Exercises for

Modern Applied Statistics with S-Plus

Third edition

by

W. N. Venables and B. D. Ripley

Springer (1999). ISBN 0-387-98825-4

Selectable links are in this colour.

Exercises and Further Exercises

For convenience we reproduce the exercises from the text as well as further exercises.

Chapter 2

2.1 How would you find the index(es) of specified values within a vector? For example, where is the hill race (in `hills`) with a climb of 2100 feet? Answer

2.2 The column `ftv` in data frame `birthwt` counts the number of visits. Reduce this to a factor with levels 0, 1 and '2 or more'. [Hint: manipulate the `levels`, or investigate functions `cut` and `merge.levels`.] Answer

2.3 Write a simple function to compute the median absolute deviation (used in robust statistics; see Section 5.5) median$|x - \mu|$ with default μ the sample median. Compare your answer with the S-PLUS function `mad`. Answer

2.4 Suppose `x` is an object with named components and `out` is a character string vector. How would you make a new object obtained from `x` by *excluding* any components whose names are in `out`? Answer

2.5 Given a matrix `X` of distinct rows and a vector `w` of the number of times that each row should occur, reconstruct the original matrix. Answer

2.6 "I calculated a cross-correlation matrix. I want to print only members of this matrix that are larger than 0.90 and I want to include dimnames in the answer."

2.7 "I have a large data frame (5 000 observations) and I would like the cases where a variable indicating ethnic group is in (1,3,4,6,7)." Answer

Chapter 3

3.1 The data frame `survey` contains the results of a survey of 237 first-year statistics students at Adelaide University. For a graphical summary of all the variables, use `plot(survey)`. Note that this produces a dotchart for factor variables, and a normal scores plot for the numeric variables.

One component of this data frame, `Exer`, is a factor object containing the responses to a question asking how often the students exercised. Produce a barchart of these responses. Use `table` and `pie` or `piechart` to create a pie chart of the responses. Do you like this better than the bar plot? Which is more informative? Which gives a better picture of exercise habits of students? The `pie` function takes an argument `names` which can be used to put labels on each pie slice. Redraw the pie chart with labels. Alternatively, you could add a legend to identify the slices.

You might like to try the same things with the `Smoke` variable, which records responses to the question ,“How often do you smoke?” Note that `table` and `levels` ignore missing values; if you wish to include non-respondents in your chart use `summary` to generate the values, and `names` on the summary object to generate the labels. Answer

3.2 Make a plot of petal width versus petal length of the `iris` data for a partially sighted audience, identifying the three species. You will need to double the annotation size, thicken the lines and change the layout to allow larger margins for the larger annotation. Answer

3.3 Plot $\sin(x)$ against x, using 200 values of x between $-\pi$ and π, but do not plot any axes yet (use parameter `axes=F` in the call to `plot`). Add a y-axis passing through the origin using the ‘extended’ style and horizontal labels. Add an x-axis with tick-marks from $-\pi$ to π in increments of $\pi/4$, twice the usual length. Answer

3.4 Cleveland (1993) recommends that the aspect ratio of line plots is chosen so that lines are ‘banked’ at $45°$. By this he means that the averaged absolute value of the slope should be around $\pm 45°$. Write a function to achieve this for a time-series plot, and try it out on the `sunspots` dataset. (Average along the arc length of the curve. See the function `banking` for Cleveland’s solution.) Answer

3.5 The Trellis function `splom` produces a complete matrix of scatterplots, as does the basic plotting functions `pairs`, but in earlier versions of S-PLUS `pairs` only plotted the lower triangle of the matrix. Write a function to emulate the earlier behaviour. [Hint: look at `pairs.default`. The graphics parameter `mfg` may be useful.] Answer

3.6 *Ternary* plots are used for compositional data (Aitchison, 1986) where there are three components whose proportions add to one. These are represented by a point in an equilateral triangle, where the distances to the sides add to a constant.

Write an S function to plot a matrix of compositions on a ternary diagram. Apply this to the dataset `Skye` on the composition of rocks on the Isle of Skye in Scotland. (Our solution can be found on the help page for this dataset, and S-PLUS 99 has a GUI-graphics example in its `samples` directory.) Answer

Chapter 5

5.1 Rice (1995, p. 390) gives the following data (Natrella, 1963) on the latent heat of the fusion of ice (*cal/gm*):

```
Method A: 79.98 80.04 80.02 80.04 80.03 80.03 80.04 79.97
          80.05 80.03 80.02 80.00 80.02
Method B: 80.02 79.94 79.98 79.97 79.97 80.03 79.95 79.97
```

(a) Assuming normality, test the hypothesis of equal means, both with and without making the assumption of equal variances.

Compare the result with a Wilcoxon/Mann–Whitney nonparametric two-sample test.

(b) Inspect the data graphically in various ways, for example, boxplots, Q-Q plots and histograms.

(c) Fit a one-way analysis of variance and compare it with your t-test. (Look ahead to the next chapter, or investigate function `oneway`.)

5.2 Write functions to produce Q-Q plots for a gamma and a Weibull distribution. Note that unlike the normal Q-Q plot, the shape parameters may need to be estimated. Answer

5.3 The ideas used in bandwidth selection for kernel density estimation that are implemented in `width.SJ` can also be applied to the choice of bin width in a histogram (Wand, 1997). Implement such a bin-width estimator in S. Answer

5.4 Experiment with dataset `galaxies`. How many modes do you think there are in the underlying density?

Chapter 6

6.1 Apply regression diagnostics to the fits to the `whiteside` energy consumption data. Note (Figure 6.1) that the evidence for a quadratic fit to the ‘after’ data stems from points with high leverage, so also try resistant fits.

6.2 Dataset `cabbages` gives the results of a field trial on the growth of cabbages (Rawlings, 1988, p. 219). Analyse this trial.

6.3 The data frame `rubber` in the library `MASS` gives 30 measurements of rubber loss under accelerated testing together with the hardness and tensile strength of the rubber itself. Explore the data in `brush`, then fit linear and quadratic regressions of `loss` on `hard` and `tens`. Select a suitable submodel of the quadratic model, and inspect the fitted surface by a perspective plot.

6.4 Criminologists are interested in the effect of punishment regimes on crime rates. This has been studied using aggregate data on 47 states of the USA for 1960, available in data frame `UScrime` (Ehrlich, 1973; Vandaele, 1978; Raftery, 1995). The response variable is the rate of crimes in a particular category per head of population. There are 15 explanatory variables; most of these and the response variable have been rescaled to convenient numbers.

(a) Analyse these data. In your report pay particular attention to how your model was selected.

(b) Comment on the effect of the last two explanatory variables in relation to the criminologists' interest in the effect of punishment.

(c) Comment critically on the assumptions needed to draw conclusions from aggregate studies such as this.

6.5 Susan Prosser collected data on the concentration of a chemical GAG in the urine of 314 children aged from 0 to 17 years. The data are in data frame `GAGurine`. Analyse these data, and produce a chart to help a paediatrican to assess if a child's GAG concentration is 'normal'.

6.6 The Janka hardness data in data frame `janka` gives the density (`Dens`) and hardness (`Hard`) of a sample of Australian Eucalypt hardwoods. The problem is to build a prediction equation for hardness in terms of density.

6.7 The `Cars93` data frame (Lock, 1993) gives data on 93 new car models on sale in the USA in 1993. Use this dataset to predict fuel consumption from the remaining variables. [Hint: The fuel consumption is in miles per US gallon. In metric units fuel consumption is expressed in litres/100km, a reciprocal scale.]

6.8 The data in Table 6.4 (from Scheffé, 1959, and in data frame `genotype`) refer to rat litters that were separated from their natural mothers at birth and given to foster mothers to rear. The rats were classified into one of four genotypes, `A`,

Table 6.4: The rat `genotype` data.

	Foster mother			
Litter	A	B	I	J
A	61.5	55.0	52.5	42.0
	68.2	42.0	61.8	54.0
	64.0	60.2	49.5	61.0
	65.0		52.7	48.2
	59.7			39.6
B	60.3	50.8	56.5	51.3
	51.7	64.7	59.0	40.5
	49.3	61.7	47.2	
	48.0	64.0	53.0	
		62.0		
I	37.0	56.3	39.7	50.0
	36.3	69.8	46.0	43.8
	68.0	67.0	61.3	54.5
			55.3	
			55.7	
J	59.0	59.5	45.2	44.8
	57.4	52.8	57.0	51.5
	54.0	56.0	61.4	53.0
	47.0			42.0
				54.0

`B`, `I` and `J`. The response is the litter average weight gain, in grams, over the time of the study. The aim is to test whether the litters' and mothers' genotypes act additively and if this may be retained to test for differences in litter and mother genotype effects.

6.9 Extend the analysis of the `coop` dataset to all the specimens.

6.10 Add the fitted lines for the final model for the `petrol` data to Figure 6.9.

6.11 Find a way to plot the `Sitka` data that facilitates comparison of the growth curves for the two treatment groups.

Add to your plot the fitted mean growth curve and some 95% confidence intervals.

6.12 Consider how to explore the assumptions made for the `lme` model for the `Sitka` data. Are the `plot` methods for `lme` objects helpful in this?

6.13 The object `Sitka89` contains the 1989 data on the same 79 Sitka trees measured on eight days in 1989. Analyse the 1989 data separately, and then in conjunction with the 1988 data.

Programming exercises

6.14 How do you obtain the standard prediction and confidence intervals for a linear model fitted by `lm`? Answer

6.15 How can we we add a confidence or prediction region to an existing plot of a simple linear regression?

As an example, add a prediction region to Figure 6.1. Answer

6.16 Write a function to fit a linear model by generalized least squares, that is to minimize

$$(\boldsymbol{y} - X\boldsymbol{\beta})^T W(\boldsymbol{y} - X\boldsymbol{\beta})$$

for a given symmetric positive definite matrix W, or given $\Sigma = W^{-1}$. Answer

6.17 Implement a ridge regression (Brown, 1993, Sen & Srivastava, 1990) function in S. Answer

Chapter 7

7.1 Explore the `anorexia` data example introduced in the discussion of offsets on page 217 and report your final linear model.

Begin with a Trellis display of the data showing post-treatment weight against pre-treatment weight for the three treatment groups. In each panel include the individual regression line, the parallel regression line and the parallel regression line with slope 1 as well as the points.

7.2 Analyse the `menarche` dataset on the proportions of female children in Warsaw at various ages during adolescence who have reached menarche (Milicer & Szczotka, 1966) using both logit and probit links.

7.3 Knight & Skagen (1988) collected the data shown in the table (and in data frame `eagles`)during a field study on the foraging behaviour of wintering Bald Eagles in Washington State, USA. The data concern 160 attempts by one (pirating) Bald Eagle to steal a chum salmon from another (feeding) Bald Eagle. The abbreviations used are

L = large S = small; A = adult I = immature

Number of successful attempts	Total number of attempts	Size of pirating eagle	Age of pirating eagle	Size of feeding eagle
17	24	L	A	L
29	29	L	A	S
17	27	L	I	L
20	20	L	I	S
1	12	S	A	L
15	16	S	A	S
0	28	S	I	L
1	4	S	I	S

Report on factors that explain the success of the pirating attempt, and give a prediction formula for the probability of success.

7.4 The following data are part of a survey by Dr Mutch of low-weight births in Scotland between 1981 and 1988. The table refers to 661 children with birth weights between 650g and 1749g all of whom survived for at least one year. The variables of interest are:

Cardiac: mild heart problems of the mother during pregnancy;

Comps: gynaecological problems during pregnancy;

Smoking: mother smoked at least one cigarette per day during the first 6 months of pregnancy;

BW: was the birth weight less than 1250g?

Cardiac		Yes				No			
Comps		Yes		No		Yes		No	
Smoking		Yes	No	Yes	No	Yes	No	Yes	No
BW	Yes	10	25	12	15	18	12	42	45
	No	7	5	22	19	10	12	202	205

Analyse this table.

7.5 A survey was made of bicycle and other traffic in the neighbourhood of the Berkeley campus of the University of California in 1993 (Gelman *et al.*, 1995, p. 91). Sixty city streets were selected at random, with a stratification into three levels of activity and whether the street had a marked bicycle lane. The counts observed in one hour are shown in the table: for two of the streets the data were lost.

Type of	Bike											
street	lane?		Counts									
Residential	yes	bikes	16	9	10	13	19	20	18	17	35	55
		other	58	90	48	57	103	57	86	112	273	64
Residential	no	bikes	12	1	2	4	9	7	9	8		
		other	113	18	14	44	208	67	29	154		
Side	yes	bikes	8	35	31	19	38	47	44	44	29	18
		other	29	415	425	42	180	675	620	437	47	462
Side	no	bikes	10	43	5	14	58	15	0	47	51	32
		other	557	1258	499	601	1163	700	90	1093	1459	1086
Main	yes	bikes	60	51	58	59	53	68	68	60	71	63
		other	1545	1499	1598	503	407	1494	1558	1706	476	752
Main	no	bikes	8	9	6	9	19	61	31	75	14	25
		other	1248	1246	1596	1765	1290	2498	2346	3101	1918	2318

Report on these data, paying particular attention to the effects of bicycle lanes.

7.6 To study the relative survival capacities of two species of native and exotic snails, here labelled A and B, groups of 20 animals were held in controlled laboratory conditions for periods of 1, 2, 3 or 4 weeks. At the end of the period the animals were checked for whether they had survived, but as the check itself is a destructive process a longitudinal study with the same animals was not possible. The groups were held in chambers where the temperature and relative humidity were held fixed at three and four levels ,respectively. There were thus $2 \times 4 \times 3 \times 4 = 96$ groups laid out in a complete factorial design.

The data are shown in Table 7.5, where each entry is the number who did not survive out of the 20 test animals. The data set is also available as the data frame `snails` in library `MASS`. Variable `Species` is a two-level factor but treat the other stimulus variables as quantitative.

(a) Fit separate logistic regression models on exposure, relative humidity and temperature for each species, that is ,a logistic regression of the form `Species/(Exposure + Rel.Hum + Temp)`.

(b) Fit parallel logistic regressions for the two species on the three stimulus variables and show that it may be retained when tested within the separate regressions model.

Table 7.5: The snail mortality data.

		Species A				Species B			
		Exposure				Exposure			
Rel. Hum.	Temp. (° C)	1	2	3	4	1	2	3	4
60.0%	10	0	0	1	7	0	0	7	12
	15	0	1	4	7	0	3	11	14
	20	0	1	5	7	0	2	11	16
65.8%	10	0	0	0	4	0	0	4	10
	15	0	1	2	4	0	2	5	12
	20	0	0	4	7	0	1	9	12
70.5%	10	0	0	0	3	0	0	2	5
	15	0	0	2	3	0	0	4	7
	20	0	0	3	5	0	1	6	9
75.8%	10	0	0	0	2	0	1	2	4
	15	0	0	1	3	0	0	3	5
	20	0	0	2	3	0	1	5	7

(c) There are no deaths for either species for the 1 week exposure time. This suggests a quadratic term in `Exposure` might be warranted. Repeat the analysis including such a quadratic term.

(d) Because deaths are so sparse a residual analysis is fairly meaningless. Nevertheless look at the residuals to see how they appear for this kind of data set.

(e) Is there a significant difference between the survival rates of the two species? Describe qualitatively how the probability of death depends upon the stimulus variables. Summarize your conclusions.

7.7 An experiment was performed in Sweden in 1961–2 to assess the effect of speed limits on the motorway accident rate (Svensson, 1981). The experiment was conducted on 92 days in each year, matched so that day j in 1962 was comparable to day j in 1961. On some days the speed limit was in effect and enforced, whereas on other days there was no speed limit and cars tended to be driven faster. The speed limit days tended to be in contiguous blocks.

The data set is given in the data frame `Traffic` with factors `year`, `day` and `limit` and the response is the daily traffic accident count `y`.

Fit Poisson log-linear models and summarize what you discover.

You might assume `day` occurs as a main effect only (fitting models with interaction terms involving factors of 92 levels may take some time and memory!), but assess if an interaction between `limit` and `year` is needed.

Check if the deviance residuals provide any hint of irregular behaviour.

7.8 The data given in data frame `Insurance` consist of the numbers of policyholders n of an insurance company who were exposed to risk, and the numbers of car insurance claims made by those policyholders in the third quarter of 1973 (Baxter, Coutts & Ross, 1980, Aitkin *et al.*, 1989. The data are cross-classified by `District` (four levels), `Group` of car (four levels), and `Age` of driver (four ordered levels). The other variables in the data frame are the numbers of `Holders` and `Claims`.

The relevant model is taken to be a Poisson log-linear model with `offset` $\log n$.

(a) Fit an initial model with all terms present up to the three-way interaction; that is,

```
Claims ~ District*Group*Age - District:Group:Age
         + offset(log(Holders))
```

(b) Using `stepAIC`, or otherwise, prune the model of unjustified terms and report your findings.

Present your results as a table of estimated claim rates per policy holder for each category of holder.

(c) It is not strictly valid to regard such data as having the obvious binomial distribution, since some policyholders may make multiple claims. Nevertheless it should be a reasonable approximation. Repeat the analysis with a binomial model and compare the outcomes on estimated claim rates (or in this case, estimated probabilities of making a claim).

Chapter 8

8.1 For the weight loss example compare the negative exponential model with quadratic and cubic polynomial regression alternative models, in particular check the behaviour of each model under extrapolation into the future. Answer

8.2 Fit the negative exponential weight loss model in the ‘goal weight’ form, equation (8.6), for the three goal weights, $w_0 = 110,\ 100$ and $90\,\text{kg}$. Plot the profiles. Answer

8.3 The model used in connection with the Stormer data may also be expressed as a generalized linear model. To do this we write

$$\frac{\beta_1 v}{w - \beta_2} = \frac{1}{\gamma_1 z_1 + \gamma_2 z_2}$$

where $\gamma_1 = 1/\beta_1$, $\gamma_2 = \beta_2/\beta_1$, $z_1 = w/v$ and $z_2 = -1/v$. This has the form of a generalized linear model with inverse link. Fit the model in this form using a `quasi` family with `inverse` link and `constant` variance function.

Back transform the estimated coefficients and show that they agree with the values obtained using the non-linear regression approach.

Also compute the estimated standard errors and verify that they also agree with the values obtained directly by the non-linear regression approach.

Finding the standard errors is more challenging. We need first to find the variance matrix from the generalized linear model. The large sample variance matrix for $\widehat{\boldsymbol{\beta}}$ is related to that for $\widehat{\boldsymbol{\gamma}}$ by $\operatorname{var}\left(\widehat{\boldsymbol{\beta}}\right) = J\operatorname{var}\left(\widehat{\boldsymbol{\gamma}}\right) J^T$ where J is the Jacobian matrix of the inverse of the parameter transformation:

$$J = \begin{bmatrix} \partial\beta_1/\partial\gamma_1 & \partial\beta_1/\partial\gamma_2 \\ \partial\beta_2/\partial\gamma_1 & \partial\beta_2/\partial\gamma_2 \end{bmatrix} = \begin{bmatrix} -1/\gamma_1^2 & 0 \\ -\gamma_2/\gamma_1^2 & 1/\gamma_1 \end{bmatrix}$$

(To achieve close agreement you may need to tighten the convergence criteria for the `glm` fit, for example, by setting `eps=1.0e-10`.) Answer

8.4 *Stable parameters*

Ross (1970) has suggested using *stable parameters* for non-linear regression, mainly to achieve estimates that are as near to uncorrelated as possible. It turns out that in many cases stable parameters also define a coordinate system within the solution locus with a small curvature.

The idea is to use the means at p well-separated points in sample space as the parameters. Writing the regression function in terms of the stable parameters is often intractable, but in the case of a negative exponential decay model of the type we considered for the weight loss data it is possible if the points are chosen equally spaced.

If the three mean parameters μ_i are chosen at x-points $x_0 + i\delta_x$, $i = 0, 1, 2$, show that the model may be written explicitly as:

$$\eta = \frac{\mu_0\mu_2 - \mu_1^2}{\mu_0 - 2\mu_1 + \mu_2} + \frac{(\mu_0 - \mu_1)^2}{\mu_0 - 2\mu_1 + \mu_2}\left(\frac{\mu_1 - \mu_2}{\mu_0 - \mu_1}\right)^{(x-x_0)/\delta_x}$$

Fit the negative exponential decay model to the weight loss data using this parametrization and choosing, say, $x_0 = 40$ days and $\delta_x = 80$ days. Look at the characteristics of the fit, including the correlations between the parameter estimates. Explain in heuristic terms why they are relatively low.

Examine the profiles of the fit and check for straightness. Can you give a possible statistical explanation for why they appear as straight as they do? Answer

8.5 *Heteroscedastic regression models*

A common heteroscedastic regression model specifies that the observations have constant coefficient of variation; that is ,$Y \sim \mathrm{N}(\mu, \theta\mu^2)$ where $\theta > 0$ and μ depends on regressor variables according to some linear model perhaps with a link function such as $\mu = \exp\eta$. Write a function to fit such models and try it out using the Quine data. Compare with the negative binomial models fitted in Section 7.4, page 234ff.

8.6 A deterministic relationship between pressure and temperature in saturated steam can be written as

$$\text{Pressure} = \alpha \exp\left(\frac{\beta T}{\gamma + T}\right)$$

where T is the temperature, considered the determining variable. Data collected to estimate the unknown parameters α, β and γ are contained in the data frame `steam`.

(a) Fit this model as a non-linear regression assuming additive errors in the pressure scale. Devise a suitable method for arriving at initial values.

(b) Fit the model again, this time taking logarithms of the relationship and assuming that the errors are additive in the log(pressure) scale, (and hence multiplicative on the original scale).

(c) Which model do you consider is better supported on the basis of model checks?

8.7 The data frame `Puromycin` supplied with S-PLUS contains data from a Michaelis–Menten experiment conducted at two levels of a factor `state`. The usual non-linear regression model proposed for such cases is

$$V_{ij} = \frac{K_j c_{ij}}{c_{ij} + \theta_j} + \varepsilon_{ij}, \qquad j = 1, 2$$

where V is the (initial) velocity of the reaction, c is the substrate concentration and j refers to the level of the factor.

Fit the model with separate asymptotes V_j and separate θ_js. Test the hypothesis that $\theta_1 = \theta_2$ and report. [Hint: use the `anova` method for `nls` objects.]

For the final model you adopt show the data and fitted curves in a two-panel Trellis display.

8.8 Fit the non-linear model (8.4) to the `muscle` data with (α_j) as a random effect; that is ,a mean asymptote plus one variance component. Compare the predictions (both mean and BLUP curves) of this model with the fixed-effects model fitted in the text.

8.9 Sarah Hogan collected data on the ‘binaural hearing’ ability of children with a history of otitis media with effusion (OME). Some of the data (and a description of the problem) are in data frame `OME`. Fit a suitable non-linear model, and assess if there is a change in ability with age and OME status.

(a) The suggested model is a logistic curve that ranges from 0.5 at low noise levels (when the response is effectively a guess) to 1.0 at high noise levels. Then the most important parameter will be the noise level `L75` at which the child has a 75% success rate. The amount of data on each child is small, so fit a model with a common slope but a separate `L75` for each child, and analyse the fitted parameters by age and group. [You may want to look up the function `nlsList`.]

(b) Consider a linear model for `L75` on age, and differences between the OME groups, for each type of noise stimulus. Assess the significance of your results via standard errors and/or F-tests.

(c) The analysis thus far does not take into account the differences between individual subjects. Repeat the analysis using non-linear mixed-effects models.

Answer

8.10 Write a function to fit a gamma distribution to n observations by maximum likelihood.

Answer

8.11 McLachlan & Jones (1988) (see also McLachlan & Krishnan, 1997, pp. 73ff) give the following grouped data on red blood cell volume, in 18 equally spaced bins of width 7.2 fl, starting at 21.6 fl.

```
Set 1: 10 21 51 77 70 50 44 40 46 54 53 54 44 36 29 21 16 13
Set 2:  9 32 64 69 56 68 88 93 87 67 44 36 30 24 21 14  8  7
```

McLachlan and Jones fit a mixture of two normal densities on log scale by an involved method using the EM algorithm. Fit this model directly to each set of data by a small modification of the approach in Section 8.7.

Further exercise

8.12 A deterministic relationship between pressure and temperature in saturated steam can be written as

$$\text{Pressure} = \alpha \exp\left(\frac{\beta T}{\gamma + T}\right)$$

where T is the temperature, considered the determining variable. Data collected to estimate the unknown parameters α, β and γ is contained in the data frame `steam`.

(a) Fit this model as a non-linear regression assuming additive errors in the pressure scale. Devise a suitable method for arriving at initial values.

(b) Fit the model again, this time taking logarithms of the relationship above and assuming that the errors are additive in the log(pressure) scale, (and hence multiplicative on the original scale).

(c) Which model do you consider is better supported on the basis of model checks?

Chapter 9

9.1 This data frame `gilgais` was collected on a line transect survey in gilgai territory in New South Wales, Australia. Gilgais are natural gentle depressions in otherwise flat land, and sometimes seem to be regularly distributed. The data collection was stimulated by the question: are these patterns reflected in soil properties? At each of 365 sampling locations on a linear grid of 4 metres spacing, samples were taken at depths 0–10 cm, 30–40 cm and 80–90 cm below the surface. pH, electrical conductivity and chloride content were measured on a 1:5 soil:water extract from each sample.

Produce smoothed maps of the measurements.

9.2 Exercise 6.5 considered linear regression for the GAG in urine data in data frame `GAGurine`. Consider using a non-linear or smooth regression for the same task.

9.3 Use neural networks to fit a smooth curve to the `mcycle` data used in Figure 9.1. Investigate ways of choosing the degree of smoothness automatically.

Chapter 11

11.1 Data frame `biopsy` contains data on 699 biopsies of breast tumours, which have been classified as benign or malignant (Mangasarian & Wolberg, 1990). The nine variables on each biopsy are a rating (1 to 10) by the coordinating physician; ratings on one variable are missing for some biopsies.

Analyse these data. In particular, investigate the differences in the two types of tumour, find a rule to classify tumours based solely on the biopsy variables and assess the accuracy of your rule.

11.2 Data frame `UScereals` describes 65 commonly available breakfast cereals in the USA, based on the information available on the mandatory food label on the packet. The measurements are normalized to a serving size of one American cup.

(i) Is there any way to discriminate among the major manufacturers by cereal characteristics, or do they each have a balanced portfolio of cereals?

(ii) Are there interpretable clusters of cereals?

(iii) Can you describe why cereals are displayed on high, low or middle shelves?

Chapter 13

13.1 Our dataset `accdeaths` gives monthly accidental deaths in the USA 1973–8, from Brockwell & Davis (1991). Find a suitable ARIMA model, and predict the deaths for the first six months of 1979. Answer

13.2 Dataset `austres` is a quarterly series of the number of Australian residents from March 1971 to March 1994. It comes from Brockwell & Davis (1996) who analyse the percentage quarterly changes. Explore suitable models in S-PLUS. Answer

13.3 Repeat Exercise 9.1 as a time series problem.

13.4 Use the information gained in the analysis of `beav1` in Section 13.5 to refine the analysis for `beav2`.

13.5 Consider the problem of estimating the effect of seat belt legislation on road accident casualties in the UK considered by Harvey & Durbin (1986). The data (from Harvey, 1989) are in the series `drivers`.

Chapter 14

14.1 Repeat exercise 9.1 as a spatial statistics problem.

Answers to Selected Exercises

Chapter 2

2.1. One way is to use

```
find.val <- function(x, val) seq(along=x)[x==val]
row.names(hills)[find.val(hills$climb, 2100)]
```

although in most cases it is easier to subscript directly by a logical vector, for example `row.names(hills)[hills$climb==2100]`.

2.2. Three solutions are:

```
res <- factor(ftv); levels(res)[-(1:2)] <- "2 or more"
res <- cut(ftv, c(-1,0,1, 10))
     levels(res) <- c("0", "1", "2 or more")
merge.levels(factor(ftv), c(1,2,3,3,3,3))
```

where the first is explained in the help page for `merge.levels`.

2.3. We used

```
mad <- function(y, mu=median(y))
           median(abs(as.vector(y)-mu))
```

where `as.vector` strips off the name attribute which `median` retains in some versions of S-PLUS. (Note that the system function by default calculates $1.4826\times$ median$|x-\mu|$ which is a consistent estimator of the standard deviation for a Gaussian model.)

2.4. The idea is to find any indices where the strings in `out` match the names of `x` and to use their negatives as an index vector. Matching is such a common problem there is a general function, `match`, to do it.

```
x.in <- x[-match(out, names(x), nomatch=0)]
```

Note the use of `nomatch=0` to generate a zero index (and hence no action) if some string in `out` is not the name of any component in `x`.

This solution relies on the uniqueness of the names of the object (which is not guaranteed in all instances), since `match` will find only the first match. An alternative approach is to match the names in `out` and use logical indexing, by

```
x.in <- x[match(names(x), out, nomatch=0) == 0]
```

2.5. All that is needed is

```
X[rep(1:nrow(X), w), ]
```

2.6. Try this:

```
r <- cor(X)
rc <- format(r)
rc[r < 0.9] <- ""
print(rc, quote=F)
```

Note the use of `format` to get consistent formating of the entries; `format` could also be used to prune the number of significant digits, if required.

2.7. The following solution extends exercise 2.4.
Use `match` and the indexing capabilities. If the data frame is `df` and the variable is `ethnic` the subset you want is

```
df[match(df$ethnic, c(1,3,4,6,7), nomatch=0) > 0, ]
```

The function `is.element` of S-PLUS 4.0 and later implements this idea as a function.

```
is.element <-
    function(el, set) !is.na(match(el, set, nomatch = NA))
df[is.element(df$ethnic, c(1,3,4,6,7)), ]
```

This is an alternative way to use `match` which has the advantage here of working even when the first argument is empty.

Chapter 3

3.1. To create a barchart of `Exer` we just use `plot(Exer)`, or

```
barplot(table(Exer), names=names(table(Exer)))
```

(Try them to see the differences.) For a pie chart, we need to tabulate the frequencies first:

```
exer.freq <- table(Exer)
exer.freq
 Freq Some None
  115   98   24
```

The command `pie(exer.freq)` will now create a pie chart, but to add labels to the slices we use the `names` argument

```
pie(exer.freq, names=levels(Exer))
```

Adding a legend is accomplished by using `legend` with the `fill` argument:

```
legend(locator(1), names(exer.freq), fill=1:3)
```

For the `Smoke` variable a slightly different approach is needed if we wish to include the missing value in the plot.

```
smoke.freq <- summary(Smoke)
smoke.freq
 Heavy Regul Occas Never NA's
    11    17    19   189    1
```

Since the missing value represents such a small proportion of the data, we highlight it with `explode=5` (because `NA's` is the fifth category) so it is not lost in the pie:

```
pie(smoke.freq, names=names(smoke.freq), explode=5)
legend(locator(1), names(smoke.freq), fill=1:5)
```

Alternatives using Trellis graphics are

```
barchart(~ exer.freq, main="Exercise frequency")
piechart(~ exer.freq, main="Exercise frequency")
piechart(~ smoke.freq, explode = 5)
```

Adding legends and other annotations is left as a further exercise for the reader.

3.2. This is straightforward once the layout is adjusted. We just increased the sizes of the margins which are to hold text.

```
ir <- rbind(iris[,,1], iris[,,2], iris[,,3])[, 3:4]
irs <- c(rep("S", 50), rep("C", 50), rep("V", 50))
par(mar=c(7,7,7,5)) # more space on label sides
plot(ir, type="n", cex=2, lwd=2, tck=-0.02)
title("The Iris Data", cex=2)
text(ir, irs, col=c(rep(2,50), rep(3,50), rep(4, 50)))
```

On-screen the title size is limited by the displayable fonts under the `motif` driver (and probably others).

3.3. Our solution was

```
x <- seq(-pi, pi, length=200)
plot(x, sin(x), type="l", axes=F, ylab="", main="sin(x)")
axis(2, pos=0, yaxs="e", las=1)
axis(1, pos = -1.1, at = pi*seq(-1, 1, 1/4), tck = -0.02,
   labels = c("-Pi", "-3Pi/4", "-Pi/2", "-Pi/4", "0",
              "Pi/4", "Pi/2", "3Pi/4", "Pi"))
```

3.4. We apply this to the annual mean sunspot series. The function `bank` (based on `banking`) computes an aspect ratio `ar` such that the average slope of the line segments `(dx, ar*dy)` is 45°. We adjust the aspect ratio by reducing the size of one of the axes of the plot.

```
sunsp <- aggregate(sunspots, 1, mean)

bank <- function(dx, dy, iter = 20, tol = 0.5)
{
  dx <- abs(dx)
  dy <- abs(dy)[dx > 0]
  dx <- dx[dx > 0]
  mad <- median(dy/dx)
  ar <- ifelse(mad > 0, mad, 1)
  radians.to.angle <- 180/pi
  for(i in 1:iter) {
    distances <- sqrt(dx^2 + (ar * dy)^2)
    orientations <- atan(ar * dy, dx)
    avg <- (radians.to.angle * sum(orientations *
              distances))/sum(distances)
    if(abs(45 - avg) < tol) break
    ar <- ar * (45/avg)
  }
  ar
}

bankplot <- function(x, y, type="l", ...)
{
  dx <- diff(x/diff(range(x)))
  dy <- diff(y/diff(range(y)))
  ar <- bank(dx, dy)
  pin <- par("pin")
  ar <- ar/(pin[2]/pin[1])
  oldpar <- par(pin=pin*c(1,ar)/max(1,ar))
  on.exit(par(oldpar))
  plot(x, y, type=type, ...)
}

bankplot(time(sunsp), sunsp)
```

3.5. The following function is based closely on `pairs.default`. We use `mfg` to choose which panel to fill in a $n \times n$ grid. As we only ever write to the panels, we need to clear the plot first with a call to `frame`.

```
mypairs <- function(x, labels = dimnames(x)[[2]],
                     panel = points, ...)
{
  doaxis <- function(which, dolabel = T)
    axis(which, outer = T, line = -0.5, labels = dolabel)
  setup <- function(x, y, ...)
    .S(plot(range(x[!is.na(x)]), range(y[!is.na(y)]),
            type = "n", axes = F,  ...), "plot")
  x <- as.matrix(x)
  if(is.character(panel)) panel <- get(panel, mode="function")
  n <- ncol(x)
  oldpar <- par("oma", "mar", "cex", "tck", "mfg", "mgp",
                "mex", "mfrow")
  oldcex <- par("cex")
  CEX <- oldcex * max(7.7/(2 * n + 3), 0.6)
  par(mfrow = c(n, n), mgp = c(2, 0.80, 0), oma = rep(3, 4),
      mar = rep(0.5, 4), tck = -0.03/n)
  on.exit({par(oldpar)})
  par(cex = CEX)
  frame()
  if(length(labels) < n)
    labels <- paste(deparse(substitute(x)),
                    "[,", 1:n, "]", sep = "")
  if(par("pty") == "s") {
    dif <- diff(par("fin"))/2
    if(dif > 0) par(omi = c(dif*n, 0, dif*n, 0) + par("omi"))
    else par(omi = c(0, -dif*n, 0, -dif*n) + par("omi"))
  }
  for(i in 1:n)
    for(j in 1:i) {
      par(mfg = c(i,j,n,n))
      setup(as.vector(x[, j]), as.vector(x[, i]), ...)
      box()
      if(i == n && j < n) doaxis(1)
      if(j == 1 && i > 1) doaxis(2)
      if(i > j) {
          panel(as.vector(x[, j]), as.vector(x[, i]), ...)
      } else {
        par(usr = c(0, 1, 0, 1))
        text(0.5, 0.5, labels[i], cex = 1.5 * CEX)
      }
    }
  invisible()
}
```

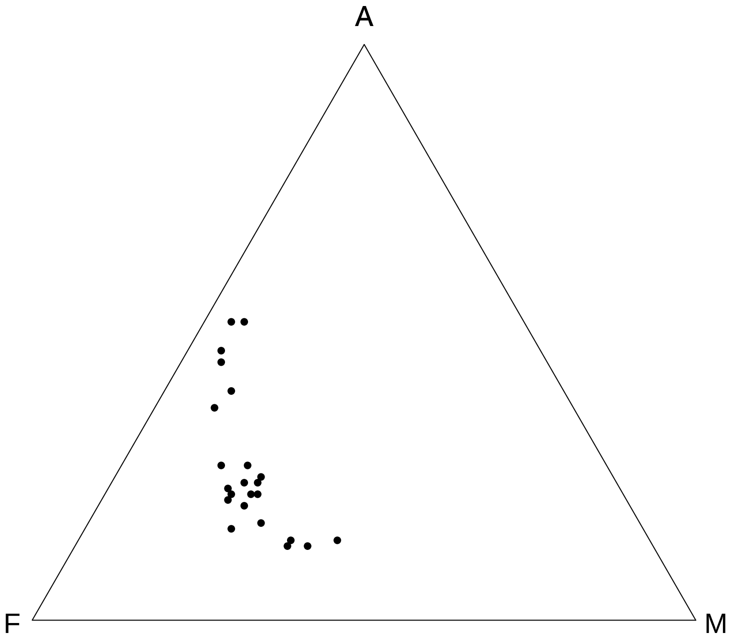
3.6. As a precaution, we rescale the entries in `X` to sum to one.

```
ternary <- function(X, pch = par("pch"), lcex = 1,
                    add = F, ord = 1:3, ...)
{
  if(any(X) < 0) stop("X must be non-negative")
  s <- drop(X %*% rep(1, ncol(X)))
  if(any(s<=0)) stop("each row of X must have a positive sum")
  if(max(abs(s-1)) > 1e-6) {
    warning("row(s) of X will be rescaled")
    X <- X / s
  }
  X <- X[, ord]
  s3 <- sqrt(1/3)
  if(!add)
  {
    oldpty <- par("pty")
    on.exit(par(pty=oldpty))
    par(pty="s")
    plot(c(-s3, s3), c(0.5-s3, 0.5+s3), type="n", axes=F,
         xlab="", ylab="")
    polygon(c(0, -s3, s3), c(1, 0, 0), density=0)
    lab <- NULL
    if(!is.null(dn <- dimnames(X))) lab <- dn[[2]]
    if(length(lab) < 3) lab <- as.character(1:3)
    eps <- 0.05 * lcex
    text(c(0, s3+eps*0.7, -s3-eps*0.7),
         c(1+eps, -0.1*eps, -0.1*eps), lab, cex=lcex)
  }
  points((X[,2] - X[,3])*s3, X[,1], ...)
}
```

This labels the vertices clockwise from the top, but other conventions are possible by altering the argument `ord`. For example, we can reproduce Fig. 1.9 of Aitchison (1986) by

```
    ternary(Skye/100, ord=c(1,3,2))
```

as



Chapter 5

5.2. The answers if the shape parameters are known are easy using `ppoints`.

```
qqgamma <- function(x, shape, ...)
  plot(qgamma(ppoints(x), shape), sort(x), ...)
qqweibull <- function(x, shape, ...)
  plot(qweibull(ppoints(x), shape), sort(x), ...)
```

To fit a gamma we can use the function `gamma.mle1` of the answer to exercise 8.10, by

```
qqgamma <- function(x, shape = gam.mle(x),
  xlab = paste("Quantiles of gamma(",
             format(shape, digits=3), ")", sep=""),
  ylab = deparse(substitute(x)), ...)
{
  gam.mle <- function(x) gamma.mle1(x)$alpha
  plot(qgamma(ppoints(x), shape), sort(x),
       xlab=xlab, ylab=ylab, ...)
}
```

For a Weibull we can fit using `survreg`, converting from its parametrization to a more standard one.

```
qqweibull <- function(x, shape = wei.shape(x),
  xlab = paste("Quantiles of Weibull(",
              format(shape, digits=3), ")", sep=""),
  ylab = deparse(substitute(x)), ...)
{
  wei.shape <- function(x) 1/survreg(Surv(x) ~ 1)$scale
## with S-PLUS 2000 use survReg
## with survival4 use
# wei.shape <- function(x) exp(-survreg(Surv(x) ~ 1)$parms)
  plot(qweibull(ppoints(x), shape), sort(x),
       xlab=xlab, ylab=ylab, ...)
}
```

(For use with `survival5` replace `parms` by `coef`) It is possible to avoid estimating the shape parameter in this case, as a QQ-plot of any Weibull against a Weibull(1,1) is a straight line *on a log-log scale*. Thus it is possible to assess the fit of a Weibull (of any shape) by `qqweibull(x, 1, log="xy")`. In any case, a log-log plot is desirable for small values (less than 0.5) of the shape parameter as those distributions have a very long right tail.

It is easy to produce Trellis versions of these plots using `qqmath`, with a common shape parameter across panels.

5.3. See the function `dpi` in Wand's library `KernSmooth`.

Chapter 6

6.14. Here is an example for the `hills` dataset of how to find the confidence interval for the fit at each data point.

```
hills.lm <- lm(time ~ dist + climb, data=hills)
hills.pred <- predict(hills.lm, se.fit = T)
hills.ci <- pointwise(hills.pred, coverage = 0.95)
```

The prediction interval is a little trickier. The simplest idea is to add s^2 to the squared standard errors returned by `predict`, noting that s^2 has in fact been stored already.

```
hills.s <- summary(hills.lm)$sigma
hills.pred$se.fit <- sqrt(hills.pred$se.fit^2 +
                       hills.pred$residual.scale^2)
hills.ci <- pointwise(hills.pred, coverage = 0.95)
```

6.15. Most of the work was done in the previous exercise. We will try this out on the data priot to insulation.

```
before <- whiteside[whiteside$Insul=="Before",]
before.lm <- lm(Gas ~ Temp, data=before)
attach(before)
plot(Temp, Gas)
abline(before.lm, lty=3)

conflines.lm <- function(obj, coverage = 0.95, pred = F, ...)
{
# Check for simple linear regression
  xnames <- attr(obj$terms,"term.labels")
  if(length(xnames) != 1)
    stop("First argument is not a simple linear fit")
# Work out the range of the existing plot.
  ux <- par("usr")[1:2]
  xp <- seq(ux[1], ux[2], length = 100)
  newdf <- data.frame(xp)
  names(newdf) <-  xnames
  pr <- predict(obj, newdf, se.fit = T)
  if(pred) {
    pr$se.fit <- sqrt(pr$se.fit + pr$res^2)
  }
  ci <- pointwise(pr, coverage = coverage)
  lines(xp, ci$lower, ...)
  lines(xp, ci$upper, ...)
}
conflines.lm(before.lm)
conflines.lm(before.lm, pred=T, lty=2)
detach()
```

Figure 6.1 is a Trellis plot, so we cannot add information to it; rather we have to create a new Trellis plot by adding to the panel function. We could do this by operating on the data for each panel, but we will illustrate a more general solution, which allows the pooling of standard errors between the sexes.

```
gas.lm <- lm(Gas ~ Insul/Temp  - 1, data=whiteside)
pr <- predict(gas.lm, se=T)
gas.ci <- pointwise(pr)
pr$se.fit <- sqrt(pr$se.fit + pr$res^2)
gas.ti <- pointwise(pr)

prepanel.gas <- function(x, y, subscripts, ...)
{
  xlim <- range(x)
  ylim <- range(y, gas.ti$fit[subscripts],
                gas.ti$upper[subscripts],
                gas.ti$lower[subscripts])
  list(xlim = xlim, ylim = ylim,
  dx = diff(xlim), dy = diff(ylim))
}
panel.gas <- function(x, y, subscripts, ...)
{
  panel.xyplot(x, y, cex = 0.5)
  ord <- order(x)
  lines(x[ord], gas.ci$fit[subscripts][ord])
  lines(x[ord], gas.ci$upper[subscripts][ord], lty=3)
  lines(x[ord], gas.ci$lower[subscripts][ord], lty=3)
  lines(x[ord], gas.ti$upper[subscripts][ord], lty=2)
  lines(x[ord], gas.ti$lower[subscripts][ord], lty=2)
}
xyplot(Gas ~ Temp | Insul, whiteside,
  prepanel = prepanel.gas, panel = panel.gas,
  xlab = "Average external temperature (deg. C)",
  ylab = "Gas consumption  (1000 cubic feet)")
```

The `prepanel` function is needed both to ensure that the tolerance bands fall inside the display and to allow the slopes of the fitted lines to be used in setting the aspect ratio.

6.16. We choose to use an eigendecomposition of W, as it is more stable than a Choleski factorization, and also makes it easier to use the same code for W or Σ. Let $W = UDU^T$. Then

$$(\boldsymbol{y}-X\boldsymbol{\beta})^T W(\boldsymbol{y}-X\boldsymbol{\beta}) = (\boldsymbol{y}-X\boldsymbol{\beta})^T UDU^T(\boldsymbol{y}-X\boldsymbol{\beta}) = \|D^{1/2}U^T(\boldsymbol{y}-X\boldsymbol{\beta})\|^2$$

so we can regress $A\boldsymbol{y}$ on AX where $A = D^{1/2}U^T$. If $W = \Sigma^{-1}$ we can take the eigendecomposition of Σ and replace D by D^{-1}. We modify `lm` as necessary. The following function[1] is in library `MASS`.

[1] The distributed version has extra code to handle safer prediction and `na.action` for prediction.

```
lm.gls <- function(formula, data, W, subset, na.action,
        inverse = F, method = "qr",
        model = F, x = F, y = F, contrasts = NULL, ...)
{
  call <- match.call()
  m <- match.call(expand = F)
  m$W <- m$inverse <- m$method <- m$model <- m$x <-
    m$y <- m$contrasts <- m$... <- NULL
  m[[1]] <- as.name("model.frame")
  m <- eval(m, sys.parent())
  if(method == "model.frame") return(m)
  Terms <- attr(m, "terms")
  Y <- model.extract(m, response)
  X <- model.matrix(Terms, m, contrasts)
  n <- nrow(X)
  if(any(dim(W) != c(n, n))) stop("dim(W) is not correct")
  eW <- eigen(W, T)
  d <- eW$values
  if(any(d <= 0)) stop("W is not positive definite")
  A <- diag(d^ifelse(inverse, -0.5, 0.5)) %*% t(eW$vector)
  fit <- lm.fit(A %*% X, A %*% Y, method, ...)
  fit$terms <- Terms
  fit$call <- call
  if(model) fit$model <- m
  if(x) fit$x <- X
  if(y) fit$y <- Y
  attr(fit, "na.message") <- attr(m, "na.message")
  class(fit) <- c("lm.gls", class(fit))
  fit
}
```

Our task is not over, since we need to be able to do something useful with the output. However, much of the `print` and `summary` methods for class `"lm"` are based on the stored results for the transformed problem and so are approximately correct. The fitted values and residuals are not simply related to the original problem.

We can test this with an example from Section 13.5. There we fitted a regression with autoregressive errors, and the covariance matrix for AR(1) errors is proportional to $(\alpha^{|i-j|})$.

```
alpha <- 0.8255; n <- 100
arow <- c(1, alpha^(1:n))
B <- matrix(c(rep(arow, n-1),1), n,n, byrow = T)
B[lower.tri(B)] <- 0
B <- B + t(B) - diag(n)
beav.gls <- lm.gls(temp ~ activ, W = B , inverse = T)
> summary(beav.gls)

Call: lm.gls(formula = temp ~ activ, W = B, inverse = T)
```

```
Coefficients:
                 Value Std. Error t value Pr(>|t|)
(Intercept)  37.166   0.091    408.776   0.000
      activ   0.669   0.098      6.809   0.000
```

This is reasonably consistent with the results of Section 15.6, where we noted appreciable end effects.

An important special case is for W a diagonal matrix. As a extension of the exercise modify `lm.gls` to allow the user to specify this case by supplying a vector of weights in `W` rather than a matrix. Note that `lm` can handle this by the use of (case) weights.

6.17. Recall what ridge regression does (Brown, 1993, Sen & Srivastava, 1990). Instead of fitting $X\beta$ to Y by least squares, it solves $[X^TX + \lambda I]\beta = X^TY$. (The case $\lambda = 0$ is the least-squares solution, but the ridge constant λ is positive in ridge regression.) Suppose X is an $n \times p$ matrix. Then the ridge regression problem is equivalent to the regression of Y' on X' where

$$X' = \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}, \qquad Y' = \begin{bmatrix} Y \\ 0 \end{bmatrix}$$

Thus we can implement ridge regression by adding p imaginary observations of 0, with $\sqrt{\lambda}$ as the value of the ith regressor and the others zero, for $i = 1, \dots, p$. Conventionally ridge regression is applied to the data with the mean removed and scaled so that the columns of X have constant length. (Any intercept term must then be removed.)

There is another approach that is more efficient if we need multiple values of λ, for example to plot a ridge trace or to choose λ by cross-validation. Let $X = U\Lambda V^T$ be the singular-value decomposition of X. Then $[X^TX + \lambda I]\beta = X^TY$ may be rewritten as $V[\Lambda^2 + \lambda]V^T\beta = V\Lambda U^TY$ and hence $V^T\hat{\beta}_\lambda = \Lambda/(\Lambda^2 + \lambda)U^TY = \Lambda^2/(\Lambda^2 + \lambda)V^T\beta_{LS}$. We implement this for a vector of values of λ, and compute some statistics to help choose λ, from Brown (1993, pp. 63–64). The following functions[2] are in library `MASS`.

```
lm.ridge <- function(formula, data, subset, na.action,
    lambda = 0, model = F, x = F, y = F, contrasts = NULL, ...)
{
  call <- match.call()
  m <- match.call(expand = F)
  m$model <- m$x <- m$y <- m$contrasts <-
    m$... <- m$lambda <- NULL
  m[[1]] <- as.name("model.frame")
  m <- eval(m, sys.parent())
  Terms <- attr(m, "terms")
  Y <- model.extract(m, response)
  X <- model.matrix(Terms, m, contrasts)
```

[2] The distributed version has extra code to handle safer prediction and `na.action` for prediction.

```
  n <- nrow(X); p <- ncol(X)
  if(Inter <- attr(Terms, "intercept"))
  {
    Xm <- apply(X[, -Inter], 2, mean)
    Ym <- mean(Y)
    p <- p - 1
    X <- X[, -Inter] - rep(Xm, rep.int(n, p))
    Y <- Y - Ym
  } else Ym <- Xm <- NA
  Xscale <- drop(rep(1/n, n) %*% X^2)^0.5
  X <- X/rep(Xscale, rep.int(n, p))
  Xs <- svd(X)
  rhs <- t(Xs$u) %*% Y
  d <- Xs$d
  lscoef <-  Xs$v %*% (rhs/d)
  lsfit <- X %*% lscoef
  resid <- Y - lsfit
  s2 <- sum(resid^2)/(n - p - Inter)
  HKB <- (p-2)*s2/sum(lscoef^2)
  LW <- (p-2)*s2*n/sum(lsfit^2)
  k <- length(lambda)
  div <- d^2 + rep(lambda, rep.int(p,k))
  a <- (d*rhs)/div
  dim(a) <- c(p, k)
  coef <- Xs$v %*% a
  dimnames(coef) <- list(names(Xscale), format(lambda))
  GCV <- apply((Y - X %*% coef)^2, 2, sum)/
         (n-apply(matrix(d^2/div,p), 2, sum))^2
  structure(list(coef = drop(coef), scales = Xscale,
     Inter = Inter, lambda = lambda, ym = Ym, xm = Xm,
     GCV = GCV, kHKB = HKB, kLW = LW), class="ridgelm")
}

print.ridgelm <- function(obj)
{
  scaledcoef <- t(as.matrix(obj$coef / obj$scales))
  if(obj$Inter) {
    inter <- obj$ym - scaledcoef %*% obj$xm
    scaledcoef<- cbind(Intercept=inter, scaledcoef)
  }
  print(drop(scaledcoef))
}

select <- function(obj) UseMethod("select")

select.ridgelm <- function(obj)
{
  cat("modified HKB estimator is", format(obj$kHKB), "\n")
  cat("modified L-W estimator is", format(obj$kLW), "\n")
  GCV <- obj$GCV
```

```
    if(length(GCV) > 0) {
      k <- seq(along=GCV)[GCV==min(GCV)]
      cat("smallest value of GCV  at",
          format(obj$lambda[k]), "\n")
    }
  }

  plot.ridgelm <- function(obj)
    matplot(obj$lambda, t(obj$coef), type = "l")
```

We can apply this to the celebrated Longley data, get a ridge trace and some estimates of λ.

```
longley <- data.frame(y = longley.y, longley.x)
lm.ridge(y ~ ., longley)
plot(lm.ridge(y ~ ., longley,
              lambda = seq(0,0.1,0.001)))
select(lm.ridge(y ~ ., longley,
              lambda = seq(0,0.1,0.0001)))
modified HKB estimator is 0.0042754
modified L-W estimator is 0.032295
smallest value of GCV  at 0.0028
```

There is only a little evidence for the necessity to use ridge regression here, but it can be seen as an alternative to variable selection.

Chapter 8

8.1. The code from the First Edition follows.

```
attach(wtloss)
plot(Days, Weight, xlab= "days", ylab ="weight (kg)",
   xlim=c(0,730), ylim=c(70, 200))
xx <- seq(0, 730, 10)
lines(xx, 81.37+ 102.68 * 2^(-xx/141.91))
wtloss.quad <- lm(Weight ~ poly(Days, 2))
lines(xx, predict.gam(wtloss.quad, data.frame(Days=xx)), lty=2)
wtloss.cub <- lm(Weight ~ poly(Days, 3))
lines(xx, predict.gam(wtloss.cub, data.frame(Days=xx)), lty=3)
legend(locator(1), c("exponential", "quadratic", "cubic"),
    lty=1:3)
```

Note the use of `predict.gam` to get valid predictions.

8.2.

```
expn2 <- deriv(~ b0 + b1*((w0 - b0)/b1)^(x/d0),
         c("b0","b1","d0"), function(b0, b1, d0, x, w0) {})
wtloss.init <- function(obj, w0) {
  p <- coef(obj)
  d0 <-  - log((w0 - p["b0"])/p["b1"], 2) * p["th"]
  c(p[c("b0", "b1")], d0 = as.vector(d0))
}
for(w0 in c(110, 100, 90)) {
    fm <- nls(Weight ~ expn2(b0, b1, d0, Days, w0),
              wtloss, start = wtloss.init(wtloss.gr, w0))
    print(plot(profile(fm)))
  }
```

8.3. [From the fourth printing of the First Edition.]

```
> attach(stormer,1)
> z1 <- Wt/Viscosity
> z2 <- -1/Viscosity
> detach(1,save="stormer")
> attach(stormer)
> storm.gm <- glm(Time ~ z1 + z2 - 1,
     family=quasi(link=inverse, variance=constant),
     data=stormer, trace=T, eps=1.0e-10)
GLM    linear loop 1: deviance = 860.92
GLM    linear loop 2: deviance = 825.06
GLM    linear loop 3: deviance = 825.05
GLM    linear loop 4: deviance = 825.05
> g <- coef(storm.gm)
> b <- coef(storm.fm)
> b0 <- c(1/g[1], g[2]/g[1])
> cbind(b,b0)
         b      b0
z1 29.4013 29.4013
z2  2.2182  2.2183
```

To find the standard errors we used

```
> J <- matrix(c(-1/g[1]^2, -g[2]/g[1]^2, 0, 1/g[1]), 2, 2)
> J %*% vcov(storm.gm) %*% t(J)
         [,1]     [,2]
[1,]  0.83820 -0.56055
[2,] -0.56055  0.44292
```

Note that to achieve agreement to this accuracy we had to tighten the convergence criteria for the `glm` fit by setting `eps=1.0e-10`. With the default convergence criteria there is agreement to about 3 significant digits.

8.4. [From the First Edition.]
We fit the model using the stable parametrization. Good initial values are always easy to find by estimating the mean at the required points by an approximating linear model.

```
> stab <- deriv3(~ ((u0*u2-u1^2) +
    (u0-u1)^2 *((u1-u2)/(u0-u1))^((x-40)/80))/(u0-2*u1+u2),
    c("u0","u1","u2"), function(x, u0, u1, u2) NULL)
> mu <- predict(lm(Weight ~ Days+Days^2, data=wtloss),
    newdata=data.frame(Days=c(40,120,200)))
> names(mu) <- paste("u", 0:2, sep="")
> wtloss.st <- nls(Weight ~ stab(Days, u0, u1, u2),
    start=mu, data=wtloss, trace=T)
43.3655 : 166.18 138.526 119.742
39.2447 : 165.834 138.515 120.033
> rms.curv(wtloss.st)
Parameter effects: c^theta x sqrt(F) = 0.0101
        Intrinsic: c^iota  x sqrt(F) = 0.0101

> summary(wtloss.st)$correlation
         u0      u1       u2
u0  1.00000 0.43675 -0.11960
u1  0.43675 1.00000  0.25806
u2 -0.11960 0.25806  1.00000
> plot(profile(wtloss.st))
```

8.9. Some of the children were tested at more than one age, so first we generate unique IDs for each experiment.

```
aa <- factor(OME$Age)
ab <- 10*OME$ID + unclass(aa)
ac <- unclass(factor(ab))
OME <- OME
OME$UID <- as.vector(ac)
OME$UIDn <- OME$UID + 0.1*(OME$Noise=="incoherent")
rm(aa, ab, ac)
```

Our first model is least-squares fitting to the success probabilities.

```
fp1 <- deriv(~ 0.5 + 0.5/(1 + exp(-(x-L75)/scal)),
            c("L75", "scal"),
            function(x,L75,scal) NULL)
```

The effective range of a logistic is about ±3 times `scal`, so by inspecting the data we can choose initial values of `L75` as 45 and `scal` as 3. It seems appropriate to analyse the two types of noise stimulus separately, at least initially.

```
> nls(Correct/Trials ~ fp1(Loud, L75, scal),
      data=OME[OME$Noise=="coherent",],
      start=c(L75=45, scal=3))
```

```
    L75   scal
 47.993 1.2594
> nls(Correct/Trials ~ fp1(Loud, L75, scal),
      data=OME[OME$Noise=="incoherent",],
      start=c(L75=45, scal=3))
    L75   scal
 38.866 2.1702
```

This suggests fixing on `scal = 2`, and fitting a separate `L75` for each experiment[3]. We used `nlsList`, and allow that a small proportion of fits will fail.

```
OMEi <- OME
fp2 <- deriv(~ 0.5 + 0.5/(1 + exp(-(x-L75)/2)),
            "L75", function(x,L75) NULL)
OMEi.nls <- nlsList(Correct/Trials ~ fp2(Loud, L75) | UIDn,
   data = OMEi, start=list(L75=45), control = list(maxiter=100))
tmp <- sapply(OMEi.nls, function(X)
              {if(is.null(X)) NA else as.vector(coef(X))})
OMEif <- data.frame(UID = round(as.numeric((names(tmp)))),
         Noise = rep(c("coherent", "incoherent"), 110),
         L75 = as.vector(tmp))
OMEif$Age <- OME$Age[match(OMEif$UID, OME$UID)]
OMEif$OME <- OME$OME[match(OMEif$UID, OME$UID)]
OMEif <- OMEif[OMEif$L75 > 30,]
```

This provides a data frame of the result of each experiment to which we can apply standard linear models. (The precise results will vary by platform, and it may be necessary to exclude 'silly' values such as -39 dB.) For example, we can consider if `L75` varies linearly with `Age` by

```
options(contrasts=c("contr.treatment", "contr.poly"))
summary(lm(L75 ~ Noise/Age, data=OMEif, na.action=na.omit))
```

and if the `OME` groups (only defined at ages 30 and 60 months) differ by

```
summary(lm(L75 ~ Noise/(Age + OME), data=OMEif,
           subset=Age >=30 & Age <= 60,
           na.action=na.omit, singular.ok=T), cor=F)
```

The analysis so far does not take the varying number of trials into account. We can do a weighted least-squares analysis by, for example

```
fpl75 <-
deriv(~ sqrt(n)*(r/n - 0.5 - 0.5/(1 + exp(-(x-L75)/scal))),
      c("L75", "scal"), function(r,n,x,L75,scal) NULL)
nls(0 ~ fpl75(Correct, Trials, Loud, L75, scal),
    data=OME[OME$Noise=="coherent",],
    start=c(L75=45, scal=3))
```

[3] In principle it would be better to fit a combined `nls` model with a separate `L75` for each level of `UIDn` and a common value of `scal`. This can be specified by `Correct/Trials 0̃.5 +0.5/(1 + exp(-(Loud - L75[UIDn])/scal))` but failed to converge.

```
    L75   scal
 47.798 1.2962
nls(0 ~ fpl75(Correct, Trials, Loud, L75, scal),
    data=OME[OME$Noise=="incoherent",],
    start=c(L75=45, scal=3))
    L75   scal
 38.553 2.0781

fpl75age <- deriv(~ sqrt(n)*(r/n -  0.5 - 0.5/
                     (1 + exp(-(x-L75-slope*age)/scal))),
   c("L75", "slope", "scal"),
   function(r,n,x,age,L75,slope,scal) NULL)
OME.nls1 <- nls(0 ~ fpl75age(Correct, Trials, Loud, Age,
                             L75, slope, scal),
    data=OME[OME$Noise=="coherent",],
    start=c(L75=45, slope=0, scal=2))
    L75     slope   scal
 48.682 -0.028716 1.2596
OME.nls1
sqrt(diag(vcov(OME.nls1)))
[1] 0.61093 0.01666 0.17565

OME.nls2 <-nls(0 ~ fpl75age(Correct, Trials, Loud, Age,
                            L75, slope, scal),
               data=OME[OME$Noise=="incoherent",],
               start=c(L75=45, slope=0, scal=2))
OME.nls2
   L75    slop    scal
 41.73 -0.10006 1.9796
sqrt(diag(vcov(OME.nls2)))
[1] 0.495592 0.013484 0.244558
```

and similarly for the individual fits. It would also be possible to extract standard errors for the individual `L75` estimates from the results of `nlsList`.

Non-linear mixed effects models

We have to use the expanded data frame `OMEf`, as it is not sensible to weight mixed models. We can generate most of the data frame using answer 2.5, by

```
OMEf <- OME[rep(1:nrow(OME), OME$Trials),]
```

To generate the `Resp` column is slightly trickier: we used

```
attach(OME)
OMEf$Resp <- unlist(lapply(1:length(Trials), function(i)
   c(rep(1, Correct[i]), rep(0, Trials[i] - Correct[i]))))
OMEf <- OMEf[, -match(c("Correct", "Trials"), names(OMEf))]
detach()
```

It is possible to fully vectorize this, for example by

```
OMEf$Resp <- rep(rep(c(1,0), length(Trials)),
                 t(cbind(Correct, Trials-Correct)))
```

where the matrix transpose is a ‘trick’ to interleave the two vectors. This approach is significantly faster (0.03 secs *versus* 2 secs on a Sun Ultra 1/170), but the thinking time was much longer.

We change the parametrization of `scal` to ensure it remains positive: we allow a random effect on log scale for this parameter. The following fits take a long time, and the precise answers vary by version of NLME and platform, and they may not converge at all.

```
fp2 <- deriv(~ 0.5 + 0.5/(1 + exp(-(x-L75)/exp(lsc))),
             c("L75", "lsc"),
             function(x, L75, lsc) NULL)
G1.nlme <- nlme(Resp ~ fp2(Loud, L75, lsc),
     fixed = list(L75 ~ Age, lsc ~ 1),
     random = L75 + lsc ~ 1 | UID,
     data=OMEf[OMEf$Noise=="coherent",], method="ML",
     start = list(fixed=c(L75=c(48, -0.03), lsc=0)), verbose=T)
summary(G1.nlme)
    ....
Random effects:
 Formula: list(L75 ~ 1, lsc ~ 1)
 Level: UID
 Structure: General positive-definite
                  StdDev   Corr
L75.(Intercept) 1.64019 L75.(I
            lsc 0.55130 -1
       Residual 0.40576

Fixed effects: list(L75 ~ Age, lsc ~ 1)
                 Value Std.Error   DF t-value p-value
L75.(Intercept)  48.121   0.66170 2141  72.723  <.0001
        L75.Age  -0.025   0.01945 2141  -1.303  0.1927
            lsc   0.267   0.15068 2141   1.773  0.0764

G2.nlme <- nlme(Resp ~ fp2(Loud, L75, lsc),
     fixed = list(L75 ~ Age, lsc ~ 1),
     random = L75 + lsc ~ 1 | UID,
      data=OMEf[OMEf$Noise=="incoherent",], method="ML",
     start = list(fixed=c(L75=c(41, -0.1), lsc=0)), verbose=T)
summary(G2.nlme)
    ....
Random effects:
 Formula: list(L75 ~ 1, lsc ~ 1)
 Level: UID
 Structure: General positive-definite
                    StdDev   Corr
L75.(Intercept) 2.168615106 L75.(I
            lsc 0.002344956 0.065
```

```
          Residual 0.317758632

Fixed effects: list(L75 ~ Age, lsc ~ 1)
                  Value Std.Error   DF t-value p-value
L75.(Intercept)  41.748     1.436 1831  29.068   0.000
        L75.Age  -0.111     0.064 1831  -1.736   0.083
            lsc  -9.377  1382.389 1831  -0.007   0.995
```

The results are remarkably similar to those by weighted least squares. In the case of `G2.nlme` this is not surprising as the estimates of the variances of the random effects are effectively zero. For `G1.nlme` the variances are reasonable but the estimate of the correlation is -1. For `G2.nlme` it seems clear that the scale is effectively not estimable.

8.10. Let us write the gamma density as

$$f(x; \lambda, \alpha) = \lambda^\alpha x^{\alpha-1} e^{-\lambda x}/\Gamma(\alpha) \qquad \text{on } [0, \infty)$$

Then the log-likelihood is

$$L(\lambda, \alpha) = \sum_i \Big[\alpha \log\lambda + (\alpha - 1)\log x_i - \lambda x_i - \log\Gamma(\alpha)\Big]$$

Reasonable initial estimates are given by the moment estimators $\mu = \alpha/\lambda, \sigma^2 = \alpha/\lambda^2$ so $\hat\lambda = \overline{x}/s^2, \hat\alpha = \overline{x}^2/s^2$. Thus a first approach might be

```
gamma.mle0 <- function(x)
{
  nloglik <- function(theta, x)
     - (theta[2] - 1)*sum(log(x)) + theta[1]*sum(x) -
     length(x) * (theta[2]*log(theta[1]) - lgamma(theta[2]))

  xbar <- mean(x)
  lambda0 <- xbar/var(x); alpha0 <- xbar*lambda0
  res <- nlminb(c(lambda0, alpha0), nloglik, lower=c(0,0), x=x)
  list(lambda = res$par[1], alpha = res$par[2],
       loglik = -res$objective)
}
```

Such a function has been posted to `S-news`, but it can be improved in a number of ways. The sufficient statistic $(\sum x_i, \sum \log x_i)$ is computed many times. The range for the parameters is not really $[0, \infty)$ but $(0, \infty)$, and we would do better to take $\theta = (e^\lambda, e^\alpha)$; at the very least we should give a lower limit at which `nloglik` can be evaluated. We could use gradient information in the calculation, but if we compute derivatives we find $\hat\lambda = \alpha/\overline{x}$ for given α, so we can reduce the problem to maximizing

$$L(\hat\lambda(\alpha), \alpha) = n\alpha \log \alpha/\overline{x} + (\alpha - 1)\sum \log x_i - n\alpha^2/\overline{x} - n\,\log\Gamma(\alpha)$$

We can easily find the derivative, but for one-dimensional optimization problems it is not particularly helpful, and `optimize` cannot make use of it.

```
gamma.mle1 <- function(x)
{
  nloglik <- function(alpha, n, xbar, st)
    -(n*alpha*log(alpha/xbar) + (alpha - 1)*st
      - n*alpha - n*lgamma(alpha))

  xbar <- mean(x); n <- length(x); st <- sum(log(x))
  alpha0 <- xbar^2/var(x)
  res <- optimize(nloglik, lower=alpha0/3, upper=alpha0*3,
                  n=n, xbar=xbar, st=st)
  alpha <- res$min
  list(lambda = alpha/xbar, alpha = alpha,
       loglik = -res$objective)
}
```

We minimize minus the log likelihood because `optimize` does not work correctly when maximizing in S-PLUS 3.x.

```
> set.seed(123)
> xg <- rgamma(500, 1.4)
> unix.time(gamma.mle0(xg))
[1] 5.34 0.20 6.00 0.00 0.00
> unix.time(gamma.mle1(xg))
[1] 0.75 0.09 1.00 0.00 0.00
> gamma.mle1(xg)
$lambda:
[1] 0.93358
$alpha:
[1] 1.3737
$loglik:
[1] -678.95
```

An alternative approach using a Newton algorithm is given in the function `gamma.shape.glm` in library `MASS`.

Chapter 13

13.1. From the First and Second Editions:

```
> dacc <- diff(accdeaths, 12)
> ts.plot(dacc)
> acf(dacc, 30)
> acf(dacc, 30, "partial")
> ddacc <- diff(dacc)
> ts.plot(ddacc)
> acf(ddacc, 30)
> acf(ddacc, 30, "partial")
> ddacc.1 <- arima.mle(ddacc-mean(ddacc),
     model=list(list(order=c(0,0,1)),
     list(order=c(0,0,1), period=12)))
$model[[1]]$ma:
[1] 0.48834
$model[[2]]$ma:
[1] 0.58534
$aic:
[1] 852.72
$loglik:
[1] 848.72
$sigma2:
[1] 94629
> sqrt(diag(ddacc.1$var.coef))
[1] 0.11361 0.10556
> ddacc.2 <- arima.mle(ddacc-mean(ddacc),
      model=list(order=c(0,0,13),
      ma.opt=c(T,F,F,F,F,T,F,F,F,F,F,T,T)),
      max.iter=50, max.fcal=100)
$model$ma:
 [1]  0.60784  0.00000  0.00000  0.00000  0.00000  0.41119
 [7]  0.00000  0.00000  0.00000  0.00000  0.00000  0.67693
[13] -0.47260
$aic:
[1] 869.85
$loglik:
[1] 843.85
$sigma2:
[1] 70540
> sqrt(diag(ddacc.2$var.coef))
 [1] 0.11473 0.10798 0.10798 0.10798 0.10798 0.10798 0.12052
 [8] 0.10798 0.10798 0.10798 0.10798 0.10798 0.11473
```

The plots (Figure 13.1) suggest the use of $\nabla\nabla_{12}X$, and this has a non-zero mean. The first model fitted is

$$\nabla\nabla_{12}X = 28.83 + (1 - 0.488B)(1 - 0.585B^{12})\epsilon$$

and the second model comes from selecting promising non-zero terms in a general MA(13) process, as

$$\nabla\nabla_{12}X = 28.83 + (1 - 0.608B - 0.411B^{6} - 0.677B^{12} + 0.473B^{13})\epsilon$$

Note that the AIC is wrong; it should be 851.85 as there are parameters set to zero (although this does not allow for selection). This fit illustrates the ability to constrain coefficients in an ARIMA fit. That standard errors are returned for zero parameters suggests that the standard errors are wrong. Standard likelihood theory suggests deleting rows from the inverse of the information matrix:

```
> dd.VI <- solve(ddacc.2$var.coef)
> sqrt(diag(
```

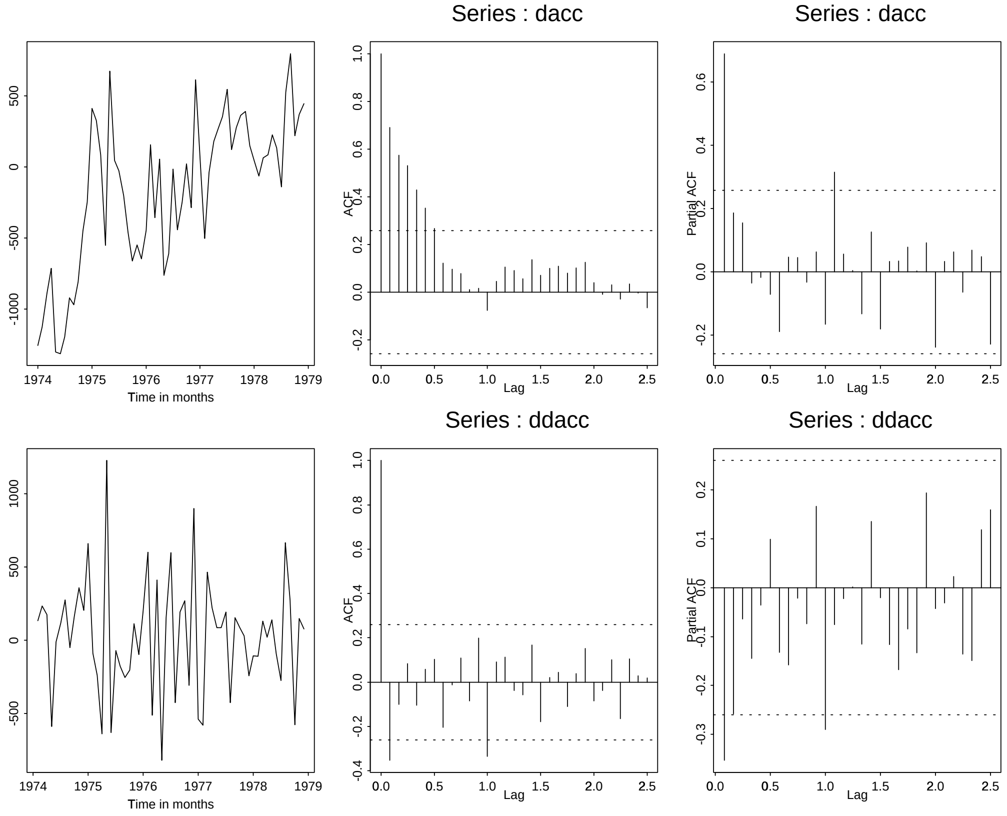


Figure 13.1: Seasonally differenced (top row) and then differenced (bottom row) versions of the accidental deaths series `accdeath` with ACF and PACF plots.

```
          solve(dd.VI[ddacc.2$model$ma.opt,ddacc.2$model$ma.opt])
          ))
     [1] 0.096691 0.085779 0.094782 0.095964
```

which shows the power of the S language.

13.2. We start by creating the quarterly percentage differences. Fitting by `ar` suggests that AR(4) and AR(6) models are almost equally good. Thus we try fractional differencing without an AR component and with AR(4) and AR(6) components. Unfortunately the likelihoods are not comparable between `arima.mle` and `arima.fracdiff`, and it seems the latter cannot be used with a specified degree d of (fractional) differencing. Since with the fractional ARIMA(6, d,0) model the estimate $\hat{d} \approx 0$ we can guess that AR(6) has AIC approximately $-2 \times 122.7 + 2 \times 6 = -233.4$ and the I(d) has an AIC of $-2 \times 116.47 + 2 = -231.1$. Thus we would choose the ARIMA(6,0,0) model. This differs from the conclusions of Brockwell & Davis (1996), but with such a short series end-effects may be important.

```
> y <- diff(austres)/austres * 100
> ar(y)
$order:
```

```
[1] 6

$ar:
          [,1]
[1,]  0.422690
[2,]  0.081845
[3,]  0.124695
[4,]  0.232673
[5,] -0.016759
[6,] -0.199008

$var.pred:
          [,1]
[1,] 0.0040826

$aic:
 [1] 30.89342  1.77094  2.48377  1.45398  0.53235  1.55605
 [7]  0.00000  1.94228  3.26542  5.07989  5.44845  6.32720
[13]  8.13424  6.68541  5.62113  7.61780  9.52565 11.23940
[19] 13.23219 14.39861

> arima.mle(y-mean(y), model=list(ar=rep(0,6)), n.cond=6)$aic
[1] -218.41
> arima.mle(y-mean(y), model=list(ar=rep(0,4)), n.cond=6)$aic
[1] -218.31
> arima.mle(y-mean(y), model=list(ar=0), n.cond=6)$aic
[1] -215.28

> arima.fracdiff(y-mean(y), model=list(d=0, ar=rep(0,6)))
$model:
$model$d:
[1] 4.583e-05

$model$ar:
[1]  0.3964324  0.1059762  0.1618604  0.2528531 -0.0094455
[6] -0.2029133
    ....
$loglik:
[1] 122.7

> arima.fracdiff(y-mean(y), model=list(d=0))
$model:
$model$d:
[1] 0.43245
    ....
$loglik:
[1] 116.47
```

13.4. We treat the AR(1) model as a simultaneous spatial autoregression (SAR) although it could also be handled as a conditional autoregression (CAR) process. Note that unlike `arima.mle` this computes an exact (not conditional) likelihood.

```
module(spatial)
beav2.nbr1 <- spatial.neighbor(2:100, 1:99, nregion=100)
slm(temp ~ activ, data = beav2, cov.family = SAR,
   spatial.arglist = list(neighbor=beav2.nbr1))
Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept)  36.586   0.126    290.197   0.000
      activ   0.476   0.126      3.774   0.000
rho =  0.99521
Residual standard error: 0.12621 on 97 degrees of freedom

slm(temp ~ activ, data = beav2, subset = 6:100,
   cov.family = SAR,
   spatial.arglist = list(neighbor=beav2.nbr1))

Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept)  37.252   0.082    453.049   0.000
      activ   0.596   0.094      6.322   0.000

rho =  0.849
Residual standard error: 0.12067 on 92 degrees of freedom
```

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data*. London: Chapman & Hall. [2, 20]

Aitkin, M., Anderson, D., Francis, B. and Hinde, J. (1989) *Statistical Modelling in GLIM*. Oxford: Oxford University Press. [9]

Baxter, L. A., Coutts, S. M. and Ross, G. A. F. (1980) Applications of linear models in motor insurance. In *Proceedings of the 21st International Congress of Actuaries, Zurich*, pp. 11–29. [9]

Brockwell, P. J. and Davis, R. A. (1991) *Time Series: Theory and Methods*. Second Edition. New York: Springer-Verlag. [13]

Brockwell, P. J. and Davis, R. A. (1996) *Introduction to Time Series and Forecasting*. New York: Springer-Verlag. [13, 36]

Brown, P. J. (1993) *Measurement, Regression and Calibration*. Oxford: Oxford University Press. [5, 25]

Cleveland, W. S. (1993) *Visualizing Data*. Summit, NJ: Hobart Press. [2]

Ehrlich, I. (1973) Participation in illegitimate activities: A theoretical and empirical investigation. *Journal of Political Economy* **81**, 521–565. [3]

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995) *Bayesian Data Analysis*. London: Chapman & Hall. [7]

Harvey, A. C. (1989) *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press. [13]

Harvey, A. C. and Durbin, J. (1986) The effects of seat belt legislation on British road casualties: A case study in structural time series modelling (with discussion). *Journal of the Royal Statistical Society series A* **149**, 187–227. [13]

Knight, R. L. and Skagen, S. K. (1988) Agnostic asymmetries and the foraging ecology of Bald Eagles. *Ecology* **69**, 1188–1194. [6]

Lock, R. H. (1993) New car data. *Journal of Statistics Education* **1**(1). See the URL http://jse.stat.ncsu.edu/jse/v1n1/datasets.lock. [4]

Mangasarian, O. L. and Wolberg, W. H. (1990) Cancer diagnosis via linear programming. *SIAM News* **23**, 1, 18. [13]

McLachlan, G. J. and Jones, P. N. (1988) Fitting mixture models to grouped and truncated data via the EM algorithm. *Biometrics* **44**, 571–578. [12]

McLachlan, G. J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: Wiley. [12]

Milicer, H. and Szczotka, F. (1966) Age at menarche in Warsaw girls in 1965. *Human Biology* **B38**, 199–203. [6]

Natrella, M. (1963) *Experimental Statistics*. Washington, DC: NBS Handbook 91. [2]

Raftery, A. E. (1995) Bayesian model selection in social research. In *Sociological Methodology 1995*, ed. P. V. Marsden, pp. 111–196. Oxford: Blackwells. [3]

Rawlings, J. O. (1988) *Applied Regression Analysis. A Research Tool*. Pacific Grove, California: Wadsworth and Brooks/Cole. [3]

Rice, J. A. (1995) *Mathematical Statistics and Data Analysis*. Second Edition. Belmont, CA: Duxbury Press. [2]

Ross, G. J. S. (1970) The efficient use of function minimization in non-linear maximum-likelihood estimation. *Applied Statistics* **19**, 205–221. [10]

Scheffé, H. (1959) *The Analysis of Variance*. New York: John Wiley and Sons. [4]

Sen, A. and Srivastava, M. (1990) *Regression Analysis*. New York: Springer-Verlag. [5, 25]

Svensson, Å. (1981) On the goodness-of-fit test for the multiplicative Poisson model. *Annals of Statistics* **9**, 697–704. [8]

Vandaele, W. (1978) Participation in illegitimate activities: Ehrlich revisited. In *Deterrence and Incapacitation*, eds A. Blumstein, J. Cohen and D. Nagin, pp. 270–335. Washington: US National Academy of Sciences. [3]

Wand, M. P. (1997) Data-based choice of histogram bin width. *American Statistician* **52**, 59–64. [3]