New R Commander Features

John Fox
2019-09-04

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 2.6-0 (September 2019)

1.1 The Official R Commander Hex Sticker

The Rcmdr package now has a hex sticker, shown in Figure 1! It can be displayed from the R Commander menus via Help > R Commander hex sticker.

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

There are two new dialog boxes for fitting linear and generalized linear mixed-effects models (LMMs and GLMMs), respectively by the lmer and glmer() functions in the lme4 package (Bates et al., 2015). The dialogs are similar, and so I'll illustrate only the LMM dialog; the GLMM dialog is slightly more complex because it includes family and link-function components (as does the GLM dialog described in Section 7.4 of the text). Once fit, a LMM or GLMM becomes the active statistical model in the R Commander, and most of the items in the Models menu, including for various regression diagnostics, are applicable.

I'll illustrate by borrowing an example from Fox and Weisberg (2019, Section 7.2.5), where it is developed in more detail with a mixed-effects model slightly different from the one that I'll fit here. The data, in the Blackmore data set from the carData package, pertain to a study of exercise and eating disorders in adolescent girls reported by Davis et al. (2005). Because the carData package is loaded by default when the Rcmdr starts up, we can access the data in the usual manner via Data > Data in packages > Read data from an attached package, and then obtain a codebook for the data set by Data > Active data set > Help on active data set, displayed in Figure 2.

Figure 3 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 \( \log_2(\text{exercise} + 5/60) \). 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 \texttt{group}, \texttt{age}, and their interaction, and random \texttt{age} slopes and (implied) intercepts that vary by subjects. Subtracting 8 from \texttt{age} locates the regression intercept at the start of the study; the identity (or “inhibit”) function \texttt{I()} is required so that the minus operator (-) is interpreted as subtraction. The resulting output is shown in Figure 4.

Selecting \textit{Models} > \textit{Hypothesis tests} > \textit{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 5. There is, therefore, very strong evidence for an interaction of \texttt{group} and \texttt{age}.

The coefficients of the model are interpretable: The intercept is the average value of log base 2 exercise at \texttt{age} 8 for the baseline "control" group; the coefficient \texttt{group[T.patient]} is the difference in average log base 2 exercise at \texttt{age} 8 between the "patient" and "control" groups; the coefficient \texttt{I(age - 8)} is the average \texttt{age} slope in the baseline "control" group; and the \texttt{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 \texttt{age} for the "patient" group.

We can visualize the fixed effects by selecting \textit{Models} > \textit{Graphs} > \textit{Predictor effect plots}, producing the dialog box in Figure 6. I completed the dialog as shown and clicked \texttt{OK}, producing the graph in Figure 7. 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 \textit{Models} menu, including those for numerical and graphical diagnostics.
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 2: Codebook for the Blackmore data set.
Figure 3: 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)
gr[T.ptnt] -0.775
I(age - 8) -0.489  0.379
    g[T.]:I(-8) 0.389 -0.489 -0.796

Figure 4: 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 5: Type-II tests for the fixed effects in the linear mixed model fit to the Blackmore data.
Figure 6: The *Predictor Effect Plots* dialog for the mixed model fit to the Blackmore data.

![Predictor Effect Plots dialog](image)

Figure 7: The predictor effect plot for *age* for the mixed model fit to the Blackmore data, showing the *age* by *group* interaction.

![age predictor effect plot](image)
2 Rcmdr Version 2.5-0 (August 2018)

2.1 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 8, 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 9, 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 10. 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 11.

The Models > Graphs menu also acquires a new Predictor effect plots menu item, leading to the dialog box illustrated in Figure 12. 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 13 suggests an incorrectly modeled nonlinear partial relationship between infantMortality and ppgdp.

2.2 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 14. I selected the variable infantMortality from the list box and left the other selections at their defaults. The Box-Cox
> 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 9: Output from the regression of infant mortality on per-capita GDP and group, followed by the output produced by the Response Transformation dialog.
Figure 10: The Influence Index Plot dialog.

Figure 11: Graph produced by the Influence Index Plot dialog.
Figure 12: The *Predictor Effect Plots* dialog.

Figure 13: Predictor effect plots for the regression model fit to the UN infant-mortality data, including partial residuals.
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 15, suggests that the log transformation makes the distribution of `infantMortality` symmetric.

Statistics > Summaries > Transform toward normality brings up the dialog box in Figure 16, 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 17, which suggests the log transformation of both `infantMortality` and `ppgdp`.

### 2.3 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 15: Symmetry boxplots for infantMortality in the UN data.
Figure 16: 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

<table>
<thead>
<tr>
<th></th>
<th>Est</th>
<th>Power Rounded</th>
<th>Pwr</th>
<th>Wald Lwr Bnd</th>
<th>Wald Upr Bnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>infantMortality</td>
<td>0.0375</td>
<td>0</td>
<td>-0.0690</td>
<td>0.1441</td>
<td></td>
</tr>
<tr>
<td>ppgdp</td>
<td>0.0517</td>
<td>0</td>
<td>-0.0197</td>
<td>0.1230</td>
<td></td>
</tr>
</tbody>
</table>

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 17: Output produced by the Transform Variables Toward Normality dialog, for the unconditional Box-Cox power transformation of infantMortality and ppgdp.
3 Rcmdr Version 2.4-0 (August 2017)

3.1 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 18), 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.

3.2 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 19, with the new default setting of the scientific.notation option. Note that the very small number 2.2e-16 = 2.2 × 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.

3.3 Dialog for Plotting A Discrete Numeric Variable

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

3.4 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.

3.5 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.
Figure 18: 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  < 2e-16 ***
income       0.59873  0.11967   5.003  1.8e-07 ***
---
Signif. codes:  *** 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 19: Output from Duncan’s occupational prestige regression, illustrating the suppression of scientific notation (cf., Figure 7.2 in the text).
3.6 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.

4 Rcmdr Version 2.3-0 (August 2016)

4.1 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 20. 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 BCa 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 21. 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.

4.2 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), 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 22.

---

1Weisberg (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.
Figure 20: 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")
> Confint(GLM.7, level=0.95, type="LR")

![Bootstrap dialog](image)

Figure 21: 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 22: 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 23.

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, \(b_1\) represents the parameter labelled t1, and \(b_2\) the parameter labelled t2, and so the expression I typed into the text box, \(b_1/b_2\), represents the ratio of the two regression coefficients. Clicking OK yields the output in Figure 24. 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 \(1.1 \times 10^{-10}\).
Figure 23: The *Delta Method* dialog for the regression model fit to the bank transactions data.

Figure 24: Delta-method standard error and confidence interval for the ratio of regression coefficients $b_1/b_2$ (i.e., $t_1/t_2$) in the transactions-data regression.
4.3 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 25. In this example, I select all three models to compare and press **OK**, producing the output in Figure 26. 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.
4.4 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 27, 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 27 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.

4.5 New View Data Dialog

There is a new *View Data* dialog, accessed by *Data > Active data set > View data*, and shown in Figure 28 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 29.
Figure 28: The View Data dialog for the Prestige data, selecting the variables education, income, and prestige, and limiting the data to professional occupations.

Figure 29: The data viewer window for the variables education, income, and prestige in the Prestige data, displaying only professional occupations.
4.6 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 30. Most word processors are able to edit rich text files. The default output format is still an HTML file.

4.7 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 31. 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 32; for comparison, I then uncheck the Welch F-test box and press OK to produce the standard ANOVA output at the bottom of Figure 32. In this instance, the Welch F-test yields a larger p-value than the standard F-test.
Figure 31: The revised *One-Way Analysis of Variance* dialog for the *Friendly* memory data, with the *Welch F-test* box checked.

```r
> 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
```

```r
> 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
```

```r
> 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 32: 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.
Figure 33: 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.

4.8 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 33. I press the *Apply* button, producing the graph at the top of Figure 34; unchecking the *Connect profiles of means* box and clicking *OK* produces the graph at the bottom of the figure.
Figure 34: Graphs of profiles of mean correct by condition for the Friendly memory data: connected profiles (top) and unconnected (bottom).
4.9 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 35. 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).\footnote{See Section 3.9.3 of the book for a discussion of color selection in the R Commander.} Clicking OK produces the graph in Figure 36.
Figure 36: 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


