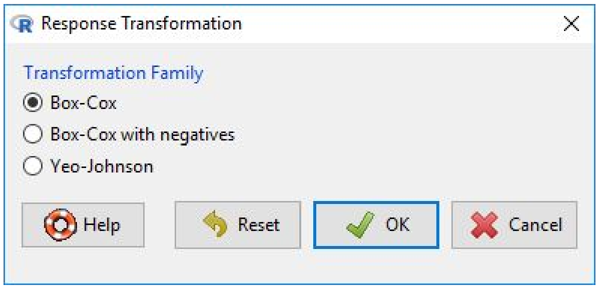
The dialog produced by *Models > Graphs > Influence index plot* appears in Figure 22. The dialog creates a compact graph of index plots of several diagnostic statistics, which are selected by check boxes. There is also provision for either automatic or interactive identification of noteworthy cases. The selections in the dialog are the defaults and they produce the graph shown in Figure 23.

The *Models > Graphs* menu also acquires a new *Predictor effect plots* menu item, leading to the dialog box illustrated in Figure 24. I've taken the defaults in the dialog except for checking the *Plot partial residuals* box. The resulting predictor effect plot (see Fox and Weisberg, 2019) in the left panel of Figure 25 suggests an incorrectly modeled nonlinear partial relationship between `infantMortality` and `ppgdp`.

3.2 Variable Transformation Dialogs

In addition to the *Response Transformation* dialog described above, there are two new dialogs for the unconditional transformation of a numeric variable.

Graphs > Symmetry boxplot brings up the dialog in Figure 26. I selected the variable `infantMortality` from the list box and left the other selections at their defaults. The *Box-Cox*



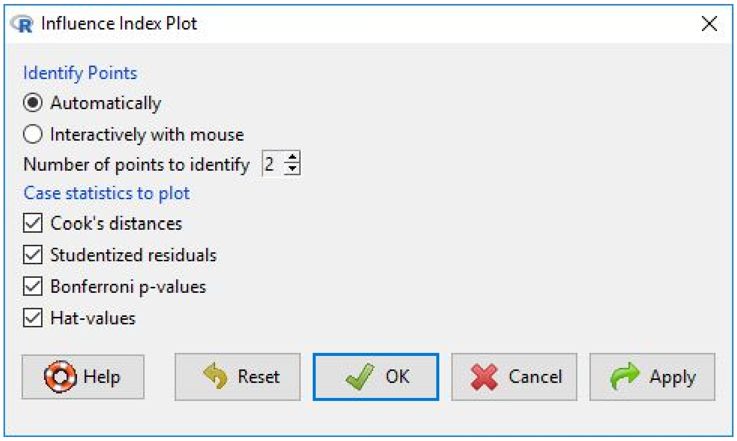
Transformation Family

● Box-Cox
○ Box-Cox with negatives
○ Yeo-Johnson

Figure 20: The *Response Transformation* dialog.

```
> data(UN)

> LinearModel.1 <- lm(infantMortality ~ ppgdp + group, data=UN)

> summary(LinearModel.1)

Call:
lm(formula = infantMortality ~ ppgdp + group, data = UN)

Residuals:
    Min      1Q  Median      3Q     Max 
-50.888  -9.711  -2.561   7.165  98.660 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     21.9030426  5.1550507   4.249 3.37e-05 ***
ppgdp           -0.0004505  0.0001038  -4.341 2.31e-05 ***
group[T.other]   4.1968188  4.7734820   0.879     0.38    
group[T.africa] 44.4705476  5.6052207   7.934 1.82e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.65 on 189 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.5998,    Adjusted R-squared:  0.5935 
F-statistic: 94.42 on 3 and 189 DF,  p-value: < 2.2e-16

> summary(powerTransform(LinearModel.1, family="bcPower"))
bcPower Transformation to Normality 
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
Y1   -0.0293           0      -0.1514       0.0929

Likelihood ratio test that transformation parameter is equal to 0
 (log transformation)
                           LRT df    pval
LR test, lambda = (0) 0.2196112  1 0.63934

Likelihood ratio test that no transformation is needed
                          LRT df       pval
LR test, lambda = (1) 215.7756  1 < 2.22e-16
```

Figure 21: Output from the regression of infant mortality on per-capita GDP and group, followed by the output produced by the *Response Transformation* dialog.



Influence Index Plot

Identify Points

● Automatically

○ Interactively with mouse

Number of points to identify 2

Case statistics to plot

☑ Cook's distances

☑ Studentized residuals

☑ Bonferroni p-values

☑ Hat-values

Figure 22: The *Influence Index Plot* dialog.

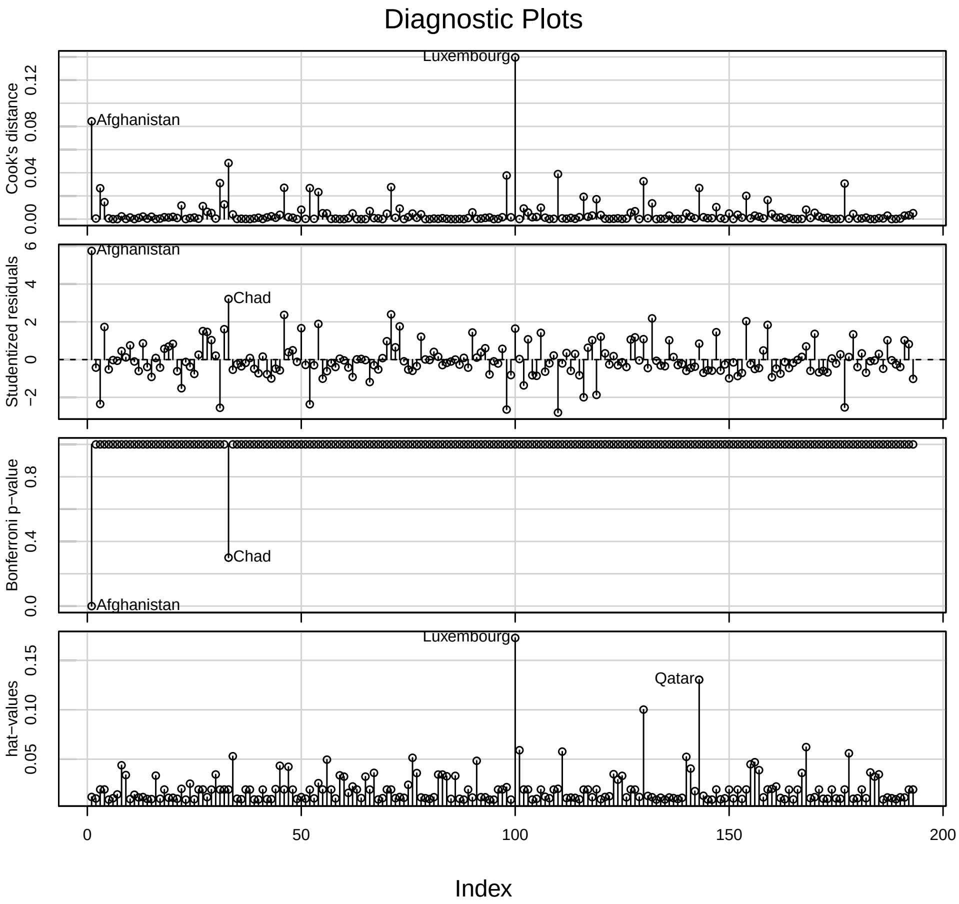
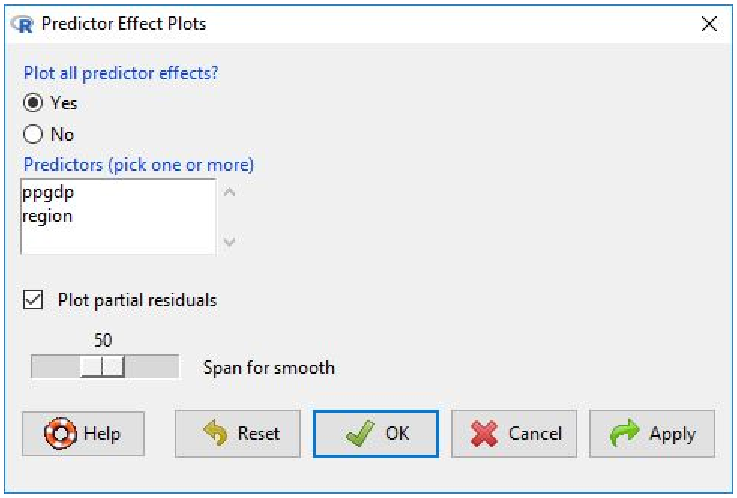


Figure 23: Graph produced by the *Influence Index Plot* dialog.



Predictor Effect Plots

Plot all predictor effects?

● Yes

○ No

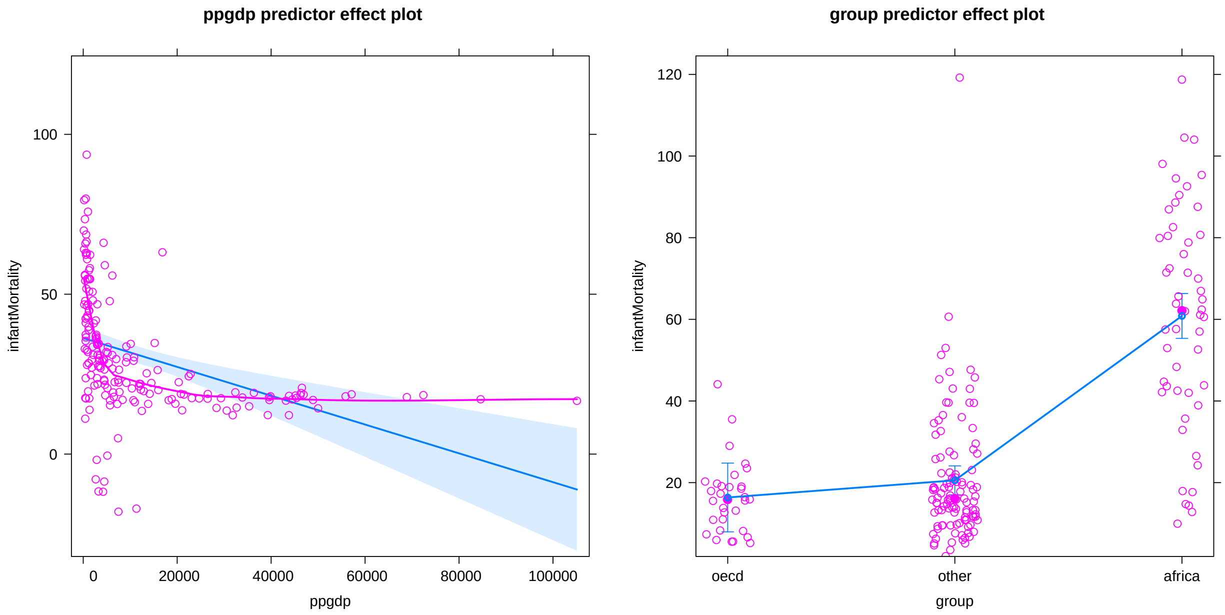
Predictors (pick one or more)

ppgdp
region

☑ Plot partial residuals

50 Span for smooth

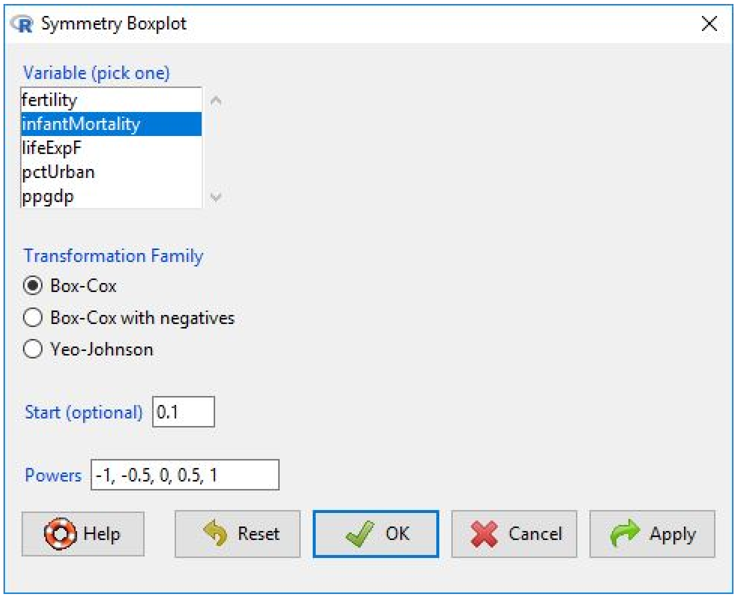
Figure 24: The *Predictor Effect Plots* dialog.



ppgdp predictor effect plot

group predictor effect plot

Figure 25: Predictor effect plots for the regression model fit to the `UN` infant-mortality data, including partial residuals.



Symmetry Boxplot

Variable (pick one)

fertility
infantMortality
lifeExpF
pctUrban
ppgdp

Transformation Family

● Box-Cox
○ Box-Cox with negatives
○ Yeo-Johnson

Start (optional) 0.1

Powers -1, -0.5, 0, 0.5, 1

Figure 26: The *Symmetry Boxplot* dialog.

power-transformation family is appropriate because `infantMortality` has no zero or negative values. An optional *Start* value, not specified in this example, is added to the data prior to Box-Cox or Box-Cox-with-negatives transformation. The resulting graph, shown in Figure 27, suggests that the log transformation makes the distribution of `infantMortality` symmetric.

Statistics > *Summaries* > *Transform toward normality* brings up the dialog box in Figure 28, which estimates power transformations towards normality or multivariate normality analytically by the unconditional or conditional Box-Cox method. One or more variables are selected in the *Data* tab: Here I picked `infantMortality` and `ppgdp`. The *Options* tab provides for selection of the transformation family and, optionally, conditioning variables in a linear-model formula; I left all options at their defaults. Clicking the *OK* button produces the output in Figure 29, which suggests the log transformation of both `infantMortality` and `ppgdp`.

3.3 Enhancement to Graphics Dialogs

The dialogs for *Graphs* > *Quantile-comparison plot* and *Graphs* > *Index plot* now both support plotting by groups, producing a separate panel for each group in the first case and values color-coded by group in the second. Usage is straightforward, with group selection through a button on the *Data* tab in each dialog, as implemented in other dialogs, such as the *Histogram* dialog, that previously supported plotting by groups (see Fig. 5.8 in the text).

The *Graphs* > *Density estimate* dialog provides more options than previously and uses an adaptive-kernel density estimator by default.

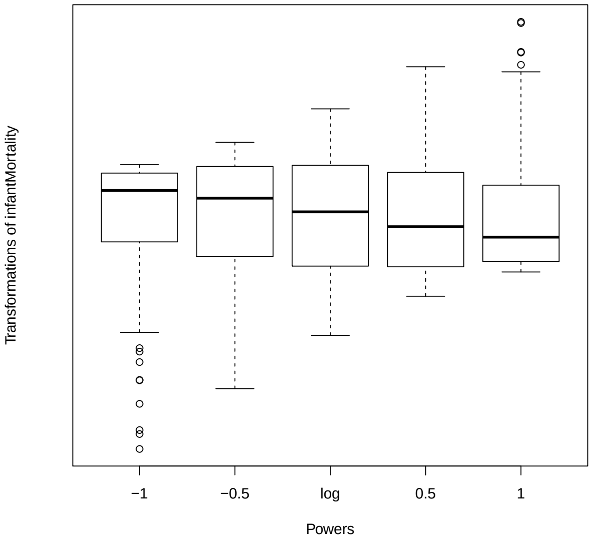
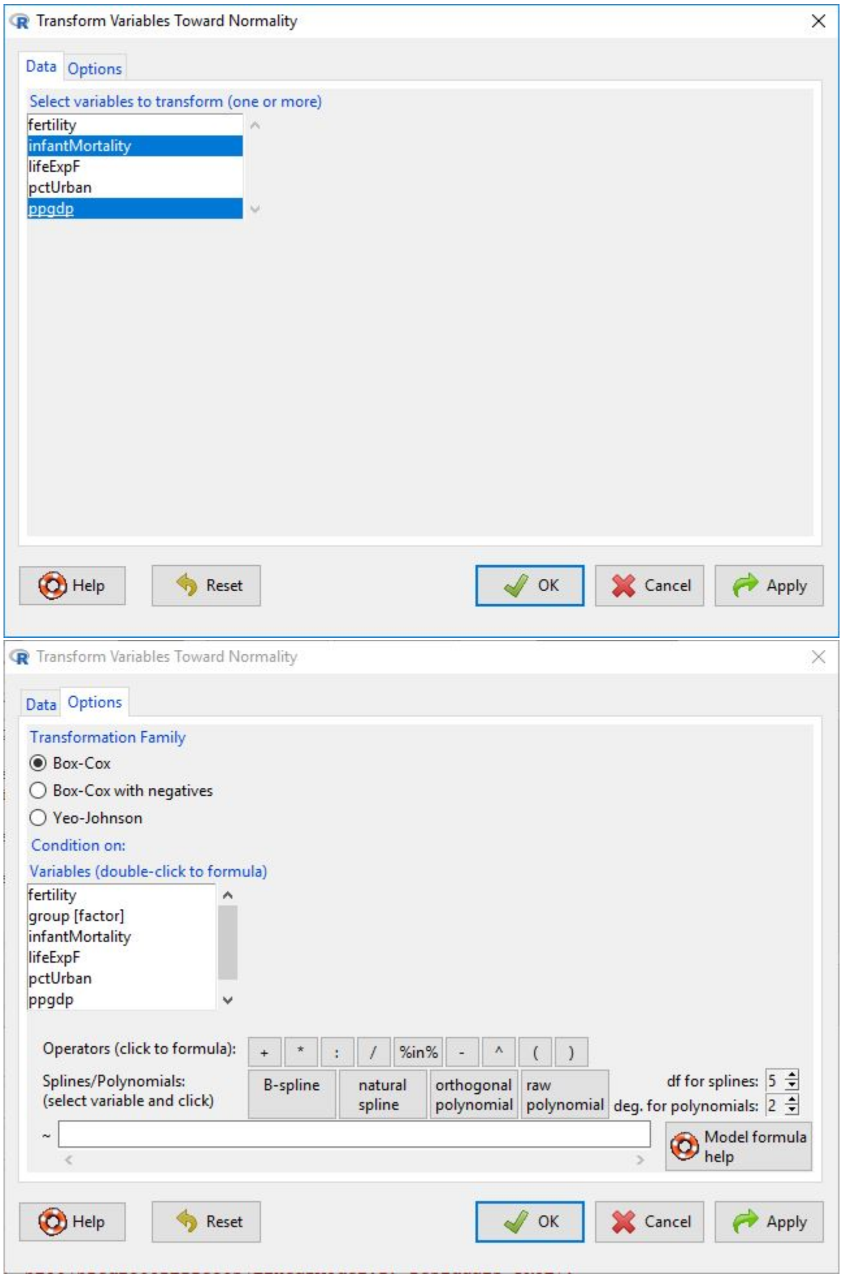


Figure 27: Symmetry boxplots for `infantMortality` in the `UN` data.



Transform Variables Toward Normality

Data | Options

Select variables to transform (one or more)

- fertility
- **infantMortality**
- lifeExpF
- pctUrban
- **ppgdp**

Transform Variables Toward Normality

Data | Options

Transformation Family

● Box-Cox

○ Box-Cox with negatives

○ Yeo-Johnson

Condition on:

Variables (double-click to formula)

- fertility
- group [factor]
- infantMortality
- lifeExpF
- pctUrban
- ppgdp

Operators (click to formula): + * : / %in% - ^ ( )

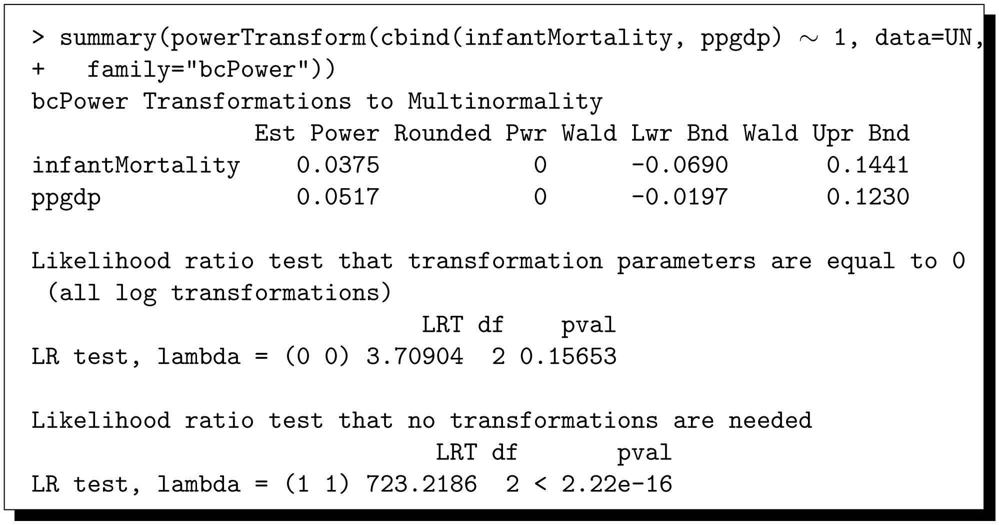
Splines/Polynomials: (select variable and click)

df for splines: 5

deg. for polynomials: 2

~

Figure 28: The *Transform Variables Toward Normality* dialog, *Data* tab (top) and *Options* tab (bottom).



```
> summary(powerTransform(cbind(infantMortality, ppgdp) ~ 1, data=UN,
+   family="bcPower"))
bcPower Transformations to Multinormality
                Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
infantMortality    0.0375           0      -0.0690       0.1441
ppgdp              0.0517           0      -0.0197       0.1230

Likelihood ratio test that transformation parameters are equal to 0
 (all log transformations)
                             LRT df    pval
LR test, lambda = (0 0) 3.70904  2 0.15653

Likelihood ratio test that no transformations are needed
                              LRT df       pval
LR test, lambda = (1 1) 723.2186  2 < 2.22e-16
```

Figure 29: Output produced by the *Transform Variables Toward Normality* dialog, for the unconditional Box-Cox power transformation of `infantMortality` and `ppgdp`.

4 Rcmdr Version 2.4-0 (August 2017)

4.1 Non-Modal R Markdown and knitr Document Editor

The R Commander editor for R Markdown and knitr documents is now a *non-modal* dialog, and so may remain open while you work. I recommend that you open the dialog at the beginning of your R Commander session in the usual manner (e.g., via the key-combination *Control-e* in the R Commander *R Markdown* or *knitr* tab: see Section 3.6.2 in the text).

Commands generated by the R Commander are entered both in the *R Markdown* (or *knitr*) tab and in the document editor. You can type any explanatory text that you like in the editor at any point during the session.

The editor maintains an independent copy of the document. To commit the text in the editor to the *R Markdown* (or *knitr*) tab, press the *Save edits* button in the editor toolbar (see Figure 30), or select *File > Save current edits* from the editor menus. Text in the editor is also saved to the R Commander *R Markdown* (or *knitr*) tab when you generate a report in the document editor or exit from the editor by the *OK* button or via *File > Exit editor saving edits*.

4.2 Suppression of Scientific Notation

I introduced an option to control the degree of suppression of scientific notation in R output. This option can be set with the R `options(Rcmdr=list(scientific.notation=n))` command (where *n* is an integer), or from the R Commander menus, via *Tools > Options*, which brings up the *Commander Options* dialog, in the *Output* tab (see Section 3.9.1 of the text). The larger the value of the `scientific.notation` option, the more fixed-decimal-point notation is preferred to scientific (exponential) notation.

The `scientific.notation` option in the R Commander corresponds to the `scipen` option in R (see `?options`). The initial default is `0` in R and `5` in the R Commander, indicating greater suppression of scientific notation in the R Commander. For example, the output from Duncan's occupational prestige regression (given in Figure 7.2 of the text) appears as in Figure 31, with the new default setting of the `scientific.notation` option. Note that the very small number `2.2e-16` $= 2.2 \times 10^{-16}$ is still given in scientific notation, but that the p-values for the `education` and `income` coefficients are now shown in fixed-decimal notation.

4.3 Dialog for Plotting A Discrete Numeric Variable

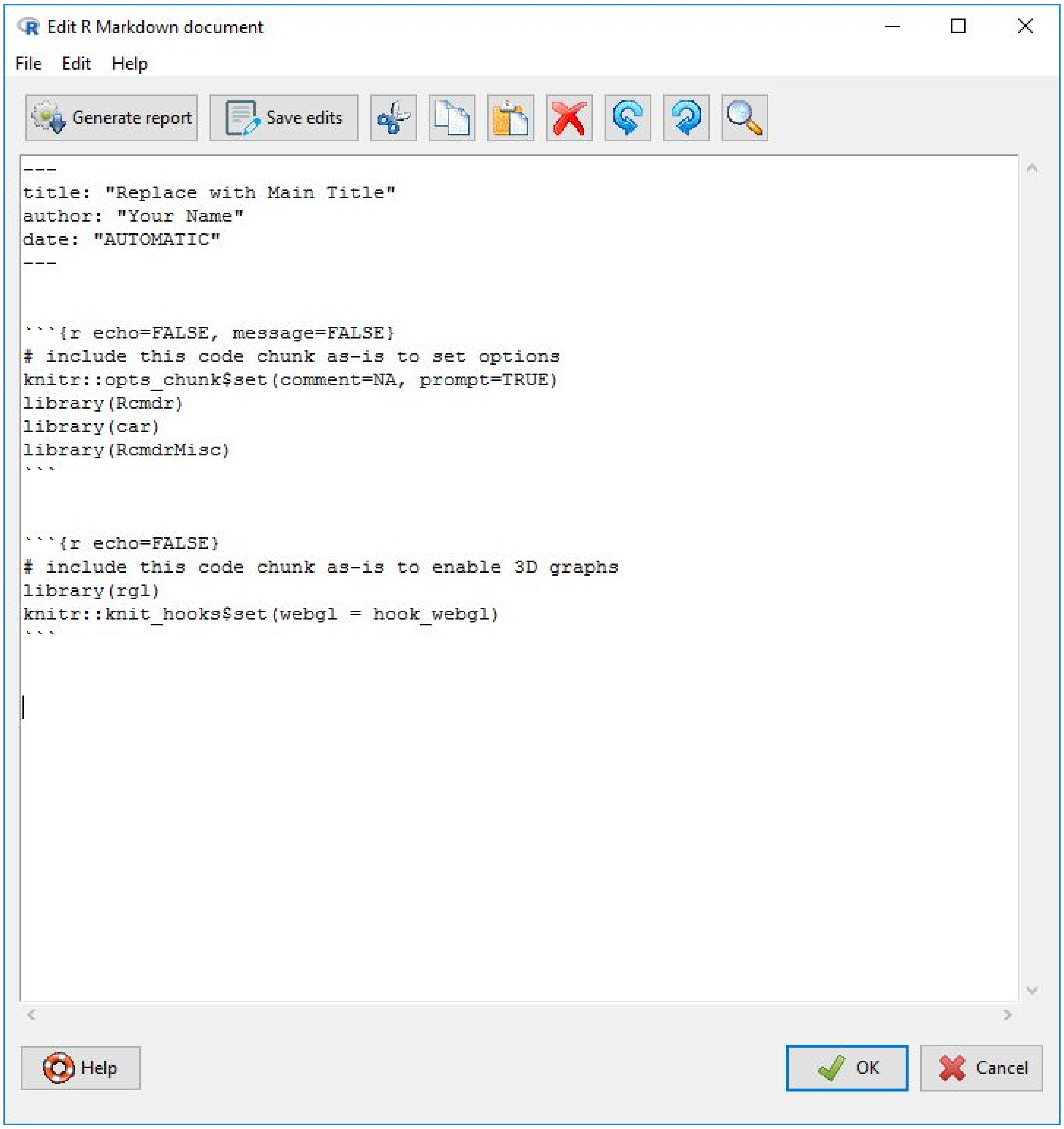
There is a new dialog for plotting the distribution of a discrete numeric variable.

4.4 Improved Handling of Blanks and Quotes in the Data Editor

It is no longer necessary to place double quotes around character strings that contain blanks in the R Commander data editor. Double or single quotes may, however, optionally be placed around *any* character strings entered in the data editor.

4.5 Improved Data Import From Other Statistical Software

The **haven** package (Wickham and Miller, 2016) is now used to import SPSS and Stata data files, and to import SAS `.b7dat` data files.



~~~
---
title: "Replace with Main Title"
author: "Your Name"
date: "AUTOMATIC"
---


```{r echo=FALSE, message=FALSE}
# include this code chunk as-is to set options
knitr::opts_chunk$set(comment=NA, prompt=TRUE)
library(Rcmdr)
library(car)
library(RcmdrMisc)
~~~

```{r echo=FALSE}
# include this code chunk as-is to enable 3D graphs
library(rgl)
knitr::knit_hooks$set(webgl = hook_webgl)
```
```

Figure 30: The R Commander R Markdown document editor, opened at the start of a session.
```

```
> RegModel.1 <- lm(prestige~education+income, data=Duncan)

> summary(RegModel.1)

Call:
lm(formula = prestige ~ education + income, data = Duncan)

Residuals:
    Min      1Q  Median      3Q     Max
-29.538  -6.417   0.655   6.605  34.641

Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept) -6.06466    4.27194  -1.420      0.163
education    0.54583    0.09825   5.555 0.00000173 ***
income       0.59873    0.11967   5.003 0.00001053 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.37 on 42 degrees of freedom
Multiple R-squared:  0.8282,    Adjusted R-squared:   0.82
F-statistic: 101.2 on 2 and 42 DF,  p-value: < 2.2e-16
```

Figure 31: Output from Duncan's occupational prestige regression, illustrating the suppression of scientific notation (cf., Figure 7.2 in the text).

4.6 Improvements to Existing Dialogs

Several existing dialogs have been enhanced, including the dialogs for bar plots, pie charts, testing for the difference between two variances, entering and analyzing a contingency table, bootstrapping, variance-inflation factors, and index plots.

5 Rcmdr Version 2.3-0 (August 2016)

5.1 New Dialog for Bootstrapping Regression Models

There's a new *Bootstrap* dialog, which uses the `Boot` function in the **car** package (which is a simplified front-end to the `boot` function in the standard R **boot** package, Canty and Ripley, 2016; Davison and Hinkley, 1997) to compute bootstrapped confidence intervals for coefficients in linear and generalized linear models. For linear models, the dialog gives a choice between bootstrapping entire cases ("random-x" resampling) and bootstrapping residuals ("fixed-x" resampling). Bootstrapping regression models is described, for example, in Fox (2016, Chapter 2.1) and Weisberg (2014, Section 7.7).

To illustrate bootstrapping, I'll use the Cowles and Davis (1987) logistic regression described in Section 7.4 of the text, where it appears as model `GLM.7` (see Figures 7.11 and 7.12). Having fit the model, selecting *Models > Bootstrap confidence intervals* from the R Commander menus produces the dialog box in Figure 32. Because this is a generalized linear model, the radio buttons for *Case resampling* versus *Residual resampling* don't appear. I retain all of the default selections in the dialog, except for increasing the number of bootstrap samples from `999` to `1999`, which is desirable for computing BC$_a$ confidence intervals.

The bootstrap is computationally intensive, and it takes a while to sample and refit the model 1999 times: The computation ran for about a minute on my Windows 10 computer, producing the output in Figure 33. This figure also shows standard likelihood-ratio based confidence intervals for the coefficients in the model, obtained via *Models > Confidence intervals*; in this case, the results are reasonably similar to those produced by the bootstrap.

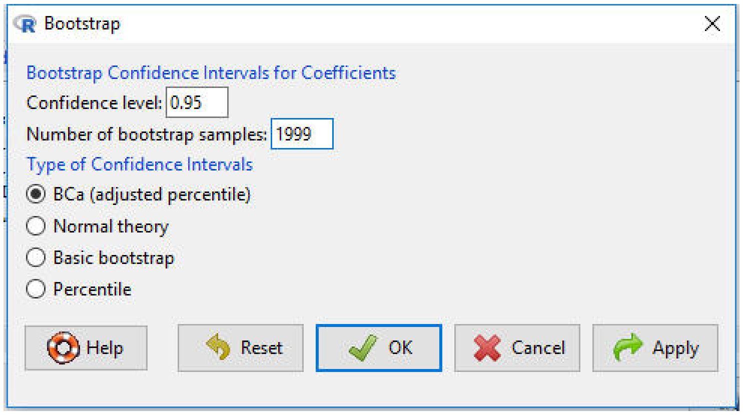
5.2 New Dialog for Delta-Method Standard Errors and Confidence Intervals

A new *Delta Method* dialog uses the `deltaMethod` function in the **car** package (called via the `DeltaMethod` function in the **RcmdrMisc** package) to compute approximate standard errors and confidence intervals for nonlinear functions of regression coefficients. The dialog supports all of the classes of models fit by the R Commander, with the exception of multinomial logit models.

I'll use the `Transact` data set from the **car** package, described by Fox and Weisberg (2019, particularly Section 5.1), for an example. The cases in the data set are 261 branches of a large bank. There are three variables in the data set: `time` is the total minutes of labor for the branch; `t1` is the number of transactions of type 1 performed in the branch; and `t2` is the number of transactions of type 2.

Fox and Weisberg, following Weisberg (2014, Section 7.7.1),[1] perform a linear least-squares regression of `time` on `t1` and `t2`. I've duplicated that regression in a fresh R Commander session after reading the `Transact` data from the **car** package, producing the output in Figure 34.

[1] Weisberg (2014) explains that there are some problems with the least-squares regression and uses this regression to illustrate bootstrapping, an analysis that the reader may wish to replicate.



Bootstrap

Bootstrap Confidence Intervals for Coefficients

Confidence level: 0.95

Number of bootstrap samples: 1999

Type of Confidence Intervals

● BCa (adjusted percentile)
○ Normal theory
○ Basic bootstrap
○ Percentile

Figure 32: The *Bootstrap* dialog for Cowles and Davis's logistic regression model. Because this is a generalized linear model, radio buttons to select case or residual resampling don't appear.

```
> confint(Boot(GLM.7, R=1999), level=0.95, type="bca")
Bootstrap quantiles, type =  bca

                                  2.5 %       97.5 %
(Intercept)                 -3.41722020 -1.270813413
sex[T.male]                 -0.47503743 -0.033311357
neuroticism                  0.03261081  0.190254157
extraversion                 0.08897822  0.246799624
neuroticism:extraversion    -0.01465451 -0.002353057

> Confint(GLM.7, level=0.95, type="LR")
                                Estimate       2.5 %       97.5 % exp(Estimate)
(Intercept)                 -2.358207325 -3.35652914 -1.389154923    0.09458964
sex[T.male]                 -0.247152026 -0.46642058 -0.028694911    0.78102195
neuroticism                  0.110776638  0.03744357  0.185227757    1.11714535
extraversion                 0.166816468  0.09374678  0.241771712    1.18153740
neuroticism:extraversion    -0.008552465 -0.01434742 -0.002833714    0.99148400
                                 2.5 %    97.5 %
(Intercept)                 0.03485603 0.2492859
sex[T.male]                 0.62724342 0.9717129
neuroticism                 1.03815341 1.2034925
extraversion                1.09828160 1.2735034
neuroticism:extraversion    0.98575501 0.9971703
```

Figure 33: Bootstrap and standard likelihood-ratio based confidence intervals for the coefficients in Cowles and Davis's logistic regression.

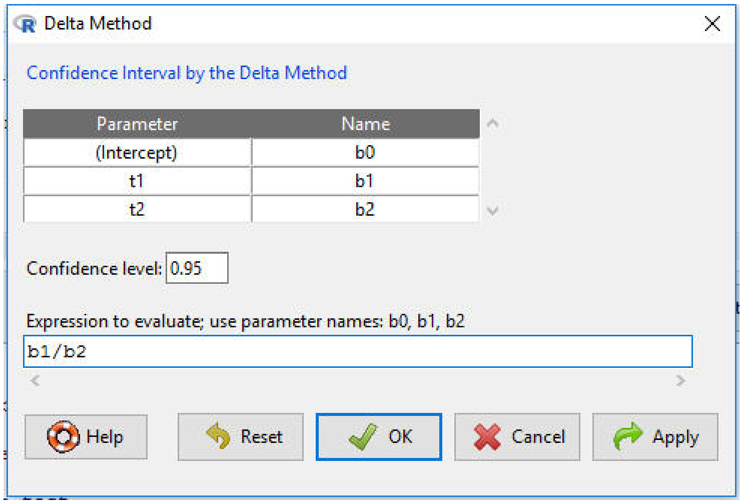
```
> LinearModel.1 <- lm(time ~ t1 + t2, data=Transact)

> summary(LinearModel.1)

Call:
lm(formula = time ~ t1 + t2, data = Transact)

Residuals:
    Min      1Q  Median      3Q     Max
-4652.4  -601.3     2.4   455.7  5607.4

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 144.36944  170.54410   0.847    0.398
t1            5.46206    0.43327  12.607   <2e-16 ***
t2            2.03455    0.09434  21.567   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1143 on 258 degrees of freedom
Multiple R-squared:  0.9091,    Adjusted R-squared:  0.9083
F-statistic:  1289 on 2 and 258 DF,  p-value: < 2.2e-16
```

Figure 34: Least-squares regression of `time` on `t1` and `t2` for the `Transact` data.

It's apparent without a formal test that `t1` transactions are more time-consuming than `t2` transactions,[2] but it's also of interest to estimate the ratio of the two regression coefficients. Selecting *Models > Delta method confidence interval* from the menus brings up the dialog box in Figure 35.

The table at the top of the dialog shows the correspondence between parameters appearing in the regression model and the names by which they're referenced in the nonlinear expression to be evaluated. In this case, `b1` represents the parameter labelled `t1`, and `b2` the parameter labelled `t2`, and so the expression I typed into the text box, `b1/b2`, represents the ratio of the two regression coefficients. Clicking *OK* yields the output in Figure 36. The ratio of estimated coefficients, 2.68, is clearly larger than 1, but the 95% confidence limits indicate that the ratio isn't very precisely estimated, with plausible values ranging between 2.06 and 3.31.

[2] Reader: Use the R Commander *Linear Hypothesis* dialog (see Section 7.7.4 in the text) to test the equality of the coefficients of `t1` and `t2`. You should get a p-value of 1.1×10^{-10}.



Confidence Interval by the Delta Method

Parameter	Name
(Intercept)	b0
t1	b1
t2	b2

Confidence level: 0.95

Expression to evaluate; use parameter names: b0, b1, b2

b1/b2

Figure 35: The *Delta Method* dialog for the regression model fit to the bank transactions data.

```
> DeltaMethod(LinearModel.1, "b1/b2", level=0.95)
  parameter name
(Intercept)   b0
         t1   b1
         t2   b2

      Estimate        SE    2.5 %   97.5 %
b1/b2 2.684653 0.3189858 2.059452 3.309853
```

Figure 36: Delta-method standard error and confidence interval for the ratio of regression coefficients `b1/b2` (i.e., `t1/t2`) in the transactions-data regression.

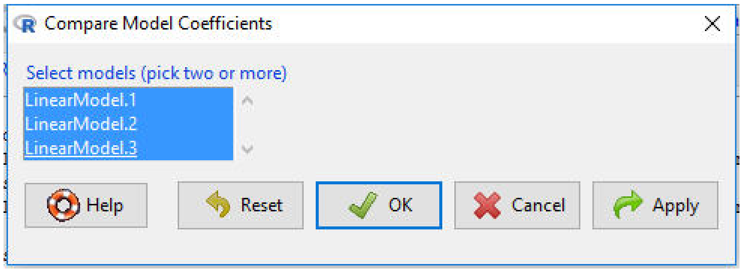


Figure 37: The *Compare Model Coefficients* dialog, selecting all three regression models fit to the Duncan occupational prestige data.

```
> compareCoefs(LinearModel.1, LinearModel.2, LinearModel.3)

Call:
1: lm(formula = prestige ~ education + income, data = Duncan)
2: lm(formula = prestige ~ education + income, data = Duncan, subset = -c(6,
   16))
3: lm(formula = prestige ~ education + income + type, data = Duncan)
               Est. 1     SE 1   Est. 2     SE 2   Est. 3     SE 3
(Intercept)   -6.0647   4.2719  -6.4090   3.6526  -0.1850   3.7138
education      0.5458   0.0983   0.3322   0.0987   0.3453   0.1136
income         0.5987   0.1197   0.8674   0.1220   0.5975   0.0894
type[T.prof]                                      16.6575   6.9930
type[T.wc]                                       -14.6611   6.1088
```

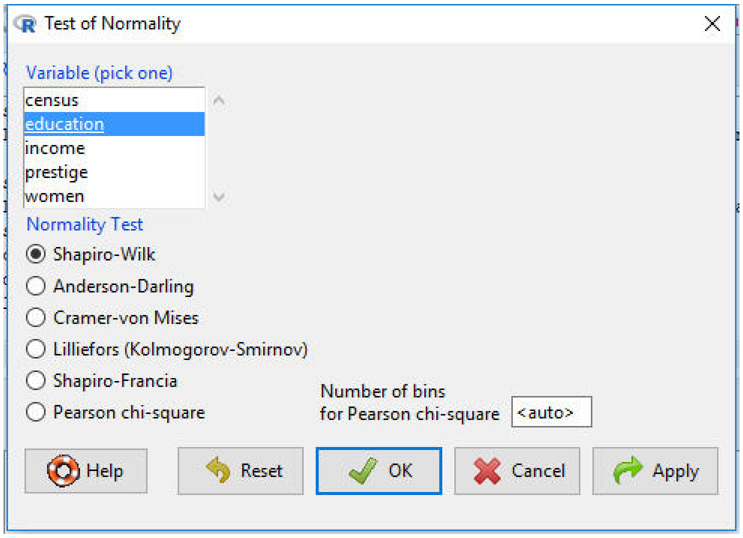
Figure 38: Output produced by the *Compare Model Coefficients* dialog, comparing the three gresson models fit to the Duncan data.

5.3 New Compare-Coefficients Dialog

The new *Compare Model Coefficients* dialog reports a table of regression coefficients and their standard errors for selected statistical models. Any models currently in memory can be compared, whether or not they share regression coefficients, and whether or not they are fit to the same data set or of the same class. The dialog uses the `compareCoefs` function in the **car** package.

To illustrate, I'll use the `Duncan` data set in the **car** package, a data set employed at several points in the text, including in Chapter 7 on linear and generalized linear models. After reading the data in the usual manner from the **car** package, I fit three linear models to the `Duncan` data, regressing `prestige` on `education` and `income`; regressing `prestige` on `education` and `income`, but excluding the unusual cases `minister` (case 6) and `conductor` (16) (see Section 7.8 in the text); and regressing `prestige` on `education`, `income`, and the factor `type` of occupation. The last model is included primarily to show what happens when different terms appear in different models.

After fitting these three models, clicking *Models* > *Compare model coefficients* in the R Commander menus brings up the dialog box in Figure 37. In this example, I select all three models to compare and press *OK*, producing the output in Figure 38. We can see how the `education` coefficient smaller and the `income` coefficients gets larger when the two unusual cases are removed, and how the `education` coefficient decreases when `type` of occupation is entered into the regression.



Test of Normality

Variable (pick one)

- census
- education
- income
- prestige
- women

Normality Test

● Shapiro-Wilk
○ Anderson-Darling
○ Cramer-von Mises
○ Lilliefors (Kolmogorov-Smirnov)
○ Shapiro-Francia
○ Pearson chi-square

Number of bins
for Pearson chi-square <auto>

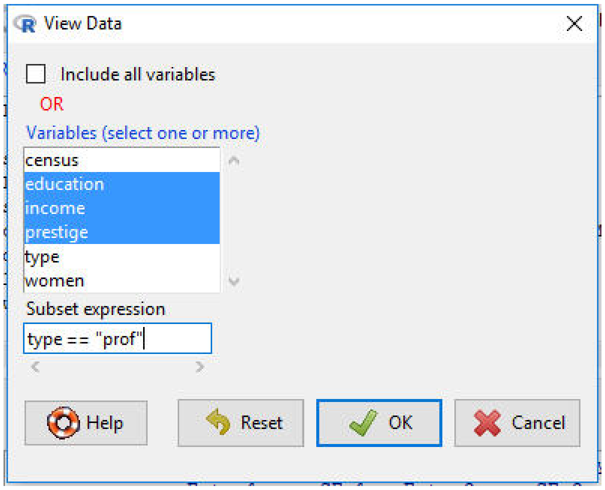
Figure 39: The *Test of Normality* dialog for the `Prestige` data, selecting `education`.

5.4 New Normality Test Dialog

Section 6.5 of the book describes the *Shapiro-Wilk Test for Normality* dialog. This dialog is now replaced with the more general *Test of Normality* dialog, selected via *Statistics > Summaries > Test of normality* from the R Commander menus. As is apparent in Figure 39, the new dialog offers several alternatives to the Shapiro-Wilk test, which remains the default. The additional tests are provided by the **nortest** package (Gross and Ligges, 2015). As in the book, Figure 39 shows the variables in the `Prestige` data set, which I read from the **car** package, and from which I select `education`. The output (which, with the default *Shapiro-Wilk* test selected, is the same as in Figure 6.18 in the text) isn't shown.

5.5 New View Data Dialog

There is a new *View Data* dialog, accessed by *Data > Active data set > View data*, and shown in Figure 40 for the currently active `Prestige` data set. Unlike the *View data set* button in the R Commander toolbar, this dialog allows you to select subsets of variables and cases. I uncheck the default *Include all variables* box, select three of the six variables in the *Variables* list, limit the cases to professional occupations with the *Subset expression* `type == "prof"`, and click *OK*, producing the data-viewer window in Figure 41.



□ Include all variables

OR

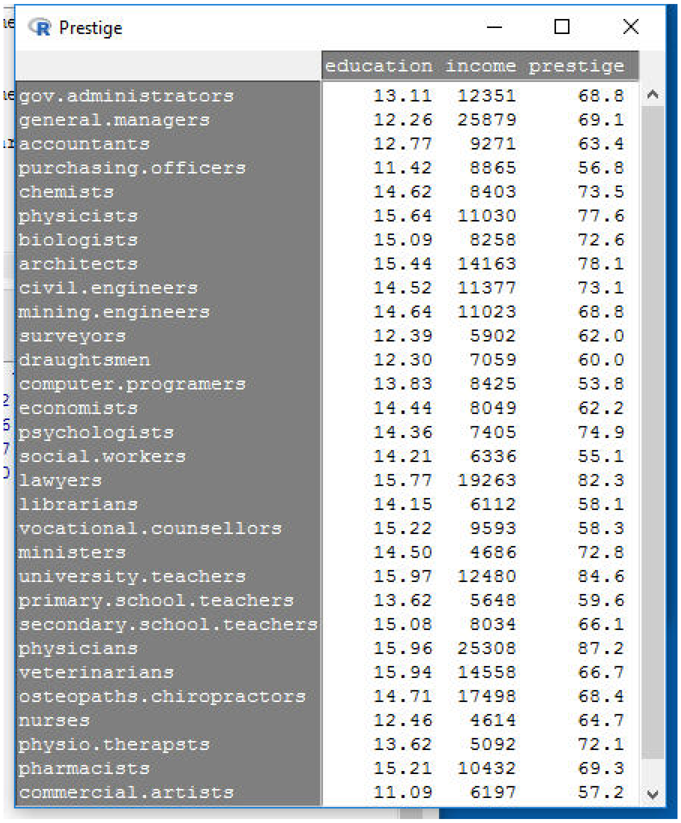
Variables (select one or more)

- census
- education
- income
- prestige
- type
- women

Subset expression

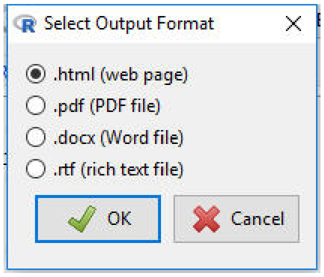
type == "prof"

Figure 40: The *View Data* dialog for the `Prestige` data, selecting the variables `education`, `income`, and `prestige`, and limiting the data to professional occupations.



```
                             education income prestige
gov.administrators               13.11  12351     68.8
general.managers                 12.26  25879     69.1
accountants                      12.77   9271     63.4
purchasing.officers              11.42   8865     56.8
chemists                         14.62   8403     73.5
physicists                       15.64  11030     77.6
biologists                       15.09   8258     72.6
architects                       15.44  14163     78.1
civil.engineers                  14.52  11377     73.1
mining.engineers                 14.64  11023     68.8
surveyors                        12.39   5902     62.0
draughtsmen                      12.30   7059     60.0
computer.programers              13.83   8425     53.8
economists                       14.44   8049     62.2
psychologists                    14.36   7405     74.9
social.workers                   14.21   6336     55.1
lawyers                          15.77  19263     82.3
librarians                       14.15   6112     58.1
vocational.counsellors           15.22   9593     58.3
ministers                        14.50   4686     72.8
university.teachers              15.97  12480     84.6
primary.school.teachers          13.62   5648     59.6
secondary.school.teachers        15.08   8034     66.1
physicians                       15.96  25308     87.2
veterinarians                    15.94  14558     66.7
osteopaths.chiropractors         14.71  17498     68.4
nurses                           12.46   4614     64.7
physio.therapsts                 13.62   5092     72.1
pharmacists                      15.21  10432     69.3
commercial.artists               11.09   6197     57.2
```

Figure 41: The data viewer window for the variables `education`, `income`, and `prestige` in the `Prestige` data, displaying only professional occupations.



Select Output Format

● .html (web page)
○ .pdf (PDF file)
○ .docx (Word file)
○ .rtf (rich text file)

Figure 42: The revised *Select Output Format* dialog.

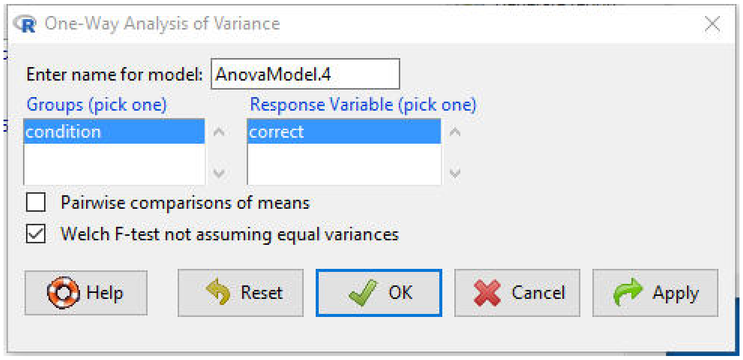
5.6 Rich Text File R Markdown Output

If you've installed the optional Pandoc software (see Section 2.5 in the book), then you'll be able to create a rich text file (`.rtf`) report from the R Markdown document generated by the R Commander, as an alternative to the previously available HTML file, PDF file, and Word file output formats (see Section 3.6 in the book). The revised *Select Output Format* dialog (cf., Figure 3.19 in the text) appears in Figure 42. Most word processors are able to edit rich text files. The default output format is still an HTML file.

5.7 One-Way ANOVA Welch F-Test Option

The *One-Way Analysis of Variance* dialog (see Section 6.1.2 of the text) has acquired a check box for the *Welch F-test not assuming equal variances*. The corresponding test, introduced by Welch (1951), is the several-samples analog to the two-sample Welch-Satterthwaite t-test described in Section 6.1.1 of the text.

For an example, I'll use the `Friendly` memory-experiment data from the **car** package (as in Section 6.1.2). Reading the data and clicking *Statistics > Means > One-way ANOVA* in the R Commander menus produces the dialog in Figure 43. Unlike in the text, I use number `correct` as the response variable rather than employing a logit transformation of the proportion correct to stabilize the within-group variances. Pressing the *Apply* button produces the output at the top of Figure 44; for comparison, I then uncheck the *Welch F-test* box and press *OK* to produce the standard ANOVA output at the bottom of Figure 44. In this instance, the Welch F-test yields a larger p-value than the standard F-test.



Enter name for model: AnovaModel.4

Groups (pick one): condition

Response Variable (pick one): correct

☐ Pairwise comparisons of means

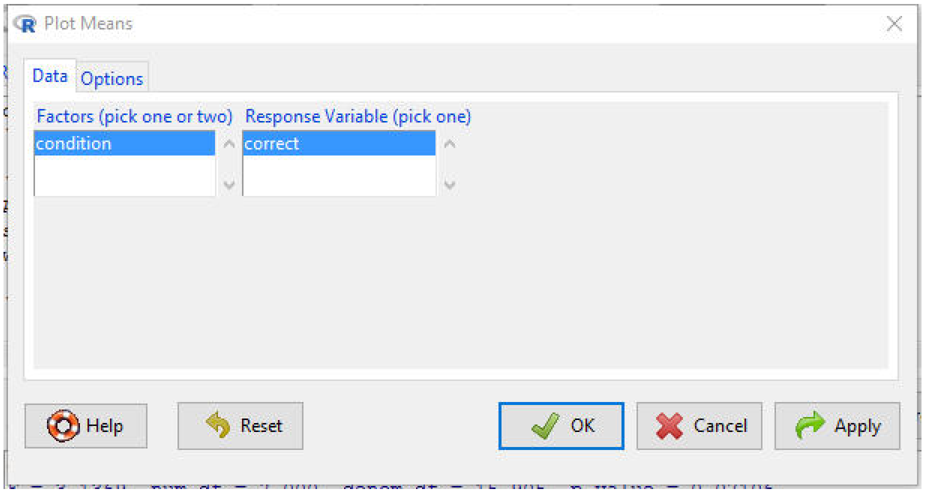
☑ Welch F-test not assuming equal variances

Figure 43: The revised *One-Way Analysis of Variance* dialog for the **Friendly** memory data, with the *Welch F-test* box checked.

```
> with(Friendly, numSummary(correct, groups=condition, statistics=c("mean",
+   "sd")))
       mean       sd data:n
Before 36.6 5.337498     10
Meshed 36.6 3.025815     10
SFR    30.3 7.334091     10

> oneway.test(correct ~ condition, data=Friendly) # Welch test

        One-way analysis of means (not assuming equal variances)

data:  correct and condition
F = 3.1369, num df = 2.000, denom df = 15.905, p-value = 0.07106
```

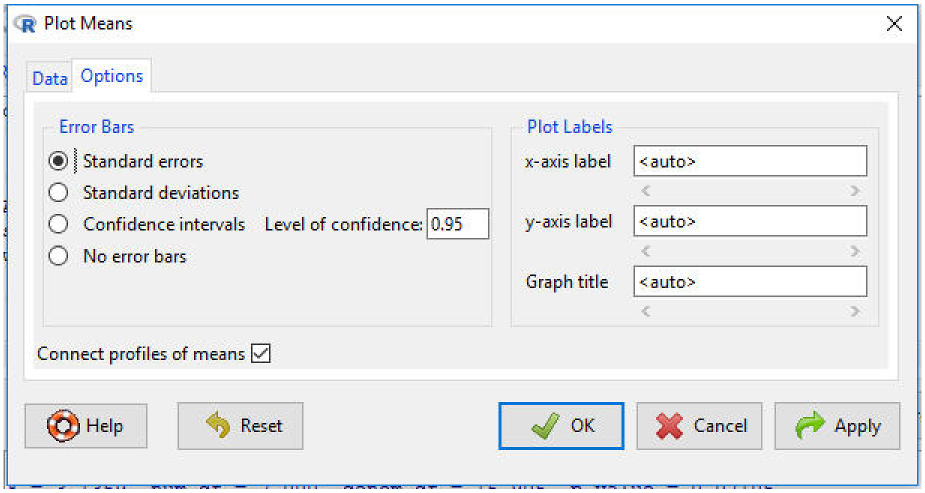
```
> AnovaModel.5 <- aov(correct ~ condition, data=Friendly)

> summary(AnovaModel.5)
            Df Sum Sq Mean Sq F value Pr(>F)
condition    2  264.6  132.30   4.341 0.0232 *
Residuals   27  822.9   30.48
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> with(Friendly, numSummary(correct, groups=condition, statistics=c("mean",
+   "sd")))
       mean       sd data:n
Before 36.6 5.337498     10
Meshed 36.6 3.025815     10
SFR    30.3 7.334091     10
```

Figure 44: Output produced by the *One-Way Analysis of Variance* dialog for the **Friendly** memory data. The Welch F-test is at the top; a standard F-test assuming equal group variances is at the bottom.



Factors (pick one or two) condition

Response Variable (pick one) correct



Error Bars

● Standard errors

○ Standard deviations

○ Confidence intervals Level of confidence: 0.95

○ No error bars

Plot Labels

x-axis label <auto>

y-axis label <auto>

Graph title <auto>

Connect profiles of means ☑

Figure 45: The revised *Plot Means* dialog for the Friendly memory data; *Data* tab (top), and *Options* tab (bottom) with the *Connect profiles of means* box checked by default.

5.8 Plotting Means With or Without Connecting Lines

The *Plot Means* (*Graphs > Plot of means*) dialog gains a *Connect profiles of means* box, which is checked by default. Continuing with the Friendly memory data, the dialog box is shown in Figure 45. I press the *Apply* button, producing the graph at the top of Figure 46; unchecking the *Connect profiles of means* box and clicking *OK* produces the graph at the bottom of the figure.

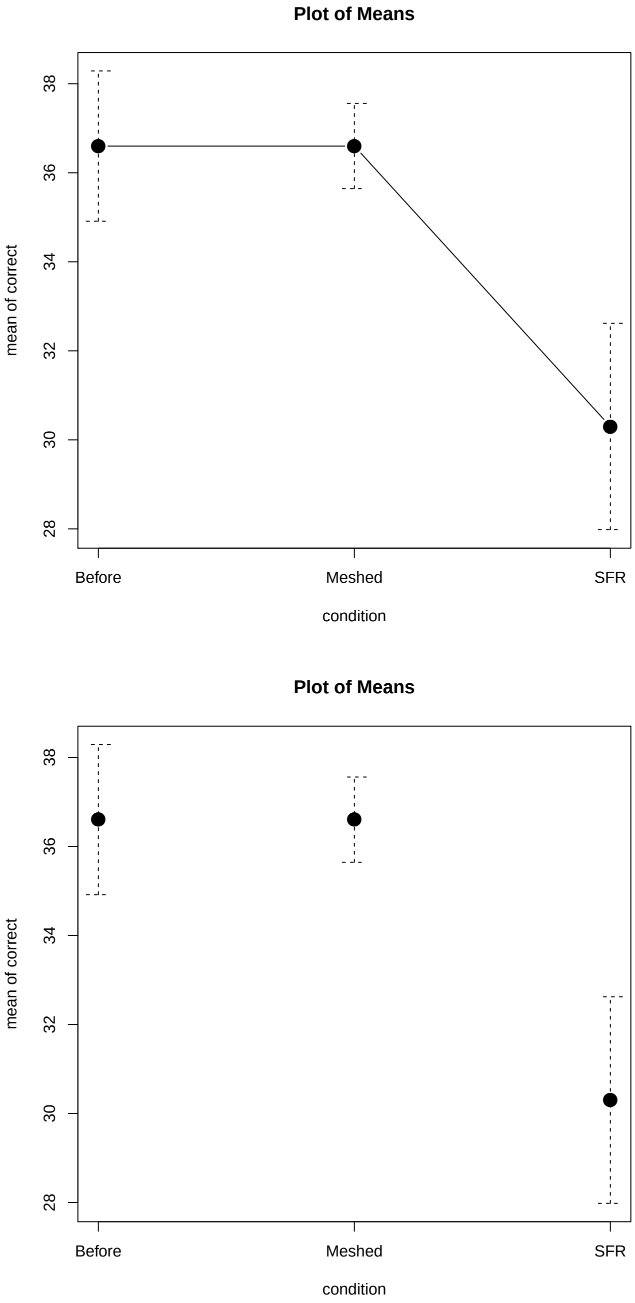
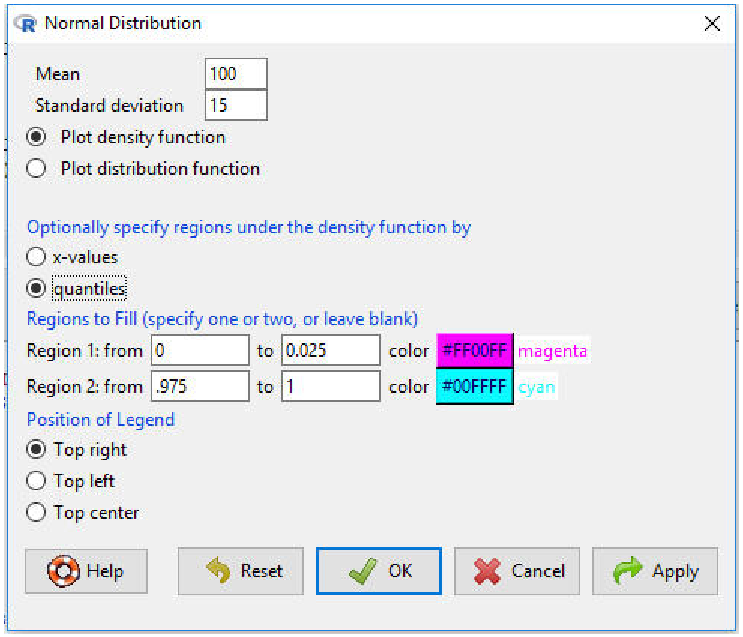


Figure 46: Graphs of profiles of mean `correct` by `condition` for the `Friendly` memory data: connected profiles (top) and unconnected (bottom).



Normal Distribution

Mean 100

Standard deviation 15

● Plot density function

○ Plot distribution function

Optionally specify regions under the density function by

○ x-values

● quantiles

Regions to Fill (specify one or two, or leave blank)

Region 1: from 0 to 0.025 color #FF00FF magenta

Region 2: from .975 to 1 color #00FFFF cyan

Position of Legend

● Top right

○ Top left

○ Top center

Figure 47: The revised *Normal Distribution* plotting dialog.

5.9 Plotting Regions Under Continuous Probability Distributions

The dialog boxes for plotting continuous probability distributions, described in the book in Section 8.2, now provide for showing up to two regions under a density curve. To demonstrate, I select *Distributions > Continuous distributions > Normal distribution > Plot normal distribution*, obtaining the dialog in Figure 47. I complete the dialog by changing the *Mean* from the default `0` to `100` and the standard deviation from `1` to `15`; clicking the *quantiles* radio button (the default is *x-values*); filling in the regions text boxes (which are blank by default); and using the color magenta for the first region and cyan for the second, chosen with the color-selection buttons (the default in each case is gray).[3] Clicking *OK* produces the graph in Figure 48.

[3] See Section 3.9.3 of the book for a discussion of color selection in the R Commander.

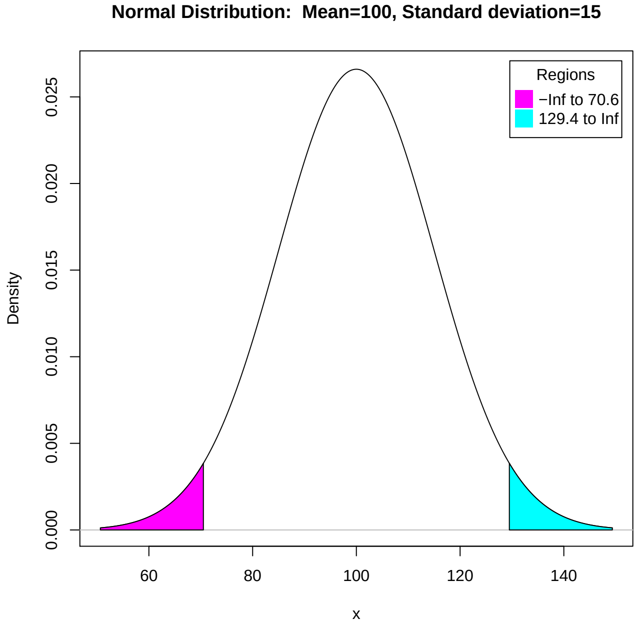


Figure 48: Graph of the normal density function for $\mu = 100$ and $\sigma = 15$, with the regions below the 0.025 quantile and above the 0.975 quantile colored respectively magenta and cyan.

References

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Canty, A. and Ripley, B. (2016). ***boot**: Bootstrap R (S-Plus) Functions.* R package version 1.3-18.

Cowles, M. and Davis, C. (1987). The subject matter of psychology: Volunteers. *British Journal of Social Psychology*, 26:97—–102.

Davis, C., Blackmore, E., Katzman, D. K., and Fox, J. (2005). Female adolescents with anorexia nervosa and their parents: A case-control study of exercise attitudes and behaviours. *Psychological Medicine*, 35:377–386.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Applications.* Cambridge University Press, Cambridge.

Fox, J. (2016). *Applied Regression Analysis and Generalized Linear Models.* Sage, Thousand Oaks, CA, third edition.

Fox, J. (2017). *Using the R Commander: A Point-and-Click Interface for R*. Chapman & Hall/CRC Press, Boca Raton FL.

Fox, J. and Weisberg, S. (2019). *An R Companion to Applied Regression.* Sage, Thousand Oaks CA, third edition.

Gross, J. and Ligges, U. (2015). ***nortest**: Tests for Normality.* R package version 1.0-4.

O'Brien, R. G. and Kaiser, M. K. (1985). MANOVA method for analyzing repeated measures designs: An extensive primer. *Psychological Bulletin*, 97:316–333.

Weisberg, S. (2014). *Applied Linear Regression.* Wiley, Hoboken, NJ, fourth edition.

Welch, B. L. (1951). On the comparison of several mean values: an alternative approach. *Biometrika*, 38:330–336.

Wickham, H. and Miller, E. (2016). ***haven**: Import and Export 'SPSS', 'Stata' and 'SAS' Files.* R package version 1.0.0.