New R Commander Features

John Fox

2017-08-29

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 2.4-0 (August 2017)

1.1 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 1), 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*.

1.2 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 2, with the new default setting of the `scientific.notation` option. Note that the very small number

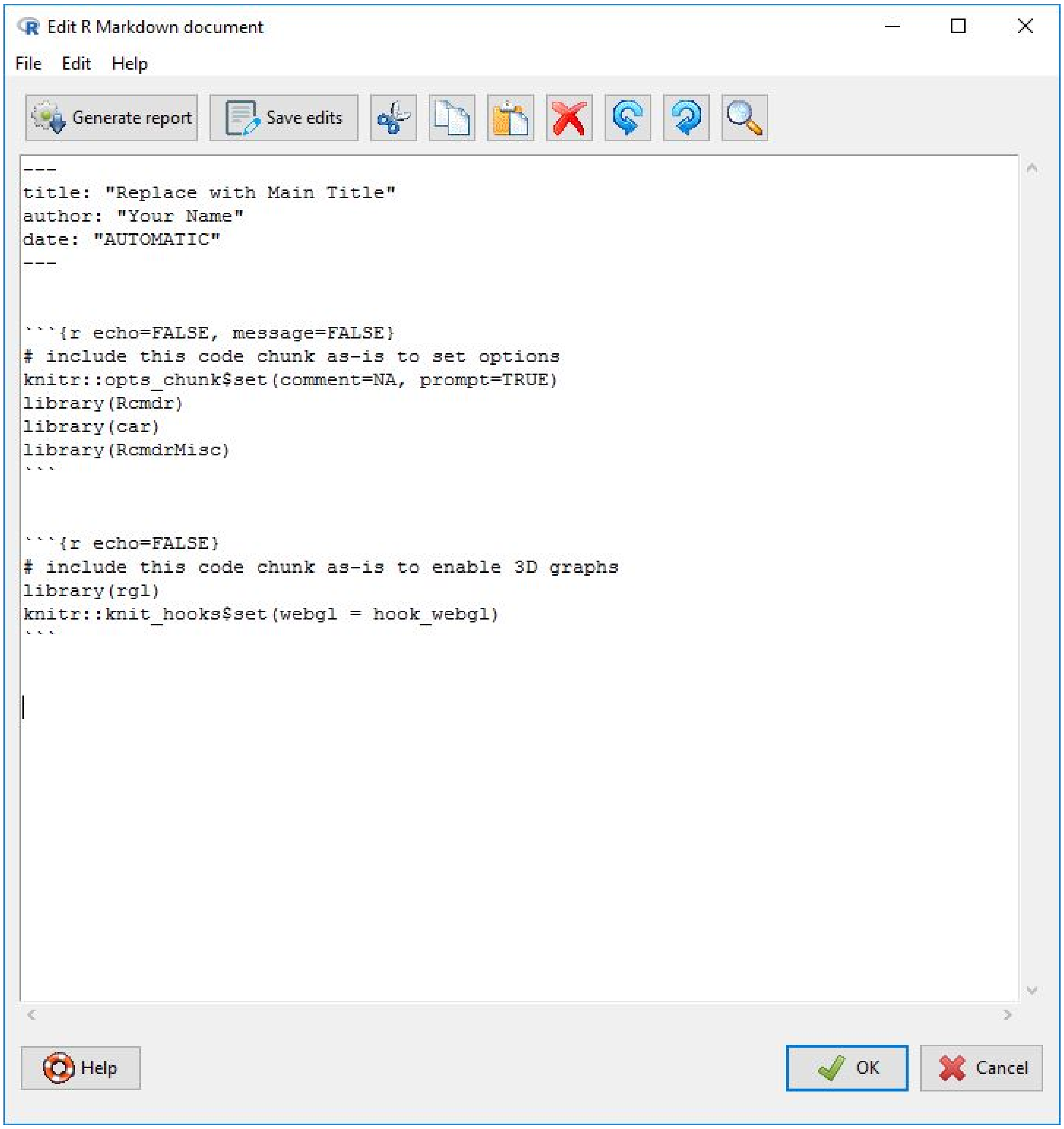


Figure 1: The R Commander R Markdown document editor, opened at the start of a session.

```
> 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 2: Output from Duncan's occupational prestige regression, illustrating the suppression of scientific notation (cf., Figure 7.2 in the text).

`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.

1.3 Dialog for Plotting A Discrete Numeric Variable

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

1.4 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.

1.5 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.

1.6 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.

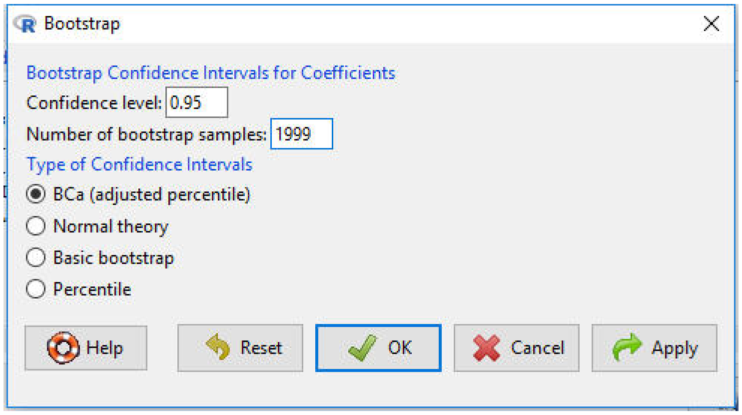


Figure 3: The *Bootstrap* dialog for Cowles and Davis's logistic regression model. Because this is a generalized linear model, radio buttons to select case or residual resampling don't appear.

2 Rcmdr Version 2.3-0 (August 2016)

2.1 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 3. 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 4. 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.

2.2 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 (2011, particularly Section 4.4.6), for an example. The cases in the data set are 261 branches of a large

```
> 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 4: 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 5: Least-squares regression of `time` on `t1` and `t2` for the `Transact` data.

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 5.

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 6.

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 7. 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.

[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.

[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}.

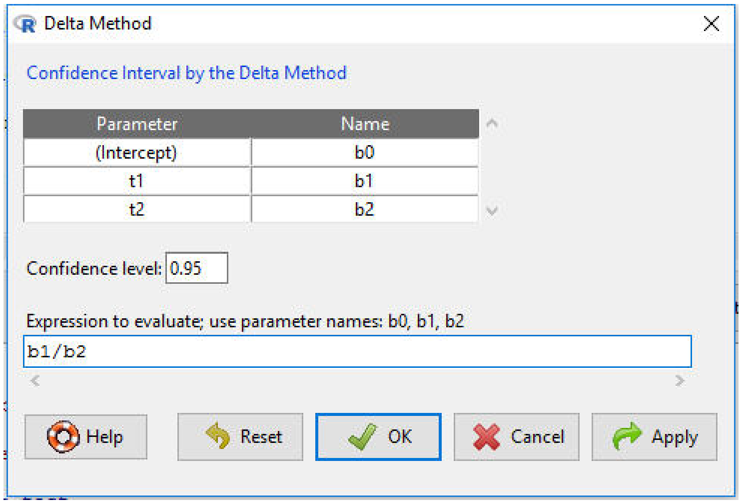


Figure 6: The *Delta Method* dialog for the regression model fit to the bank transactions data.

```
> 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 7: Delta-method standard error and confidence interval for the ratio of regression coefficients `b1/b2` (i.e., `t1/t2`) in the transactions-data regression.

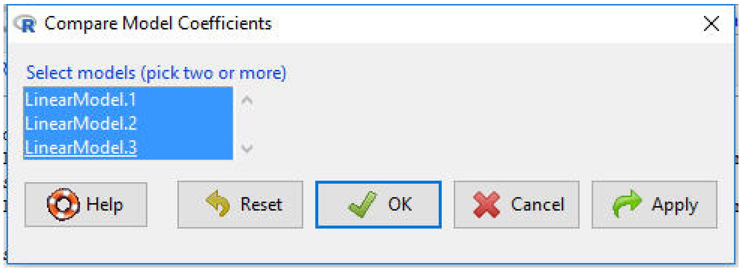


Figure 8: The *Compare Model Coefficients* dialog, selecting all three regression models fit to the `Duncan` occupational prestige data.

```
> 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 9: Output produced by the *Compare Model Coefficients* dialog, comparing the three regression models fit to the `Duncan` data.

2.3 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 8. In this example, I select all three models to compare and press *OK*, producing the output in Figure 9. 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.

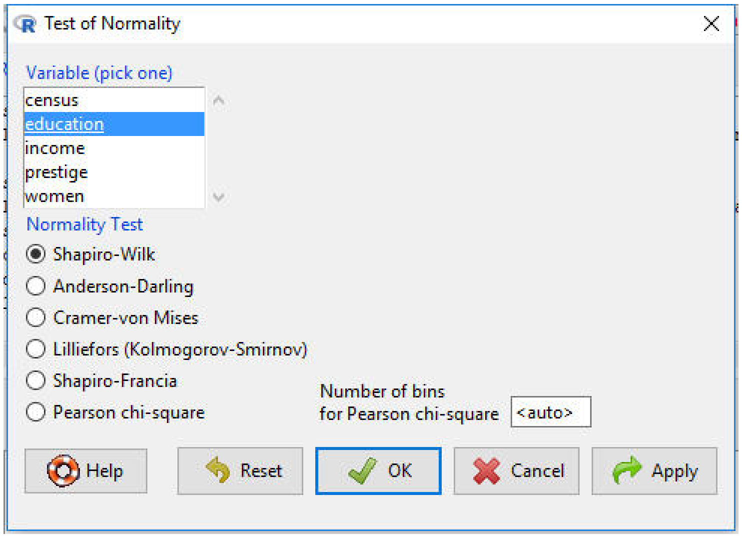


Figure 10: The *Test of Normality* dialog for the `Prestige` data, selecting `education`.

2.4 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 10, 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 10 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.

2.5 New View Data Dialog

There is a new *View Data* dialog, accessed by *Data > Active data set > View data*, and shown in Figure 11 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 12.

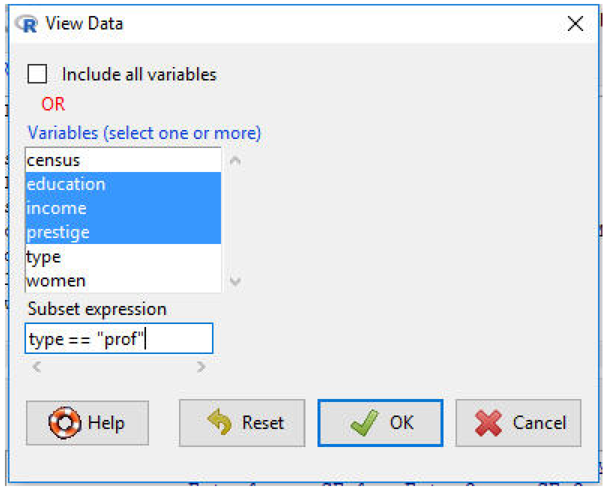
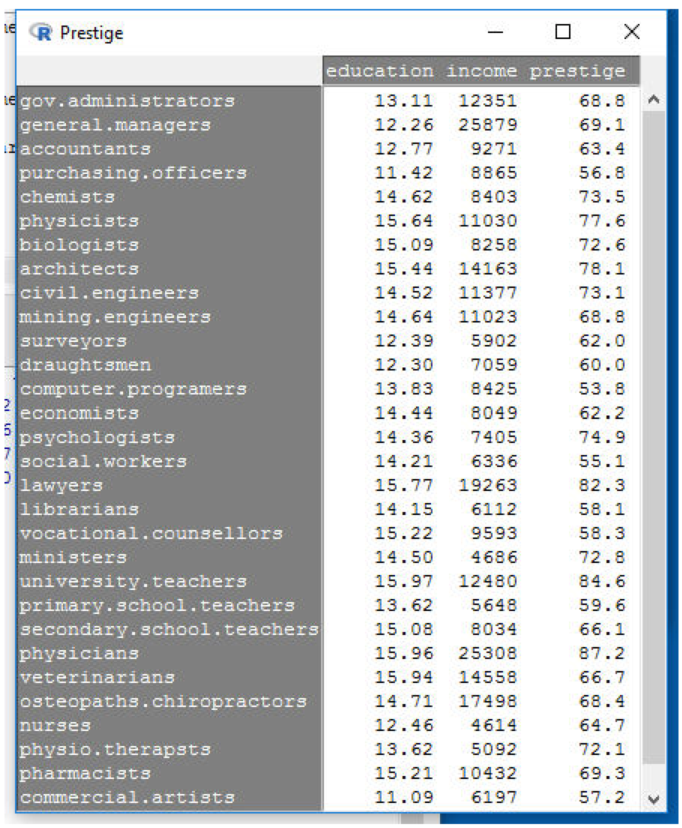


Figure 11: The *View Data* dialog for the `Prestige` data, selecting the variables `education`, `income`, and `prestige`, and limiting the data to professional occupations.



Prestige

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

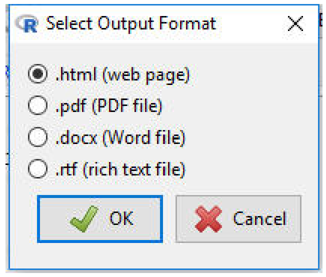


Figure 13: The revised *Select Output Format* dialog.

2.6 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 13. Most word processors are able to edit rich text files. The default output format is still an HTML file.

2.7 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 14. 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 15; for comparison, I then uncheck the *Welch F-test* box and press *OK* to produce the standard ANOVA output at the bottom of Figure 15. In this instance, the Welch F-test yields a larger p-value than the standard F-test.

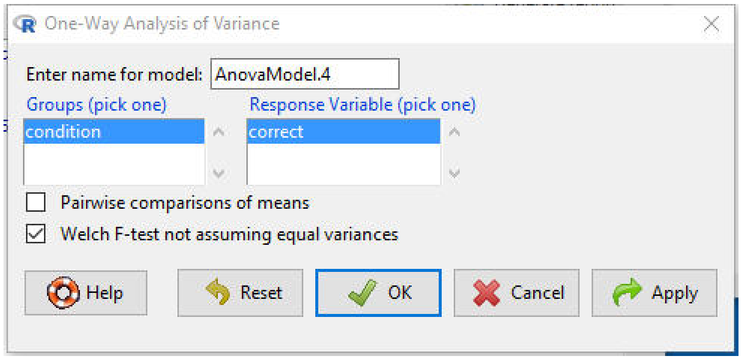


Figure 14: The revised *One-Way Analysis of Variance* dialog for the `Friendly` memory data, with the *Welch F-test* box checked.

```
> 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 15: 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.

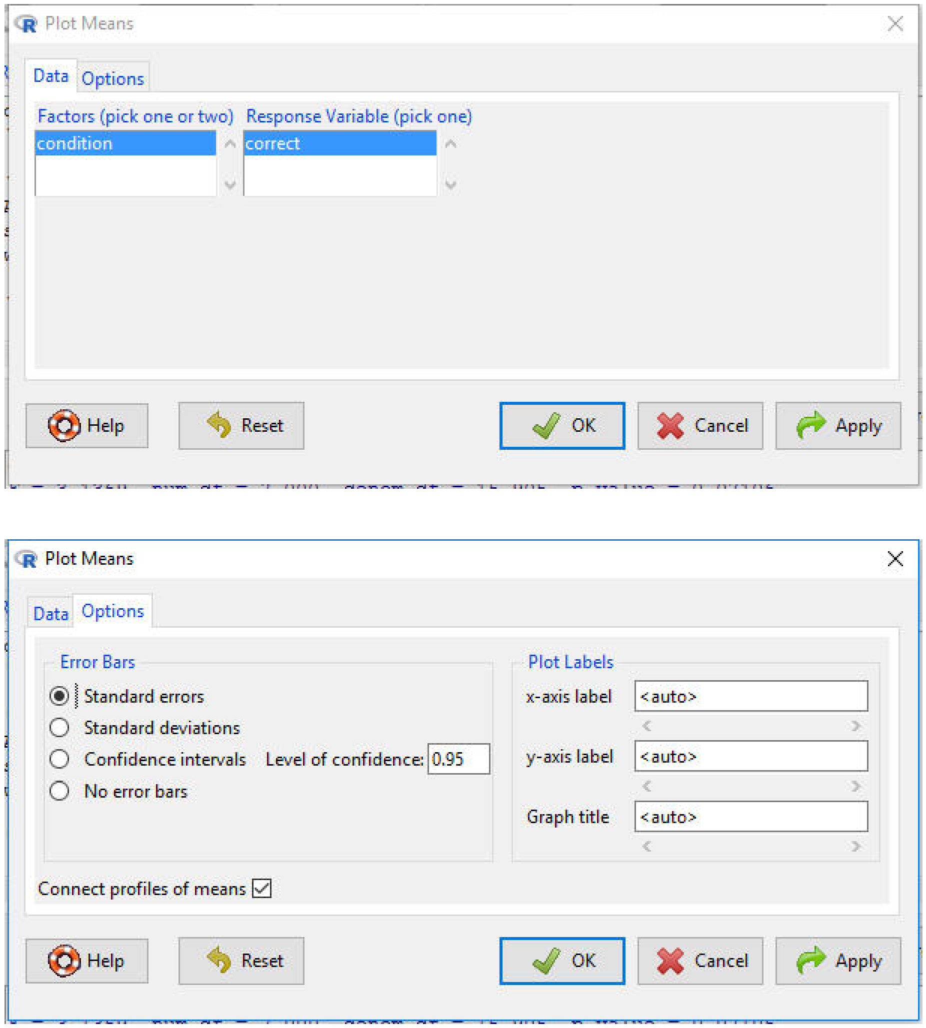


Figure 16: The revised *Plot Means* dialog for the `Friendly` memory data; *Data* tab (top), and *Options* tab (bottom) with the *Connect profiles of means* box checked by default.

2.8 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 16. I press the *Apply* button, producing the graph at the top of Figure 17; unchecking the *Connect profiles of means* box and clicking *OK* produces the graph at the bottom of the figure.

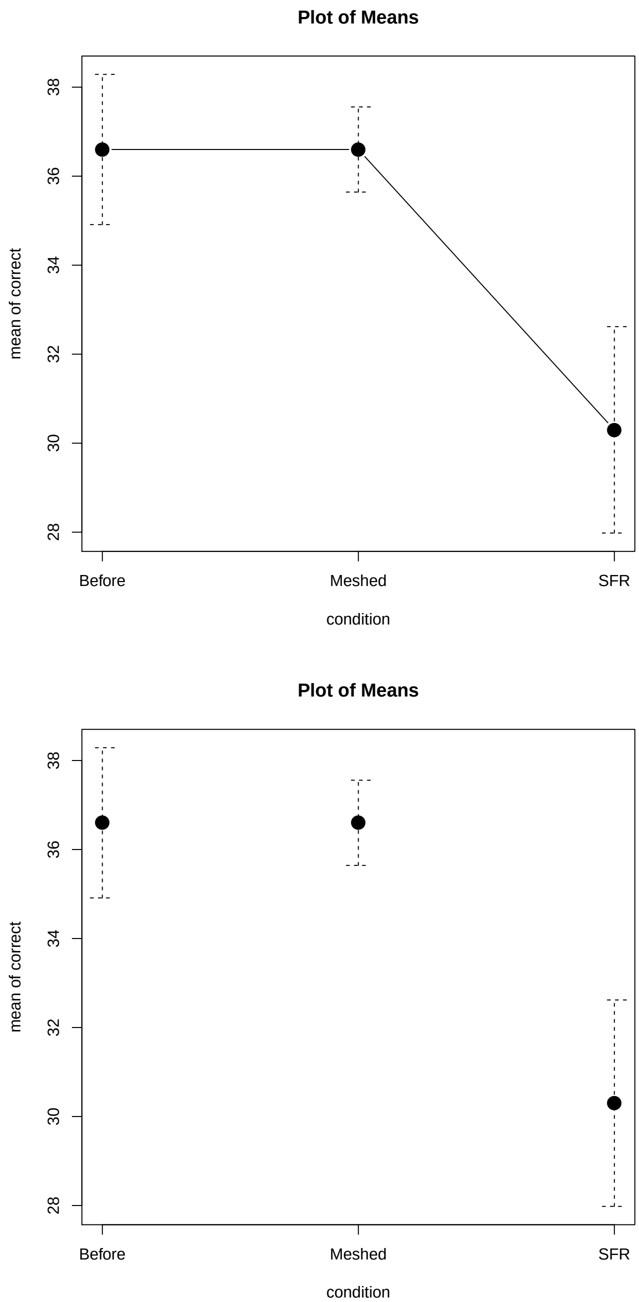


Figure 17: Graphs of profiles of mean `correct` by `condition` for the `Friendly` memory data: connected profiles (top) and unconnected (bottom).

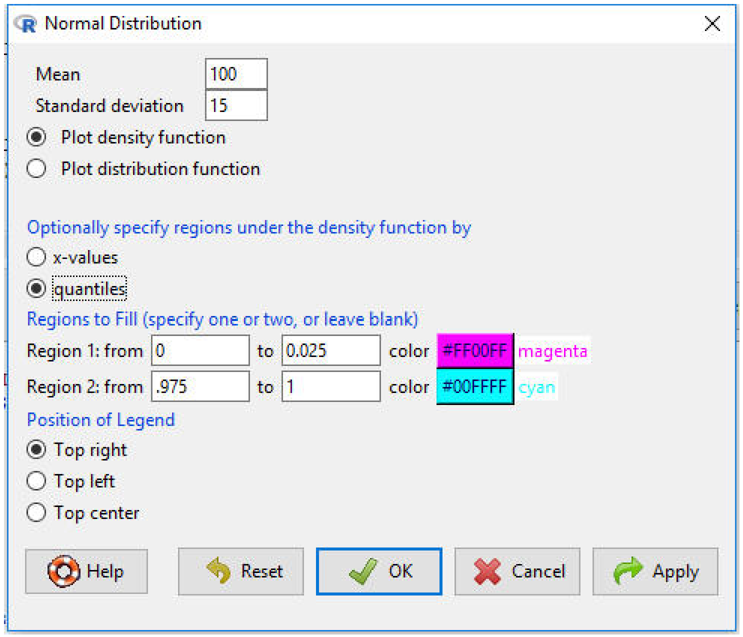


Figure 18: The revised *Normal Distribution* plotting dialog.

2.9 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 18. 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 19.

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

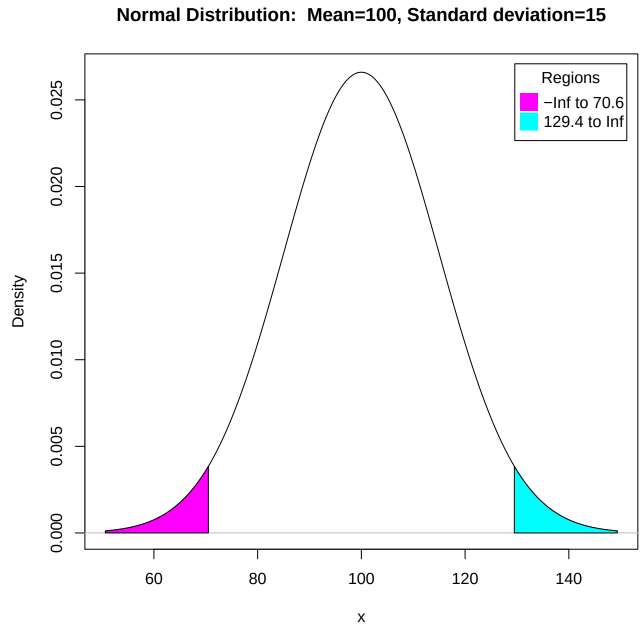


Figure 19: Graph of the normal density function for $\mu = 100$ and $\sigma = 15$, with the regions below the 0.025 quantile and above the 0.975 quantile colored respectively magenta and cyan.

References

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.

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. (2011). *An R Companion to Applied Regression.* Sage, Thousand Oaks, CA, second edition.

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

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.