Chapter 7
Regression

7.1 Introduction

Regression is a general statistical method to fit a straight line or other model to data. The objective is to find a model for predicting the dependent variable (*response*) given one or more independent (*predictor*) variables.

The simplest example is a *simple linear regression model* of Y on X, defined by

$$Y = \beta_0 + \beta_1 X + \varepsilon, \tag{7.1}$$

where ε is a random error term. The word "simple" means that there is one predictor variable in the model. The linear model (7.1) describes a straight line relation between the response variable Y and predictor X.

In least squares regression the unknown parameters β_0 and β_1 are estimated by minimizing the sum of the squared deviations between the observed response Y and the value $\hat{Y}$ predicted by the model. If these estimates are b_0 (intercept) and b_1 (slope), the estimated regression line is

$$\hat{Y} = b_0 + b_1 X.$$

For a set of data (x_i, y_i), $i = 1, \ldots, n$, the errors in this estimate are $y_i - \hat{y}_i$, $i = 1, \ldots, n$. Least squares regression obtains the estimated intercept b_0 and slope b_1 that minimizes the sum of squared errors: $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.

A *multiple linear regression model* has more than one predictor variable (multiple predictors). Linear models can describe many relations other than straight lines or planes. Any model that is linear in the parameters is considered a linear model. Thus a quadratic relation $y = \beta_0 + \beta_1 x + \beta_2 x^2$ corresponds to a linear model with two predictors, $X_1 = X$ and $X_2 = X^2$. The exponential relation $y = \beta_0 e^{\beta_1 x}$ is not linear, but the relation can be expressed by taking the natural logarithm of both sides. The corresponding linear equation is $\ln y = \ln \beta_0 + \beta_1 x$.

J. Albert and M. Rizzo, *R by Example*, Use R, DOI 10.1007/978-1-4614-1365-3_7,

7.2 Simple Linear Regression

7.2.1 Fitting the model

Example 7.1 (cars). Consider a set of paired observations of speed and stopping distance of cars. Is there a linear relation between stopping distance and speed of a car?

The data set `cars` is one of the data sets installed with R. We `attach` the data set `cars`, and from the help page for `cars` (displayed using `?cars`), we learn that there are 50 observations of `speed` (mph) and `dist` (stopping distance in feet), and that this data was recorded in 1920. Our first step in the analysis is to construct a scatterplot of `dist` vs `speed`, using the `plot` function.

```
> attach(cars)     #attach the data
> ?cars            #display the help page for cars data
> plot(cars)       #construct scatterplot
```

The scatterplot displayed in Figure 7.1 reveals that there is a positive association between distance `dist` and `speed` of cars. The relation between distance and speed could be approximated by a line or perhaps by a parabola. We start with the simplest model, the straight line model. The response variable in this example is stopping distance `dist` and the predictor variable speed is `speed`. To fit a straight line model

$$dist = \beta_0 + \beta_1\, speed + \varepsilon,$$

we need estimates of the intercept β_0 and the slope β_1 of the line.

The `lm` function and the model formula

The linear model function is `lm`. This function estimates the parameters of a linear model by the least squares method. A linear model is specified in R by a model `formula`.

The R `formula` that specifies a simple linear regression model $dist = \beta_0 + \beta_1 speed + \varepsilon$ is simply

$$\texttt{dist} \sim \texttt{speed}$$

The model formula is the first argument to the `lm` (linear model) function. In this example, the estimated regression model is obtained by

```
> lm(dist ~ speed)
```

The `lm` command above produces the following output.

```
Call:
lm(formula = dist ~ speed)
```

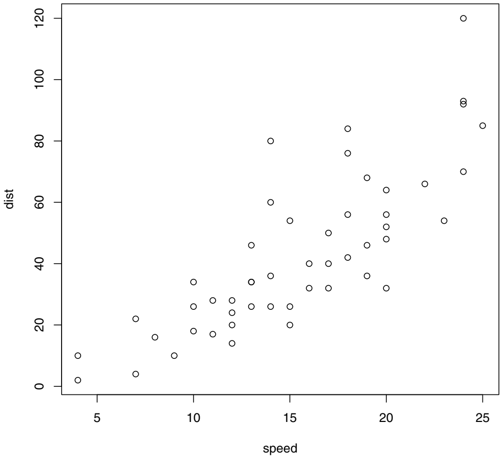


Fig. 7.1 Scatterplot of stopping distance vs speed in Example 7.1.

```
Coefficients:
(Intercept)        speed
    -17.579        3.932
```

The function `lm` displays only the estimated coefficients, but the object returned by `lm` contains much more information, which we will explore below. As we want to analyze the fit of this model, it is useful to store the result:

```
> L1 = lm(dist ~ speed)
> print(L1)
```

In this case the value of `L1` will be displayed after we type the symbol `L1` or `print(L1)`. The result will be as shown above.

The fitted regression line is: dist $= -17.579 + 3.932$ speed. According to this model, average stopping distance increases by 3.932 feet for each additional mile per hour of speed. The `abline` or `curve` functions can be used to add the fitted line to the scatterplot. See Figure 7.2(a).

```
> plot(cars, main="dist = -17.579 + 3.932 speed", xlim=c(0, 25))
> #line with intercept=-17.579, slope=3.932
> abline(-17.579, 3.932)
> curve(-17.579 + 3.932*x, add=TRUE)  #same thing
```

℞ **7.1** *A shortcut to add a simple linear regression line to a plot is to supply the result of* `lm` *as the first argument to* `abline`*. In the above example, we could have used* `abline(lm(dist ~ speed))` *or* `abline(L1)` *to add the fitted line to the plot in Figure 7.2(a).*

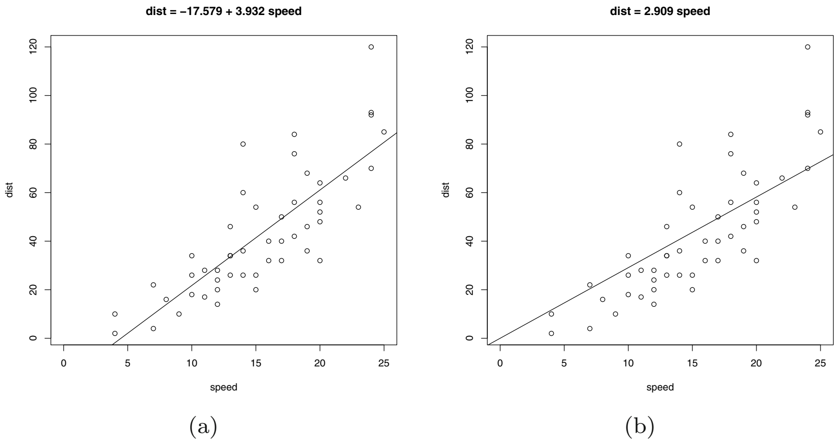


Fig. 7.2 Regression lines for `speed` vs `dist` in Examples 7.1–7.2. Figure (a) displays the data with the fitted line $\hat{y} = -17.579 + 3.932x$ from Example 7.1. Figure (b) displays the same data with the regression-through-the-origin fit, $\hat{y} = 2.909x$ from Example 7.2.

7.2.2 Residuals

The *residuals* are the vertical distances from the observed stopping distance `dist` (the plotting symbol) to the line. The paired observations are $(x_i, y_i) = (\texttt{speed}_i, \texttt{dist}_i)$, $i = 1, \ldots, 50$. The residuals are

$$e_i = y_i - \hat{y}_i,$$

where $\hat{y}_i$ denotes the value of stopping distance predicted by the model at speed x_i. In this problem, $\hat{y}_i = -17.579 + 3.932x_i$. One can observe in Figure 7.2(a) that the model tends to fit better at slow speeds than at higher speeds. It is somewhat easier to analyze the distribution of the errors using a residual plot. Figure 7.3 is a scatterplot of residuals vs fitted values. One way to generate the residual plot is to use the plot method for the `lm` result. The `which=1` argument specifies the type of plot (residuals vs fitted values). The `add.smooth` argument controls whether a certain type of curve is fitted to the residuals.

```
> plot(L1, which=1, add.smooth=FALSE)
```

The residual plot (Figure 7.3) has three unusually large residuals labeled; observations 23, 35, and 49. One can also observe that the residuals are closer to zero at slow speeds; the variance of the residuals is not constant across all speeds, but increasing with speed. Inference (tests or confidence intervals) about the model is usually based on the assumption that the errors are normally distributed with mean zero and constant variance.

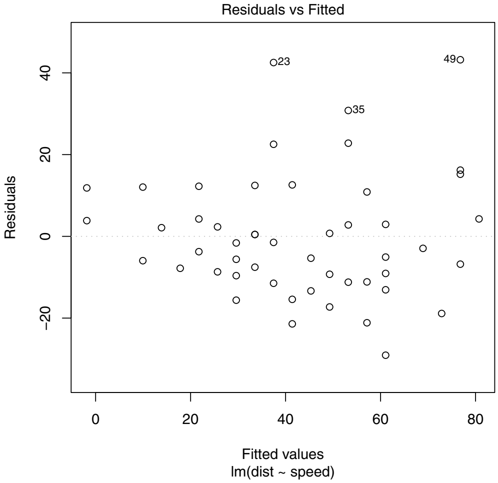


Fig. 7.3 Scatterplot of residuals vs fitted values for the `cars` data in Example 7.1.

7.2.3 Regression through the origin

Example 7.2 (cars, cont.). The `cars` data includes speeds as slow as 4 mph, and the estimated intercept should correspond to the expected distance required to stop a car that is not moving; however, our estimated intercept is -17.579 feet. The model with intercept zero

$$Y = \beta_1 X + \varepsilon$$

can be estimated by explicitly including the intercept 0 in the model formula. Then `lm(dist ~ 0 + speed)` sets intercept equal to zero and estimates the slope by the least squares method.

```
> L2 = lm(dist ~ 0 + speed)
> L2
Call:
lm(formula = dist ~ 0 + speed)

Coefficients:
speed
2.909
```

The estimated slope for this model is 2.909. For each additional one mph of speed, the estimated average stopping distance increases 2.909 feet. The fitted line plot for this model is in Figure 7.2(b). It is generated by the following code.

```
> plot(cars, main="dist = 2.909 speed", xlim=c(0,25))
> #line with intercept=0, slope=2.909
> abline(0, 2.909)
```

Again we observe that the fit is better at slow speeds than at faster speeds. A plot of residuals vs fitted values for this model can be generated by

```
> plot(L2, which=1, add.smooth=FALSE)
```

as described above. This plot (not shown) looks very similar to Figure 7.3.

A quadratic model could be considered for this data; see Exercise 7.8.

The `cars` data can be detached when it is no longer needed, using

```
> detach(cars)
```

7.3 Regression Analysis for Data with Two Predictors

In the next example there are two predictor variables. One could fit a simple linear regression model with either of these variables, or fit a multiple linear regression model using both predictors.

7.3.1 Preliminary analysis

Example 7.3 (Volume of Black Cherry Trees). The data file "cherry.txt" can be obtained from *StatSci* at `http://www.statsci.org/data/general/cherry.html` and can also be found in Hand et al. [21]. The data were collected from 31 black cherry trees in the Allegheny National Forest, Pennsylvania, in order to find an estimate for the volume of a tree (and therefore the timber yield), given its height and diameter. The data set contains a sample of 31 observations of the variables

Variable	*Description*
Diam	diameter in inches
Height	height in feet
Volume	cubic feet

This data set is also available in R as `trees`. It is identical to "cherry.txt" except that the diameter variable is named "Girth". We use the R data and rename the diameter as `Diam`, creating a new data frame called `Trees`, then `attach` the data frame.

```
> Trees = trees
> names(Trees)[1] = "Diam"
> attach(Trees)
```

The pairs function generates an array of scatterplots for each pair of variables. This type of plot (Figure 7.4) helps visualize the relations between variables.

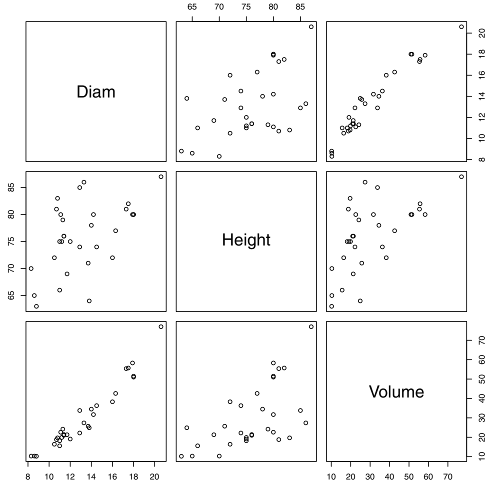


Fig. 7.4 Pairs plot of cherry tree data in Example 7.3.

In the pairs plot (Figure 7.4) the variables Diam and Volume appear to have a strong linear association, and Height and Volume are also related.

We also print a correlation matrix.

```
> pairs(Trees)
> cor(Trees)
```

```
            Diam    Height    Volume
Diam   1.0000000 0.5192801 0.9671194
Height 0.5192801 1.0000000 0.5982497
Volume 0.9671194 0.5982497 1.0000000
```

The correlation between diameter and volume is 0.97, indicating a strong positive linear relation between Diam and Volume. The correlation between height and volume is 0.60, which indicates a moderately strong positive linear association between Height and Volume.

As a first step, let us fit a simple linear regression model with diameter as the predictor:

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon,$$

where Y is the volume, X_1 is the diameter, and ε is random error; call this Model 1. The `lm` function is used to fit the model and we store the result in `M1`. The intercept term is included in the formula by default.

```
> M1 = lm(Volume ~ Diam)
> print(M1)

Call:
lm(formula = Volume ~ Diam, data = Trees)

Coefficients:
(Intercept)         Diam
    -36.943        5.066
```

The estimated intercept is -36.943 and the estimated slope is 5.066. According to this model, the average volume increases by 5.066 cubic feet for each additional 1 inch in diameter.

The fitted model `M1` contains the estimated coefficients. We add the line to the scatterplot of the data using the vector of coefficients `M1$coef`. The scatterplot with fitted line is shown in Figure 7.5

```
> plot(Diam, Volume)     #response vs predictor
> abline(M1$coef)        #add fitted line
```

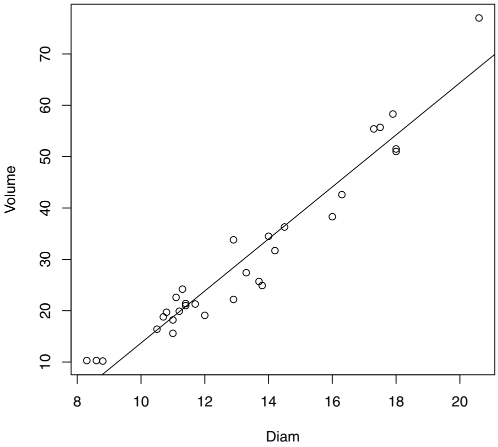


Fig. 7.5 Scatterplot of cherry tree volume vs tree diameter with fitted regression line in Example 7.3.

To predict volume for new trees, the `predict` method can be used. Store the diameter value for the new tree(s) in a data frame using the name `Diam` of the original model formula specified in the `lm` call. For example, the predicted volume for a new tree with diameter 16 in. is obtained by

```
> new = data.frame(Diam=16)
> predict(M1, new)

       1
44.11024
```

The predicted volume of the new tree is 44.1 cubic feet.

For inference, one requires some assumptions about the distribution of the error term. We assume that the random errors ε are independent and identically distributed (iid) as Normal$(0, \sigma^2)$ random variables. Residual plots help us to assess the fit of the model and the assumptions for ε.

One can obtain residual plots using the `plot` method for `lm`; here we are requesting two plots: a plot of residuals vs fits (1) and a QQ plot to check for normality of residuals (2).

```
  plot(M1, which=1:2)
```

The user is prompted for each graph with a message at the console:

```
Waiting to confirm page change...
```

The residual plots are shown in Figure 7.6(a) and 7.6(b). In Figure 7.6(a) a curve has been added. This curve is a fitted `lowess` (local polynomial regression) curve, called a *smoother*. The residuals are assumed iid, but there is a pattern evident. The residuals have a "U" shape or bowl shape. This pattern could indicate that there is a variable missing from the model. In the QQ plot 7.6(b), normally distributed residuals should lie approximately along the reference line shown in the plot. The observation with the largest residual corresponds to the tree with the largest volume, observation 31. It also has the largest height and diameter.

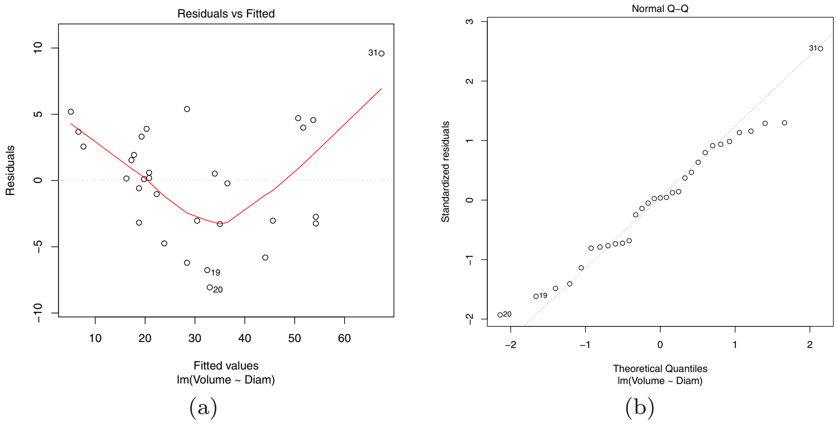
7.3.2 Multiple regression model

A multiple linear regression model with response variable Y and two predictor variables X_1 and X_2 is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon,$$

where ε is a random error term. For inference we assume that the errors are normally distributed and independent with mean zero and common variance σ^2.

Example 7.4 (Model for Volume of Cherry Trees, cont.). Next we consider the two variable model for predicting volume of cherry trees given diameter and height. This is a multiple regression model with two predictors; X_1 is the diameter and X_2 is the height of the tree. The response variable is Y, the volume. Call this Model 2.



(a) (b)

Fig. 7.6 Residuals vs fits (a) and normal QQ plot of residuals (b) in Example 7.3 for Model 1.

Least squares estimates of the parameters of multiple linear regression models are obtained by `lm`, using similar syntax as for simple linear regression. The model formula determines which type of model is fit. The model formula we require is

$$\texttt{Volume} \sim \texttt{Diam} + \texttt{Height}$$

and we fit the model, store it as `M2`, then print the result with the commands

```
> M2 = lm(Volume ~ Diam + Height)
> print(M2)

Call:
lm(formula = Volume ~ Diam + Height)

Coefficients:
(Intercept)         Diam       Height
   -57.9877       4.7082       0.3393
```

The fitted regression model is

$$\hat{Y} = -57.9877 + 4.7082X_1 + 0.3393X_2$$

or $\text{Volume} = -57.9877 + 4.7082\,\text{Diam} + 0.3393\,\text{Height} + \text{error}$. According to this model, when height is held constant, average volume of a tree increases by 4.7082 cubic feet for each additional inch in diameter. When diameter is held constant, average volume of a tree increases by 0.3393 cubic feet for each additional inch of height.

The residual plots for Model 2 are obtained by

```
> plot(M2, which=1:2)
```

(see the previous section). These residual plots for Model 2 (not shown) look similar to the corresponding plots for M1 in Figure 7.6(a). The "U" shaped pattern of residuals in the plot of residuals vs fits (similar to Figure 7.6(a)) may indicate that a quadratic term is missing from the model.

Example 7.5 (Model for Cherry Trees, cont.). Finally, let us fit a model that also includes the square of diameter as a predictor. Call this Model 3. The model is specified by the formula

```
    Volume ~ Diam + I(Diam^2) + Height
```

where I(Diam^2) means to interpret Diam^2 "as is" (the square of Diam) rather than interpret the exponent as a formula operator. We fit the model, storing the result in M3.

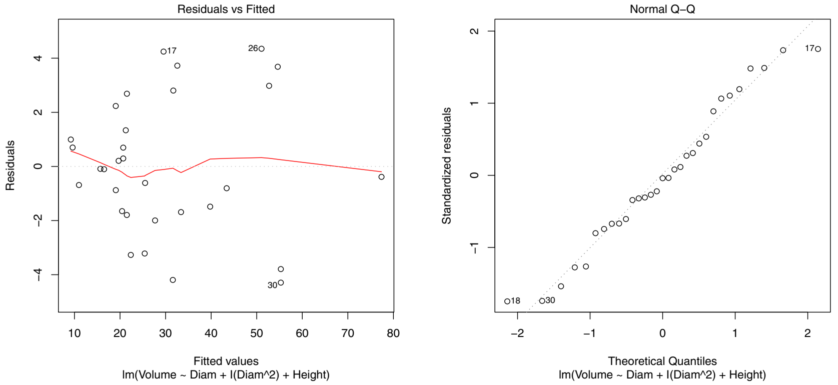
```
> M3 = lm(Volume ~ Diam + I(Diam^2) + Height)
> print(M3)

Call:
lm(formula = Volume ~ Diam + I(Diam^2) + Height)

Coefficients:
(Intercept)         Diam    I(Diam^2)       Height
    -9.9204      -2.8851       0.2686       0.3764
```

Then we display the residual plots, which are shown in Figures 7.7(a) and 7.7(b).

```
  plot(M3, which=1:2)
```



(a) (b)

Fig. 7.7 Residuals vs fits (a) and normal QQ plot of residuals in Example 7.5 for Model 3.

For Model 3, the plot of residuals vs fits in Figure 7.7(a) does not have the "U" shape that was apparent for Models 1 and 2. The residuals are approximately centered at 0 with constant variance. In the normal QQ plot (Figure 7.7(b)), the residuals are close to the reference line on the plot. These residual plots are consistent with the assumption that errors are iid with a Normal$(0, \sigma^2)$ distribution.

7.3.3 The `summary` *and* `anova` *methods for* `lm`

The `summary` of the fitted model contains additional information about the model. In the result of `summary` we find a table of the coefficients with standard errors, a five number summary of residuals, the coefficient of determination (R^2), and the residual standard error.

Example 7.6 (Cherry Trees Model 3). The `summary` of our multiple regression fit stored in `M3` is obtained below.

```
> summary(M3)

Call:
lm(formula = Volume ~ Diam + I(Diam^2) + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2928 -1.6693 -0.1018  1.7851  4.3489

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.92041   10.07911  -0.984 0.333729
Diam        -2.88508    1.30985  -2.203 0.036343 *
I(Diam^2)    0.26862    0.04590   5.852 3.13e-06 ***
Height       0.37639    0.08823   4.266 0.000218 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.625 on 27 degrees of freedom
Multiple R-squared: 0.9771,     Adjusted R-squared: 0.9745
F-statistic: 383.2 on 3 and 27 DF,  p-value: < 2.2e-16
```

The adjusted R^2 value of 0.9745 indicates that more than 97% of the total variation in Volume about its mean is explained by the linear association with the predictors `Diam`, `Diam`^{2}, and `Height`. The residual standard error is 2.625. This is the estimate of σ, the standard deviation of the error term ε in Model 3.

The table of coefficients includes standard errors and t statistics for testing $H_0 : \beta_j = 0$ vs $H_1 : \beta_j \neq 0$. The p-values of the test statistics are given under `Pr(>|t|)`. We reject the null hypothesis $H_0 : \beta_j = 0$ if the corresponding p-

value is less than the significance level. At significance level 0.05 we conclude that `Diam`, `Diam`^{2}, and `Height` are significant.

The analysis of variance (ANOVA) table for this model is obtained by the `anova` function.

```
> anova(M3)

Analysis of Variance Table

Response: Volume
          Df Sum Sq Mean Sq  F value    Pr(>F)
Diam       1 7581.8  7581.8 1100.511 < 2.2e-16 ***
I(Diam^2)  1  212.9   212.9   30.906 6.807e-06 ***
Height     1  125.4   125.4   18.198 0.0002183 ***
Residuals 27  186.0     6.9
```

From the ANOVA table, one can observe that `Diam` explains most of the total variability in the response, but the other predictors are also significant in Model 3.

A way to compare the models (Model 1 in Example 7.3, Model 2 in Example 7.4, and Model 3 in Example 7.5) is to list all of the corresponding `lm` objects as arguments to `anova`,

```
> anova(M1, M2, M3)
```

which produces the following table:

```
Analysis of Variance Table

Model 1: Volume ~ Diam
Model 2: Volume ~ Diam + Height
Model 3: Volume ~ Diam + I(Diam^2) + Height
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     29 524.30
2     28 421.92  1    102.38 14.861 0.0006487 ***
3     27 186.01  1    235.91 34.243  3.13e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This table shows that the residual sum of squares decreased by 102.38 from 524.30 when `Height` was added to the model, and decreased another 235.91 from 421.92 when the square of diameter was added to the model.

7.3.4 Interval estimates for new observations

Regression models are models for predicting a response variable given one or more predictor variables. We have seen how to obtain predictions (point estimates) of the response variable using `predict` for `lm` in Example 7.3 (see the `predict.lm` help topic). The `predict` method for `lm` also provides two types of interval estimates for the response:

a. Prediction intervals for new observations, for given values of the predictor variables.
b. Confidence intervals for the expected value of the response for given values of the predictors.

Example 7.7 (Cherry Trees Model 3, cont.).

To predict volume for new trees with given diameter and height, the `predict` method can be used. Store the diameter and height values for the new tree(s) in a data frame using the identical names as in the original model formula specified in the `lm` call. For example, to apply the Model 3 fit to obtain a point estimate for the volume of a new tree with diameter 16 in. and height 70 ft., we enter

```
> new = data.frame(Diam=16, Height=70)
> predict(M3, newdata=new)

       1
39.03278
```

The predicted volume of the new tree is 39.0 cubic feet. This estimate is about 10% lower than the prediction we obtained from Model 1, which used only diameter as a predictor.

To obtain a prediction interval or a confidence interval for the volume of the new tree, the `predict` method is used with an argument called `interval` specified. The confidence level is specified by the `level` argument, which is set at 0.95 by default. One can abbreviate the argument values. For a prediction interval specify `interval="pred"` and for a confidence interval use `interval="conf"`.

```
> predict(M3, newdata=new, interval="pred")

       fit      lwr      upr
1 39.03278 33.22013 44.84544
```

The prediction interval for volume of a randomly selected new tree of diameter 16 and height 70 is (33.2, 44.8) cubic feet. The confidence interval for the expected volume of all trees of diameter 16 and height 70 is obtained by

```
> predict(M3, newdata=new, interval="conf")

       fit      lwr      upr
1 39.03278 36.84581 41.21975
```

so the confidence interval estimate for expected volume is (36.8, 41.2) cubic feet. The prediction interval is wider than the confidence interval because the prediction for a single new tree must take into account the variation about the mean and also the variation among all trees of this diameter and height.

To obtain point estimates or interval estimates for several new trees, one would store the new values in a data frame like our data frame `new`. For example, if we require confidence intervals for diameter 16, at a sequence of values of height 65 to 70, we can do the following.

```
> diameter = 16
> height = seq(65, 70, 1)
> new = data.frame(Diam=diameter, Height=height)
> predict(M3, newdata=new, interval="conf")
```

which produces the following estimates:

```
       fit      lwr      upr
1 37.15085 34.21855 40.08315
2 37.52724 34.75160 40.30287
3 37.90362 35.28150 40.52574
4 38.28001 35.80768 40.75234
5 38.65640 36.32942 40.98338
6 39.03278 36.84581 41.21975
```

7.4 Fitting a Regression Curve

In this section we discuss two examples for which we want to estimate a regression curve rather than a straight line relation between a response variable Y and a single predictor X. In Example 7.8 the response variable is linearly related to the reciprocal of the predictor. In Example 7.9, we fit an exponential model.

Example 7.8 (Massachusetts Lunatics Data).

In Chapter 1, the *Massachusetts Lunatics* data[1] was introduced. These data are from an 1854 survey conducted by the Massachusetts Commission on Lunacy. We created the data file "lunatics.txt" from the table on the web. See Chapter 1 (Example 1.12, page 29) for a detailed explanation of how to import the data into R. We import the data into a data frame `lunatics` and `attach` it using

```
> lunatics = read.table("lunatics.txt", header=TRUE)
> attach(lunatics)
```

The data frame `lunatics` has 14 rows and six columns, corresponding to the following variables:

Variable	*Description*
COUNTY	Name of county
NBR	Number of lunatics, by county
DIST	Distance to nearest mental health center
POP	County population , 1950 (thousands)
PDEN	County population density per square mile
PHOME	Percent of lunatics cared for at home

[1] Data and Story Library, `http://lib.stat.cmu.edu/DASL/Datafiles/lunaticsdat.html`

In this example we investigate the relationship between the percentage of patients cared for at home and distance to the nearest health center.

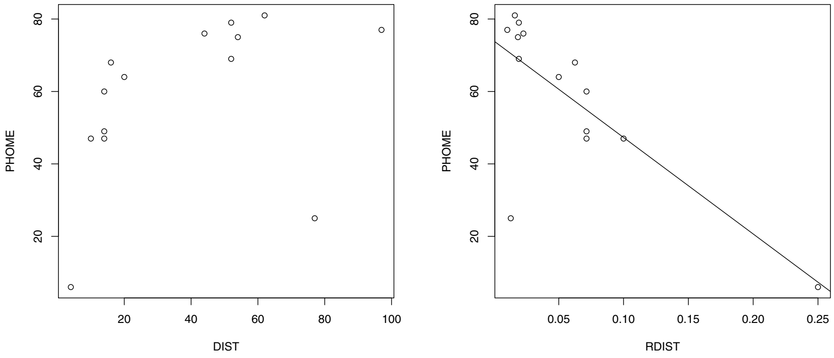
First we plot PHOME vs DIST to see if a linear relation is a plausible model, and print the sample correlation.

```
> plot(DIST, PHOME)
> cor(DIST, PHOME)
[1] 0.4124404
```

The sample correlation 0.41 measures the linear association between the two variables. The scatterplot in Figure 7.8(a) suggests that the relation between PHOME and DIST is nonlinear, perhaps more like a hyperbola. With this in mind, we create the variable RDIST, the reciprocal of distance, compute sample correlation, and plot PHOME vs RDIST.

```
> RDIST = 1/DIST
> plot(RDIST, PHOME)
> cor(RDIST, PHOME)
[1] -0.7577307
```

Here |cor(RDIST,PHOME)| > |cor(DIST,PHOME)|, indicating a stronger linear association between RDIST and PHOME than between the original variables DIST and PHOME. In Figure 7.8(b) a linear relation between PHOME and RDIST appears to be a plausible model. (The line on the plot is added below after fitting the model.)



(a) (b)

Fig. 7.8 Percent of lunatics cared for at home vs distance (a) and reciprocal of distance (b) in Example 7.8.

We fit the simple linear regression model

$$\text{PHOME}_i = \beta_0 + \beta_1 \text{RDIST}_i + \varepsilon_i, \qquad i = 1, \ldots, 14,$$

using the lm function. Typically we want to save the result in an object for further analysis.

```
> M = lm(PHOME ~ RDIST)
> M

Call:
lm(formula = PHOME ~ RDIST)

Coefficients:
(Intercept)        RDIST
      73.93      -266.32
```

The estimated regression line is $\text{PHOME} = 73.93 - 266.32\,\text{RDIST}$, and it can be added to the plot in Figure 7.8(b) with the abline function:

```
> abline(M)
```

Although data points for 13 of the 14 counties are close to the fitted line in Figure 7.8(b), there is one observation that is far from the line. We may also want to plot the fits for the original data. The fits are points on the curve

$$\text{PHOME} = \hat{\beta}_0 + \hat{\beta}_1 \frac{1}{\text{DIST}},$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the intercept and slope estimates stored in the $coef vector of our lm object. The plot in Figure 7.9 is obtained by

```
> plot(DIST, PHOME)
> curve(M$coef[1] + M$coef[2] / x, add=TRUE)
```

Again, we observe one observation that is far from the fitted curve. One also can observe that most of the observed data are above the fitted curve; the fitted model tends to underestimate the response.

A plot of residuals vs fits is produced by the command

```
> plot(M$fitted, M$resid, xlab="fitted", ylab="residuals")
```

We added a dashed horizontal line through 0 to the plot by

```
> abline(h=0, lty=2)
```

The plot is shown in Figure 7.10(a). On the residual plot we find that there is an outlier among the residuals at the lower right corner.

The identify function is helpful to identify which observation is the outlier. This function waits for the user to identify n points on the plot, and optionally labels the points. We want n=1 to identify one point, and we specify an abbreviation for the COUNTY as the label.

```
> lab = abbreviate(COUNTY)
> identify(M$fitted.values, M$residuals, n=1, labels=lab)
 [1] 13
```

The identify function returns the row number of the observation(s) identified on the plot. Row number 13 corresponds to NANTUCKET county, which

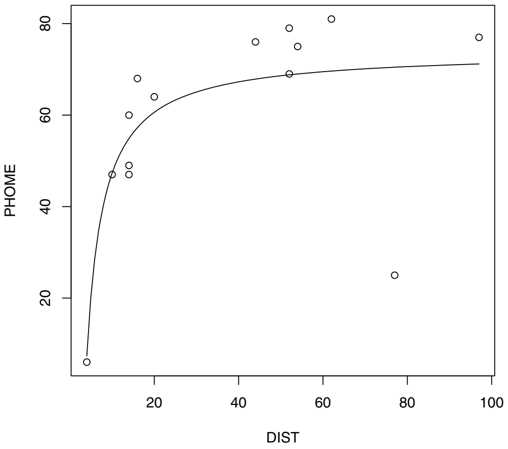
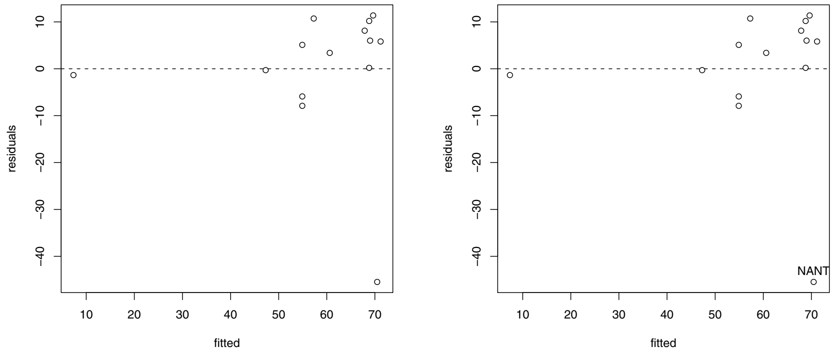


Fig. 7.9 Predicted values of percentage of patients cared for at home in Example 7.8.



(a) (b)

Fig. 7.10 Fitted values vs residuals for the regression of `PHOME` on `RDIST` in Example 7.8. In (b), the outlier has been labeled as “NANT” (NANTUCKET county), after using the `identify` function.

is labeled on the plot as “NANT” where we clicked (See Figure 7.10(b).) We can extract this observation from the data set by

```
> lunatics[13, ]

      COUNTY NBR DIST  POP PDEN PHOME
13 NANTUCKET  12   77 1.74  179    25
```

According to the documentation provided with the data set on the DASL web site, Nantucket county is an offshore island, which may need to be taken into account in the model.

Finally, when the data frame is no longer required, it can be detached.

```
> detach(lunatics)
```

Example 7.9 (Moore's Law). In a recent interview,[2] Google's CEO Eric Schmidt discussed the future of the internet. According to Moore's Law, Schmidt said, "in 10 years every computer device you use will be 100 times cheaper or 100 times faster." Moore's Law, named for Intel co-founder Gordon Moore [34], states that the number of transistors on a chip (a measure of computing power) doubles every 24 months. In 1965 Moore predicted that the number of transistors would double every year, but in 1975 he modified that doubling time to every two years.

Moore's Law has been applied to various measurements of computing power. If the rate of growth in computing power is assumed constant, Moore's Law is the model

$$y = b_0 2^{b_1 t}, \qquad t \geq 0,$$

where y is the measurement at time t, b_0 is the initial measurement, and $1/b_1$ is the time to double. That is, taking logarithms base 2 of both sides, we can write the model as

$$\log_2(y) = \log_2(b_0) + b_1 t, \qquad t \geq 0, \tag{7.2}$$

a linear model for logarithm of y at time t.

In this example, we fit an exponential model to computer processor speed. The data file "CPUspeed.txt" contains the maximum Intel CPU speed vs time from 1994 through 2004. The variables are:

Variable	*Description*
year	calendar year
month	month
day	day
time	time in years
speed	Max IA-32 Speed (GHz)
log10speed	logarithm base 10 of speed

The following code reads in the data from the file "CPUspeed.txt,"

```
> CPUspeed = read.table("CPUspeed.txt", header=TRUE)
```

and `head` displays the first few observations.

[2] http://firstdraftofhistory.theatlantic.com/analysis/internet_is_good.php

```
> head(CPUspeed)

  year month day     time speed log10speed
1 1994     3   7 1994.179 0.100 -1.0000000
2 1995     3  27 1995.233 0.120 -0.9208188
3 1995     6  12 1995.444 0.133 -0.8761484
4 1996     1   4 1996.008 0.166 -0.7798919
5 1996     6  10 1996.441 0.200 -0.6989700
6 1997     5   7 1997.347 0.300 -0.5228787
```

If Moore's Law holds (if an exponential model is correct), then we should expect that the logarithm of speed vs time follows an approximately linear trend. Here the base 2 logarithm is natural because of the base 2 logarithm in the proposed model (7.2), so we apply the change of base formula $\log_2(x) = \log_{10}(x)/\log_{10}(2)$. Since the earliest observation is in year 1994, we compute the time (`years`) measured in years since the start of 1994.

```
> years = CPUspeed$time - 1994
> speed = CPUspeed$speed
> log2speed = CPUspeed$log10speed / log10(2)
```

We construct scatterplots of speed vs time and $\log_2$(speed) vs time (where time is the time elapsed since 1994).

```
> plot(years, speed)
> plot(years, log2speed)
```

The plots are displayed in Figure 7.11(a) and 7.11(b). Figure 7.11(b) suggests that a linear association may exist between `log2speed` and `years`, so we fit a linear model using the `lm` function.

```
> L = lm(log2speed ~ years)
> print(L)

Call:
lm(formula = log2speed ~ years)

Coefficients:
(Intercept)        years
    -3.6581       0.5637
```

The fitted model is $\widehat{\ln y} = -3.6581 + 0.5637t$ or

$$\hat{y} = 2^{-3.6581+0.5637t} = 0.0792\,(2^{0.5637t}). \tag{7.3}$$

At time $t = 1/0.5637 = 1.774$ years, the predicted speed is $\hat{y} = 0.0792(2)$; thus, expected speed will double in an estimated 1.774 years. According to this model, CPU speeds are predicted to increase by a factor of $2^{0.5637(10)} \approx 50$ in 10 years (about 50 times faster, rather than 100 times faster as claimed in the interview).

To add the fitted regression curve to the plot in Figure 7.11(a) the `curve` function can be used with the exponential model (7.3).

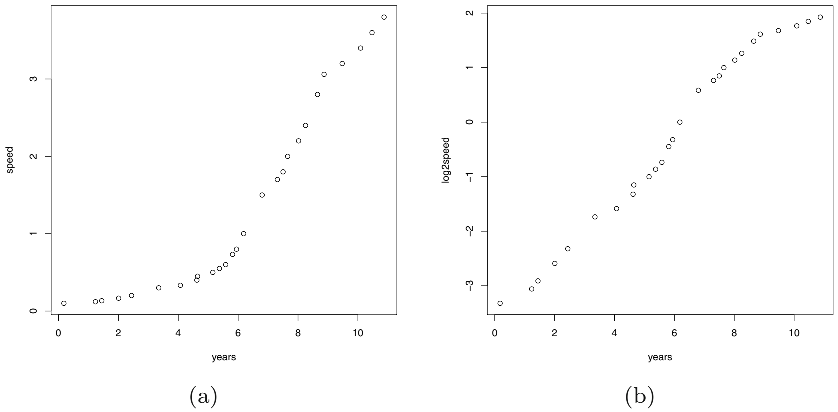


Fig. 7.11 Plot of CPU speed vs time (a) and log2(CPU speed) vs time (b) in Example 7.9

```
> plot(years, speed)
> curve(2^(-3.6581 + 0.5637 * x), add=TRUE)
```

To add the fitted regression line to the plot in Figure 7.11(b) the `abline` function can be used.

```
> plot(years, log2speed)
> abline(L)
```

These two plots are displayed in Figures 7.12(a) and 7.12(b).

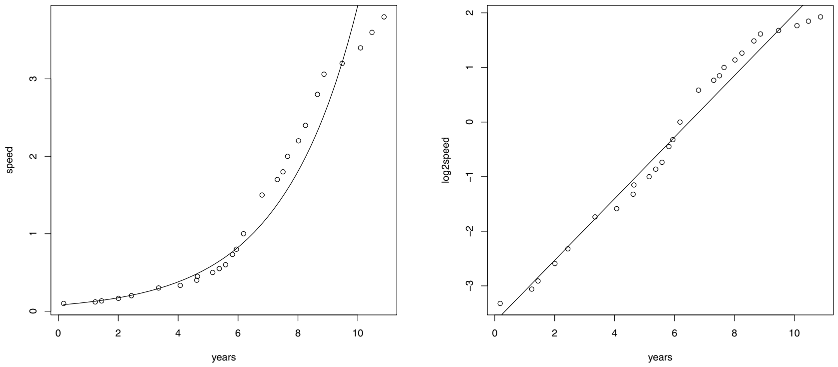
Moore's Law: residual analysis

The fit of the model appears to be good from a visual inspection of the fitted line in Figure 7.11(b). The model adequacy can be investigated further through residual plots.

The residuals are observed errors $e_i = y_i - \hat{y}_i$, where y_i is the observed response and $\hat{y}_i$ is the fitted value for observation i. The `lm` function returns an object containing residuals, fitted values, and other values. If we store the model rather than print it, we can access the residuals and other data returned by `lm`. In addition we can use available methods such as `summary`, `anova`, or `plot`.

The `plot` method for `lm` objects displays several residual plots. Using the argument `which=1:2` selects the first two plots.

```
> plot(L, which=1:2)
```



(a) (b)

Fig. 7.12 Plot of CPU speed vs time with fitted regression curve (a) and log2(CPU speed) vs time with fitted regression line (b) in Example 7.9

The two residual plots are shown in Figure 7.13. Figure 7.13(a) is a plot of residuals vs fitted values, with a curve added that has been fit to the data using a local regression "smoother" (see `lowess`). Figure 7.13(b) is a normal QQ plot of residuals.

An assumption for inference in regression is that the errors are independent and identically distributed (iid) as Normal(0,σ^2), but the normal QQ plot suggests that the residuals are not normal. In Figure 7.13(a) it appears that the residuals are not iid.

Three larger residuals are identified on the plot of residuals vs fitted values (observations 16, 26, 27) and the same points are identified on the QQ plot. These are

```
> CPUspeed[c(16, 26, 27), ]

   year month day     time speed log10speed
16 2000    10  20 2000.802   1.5  0.1760913
26 2004     6  21 2004.471   3.6  0.5563025
27 2004    11  15 2004.873   3.8  0.5797836
```

Observations 26 and 27 are the two most recent, possibly indicating that the model is not a good fit for the near future.

The summary method produces additional information about the model fit. Suppose that one only needs the coefficient of determination R^2, rather than the complete output of `summary`. For simple linear regression extract `$r.squared` from the summary, and for multiple linear regression extract `$adj.r.squared` (adjusted R^2).

```
> summary(L)$r.squared
[1] 0.9770912
```

The coefficient of determination is 0.9770912; more than 97.7% of the total variation in the logarithm of speed is explained by the model.

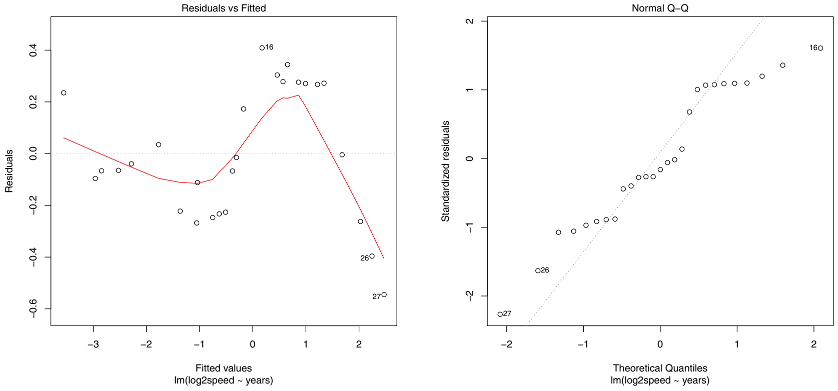
On November 13, 2005 the maximum processor speed was 3.8 GHz. What does the fitted linear regression model predict for expected maximum speed at this time? The `predict` method for `lm` objects returns the predicted values for the data or new observations. The new observations should be stored in a data frame, using the same names for the predictor variables as in the model formula. In this case the data frame has a single variable, `years`. The fractional year up to November 13 should be expressed as a decimal. If we take the fractional year elapsed in 2005 to be 316.5/365, we have

```
> new = data.frame(years = 2005 + 316.5 / 365 - 1994)
> lyhat = predict(L, newdata=new)
> lyhat
       1
3.031005
```

Recall that the response variable in the fitted model is $\log_2$(speed), so the speed predicted by the model on November 13, 2005 is

```
> 2^lyhat
       1
8.173792
```

GHz and the error is 8.2 - 3.8 = 4.4 GHz. This illustrates the danger of extrapolation; note that our latest observation in `CPUspeed` is about one full year earlier, Nov. 15, 2004.



(a) (b)

Fig. 7.13 Residual plots from the fitted regression model, log2(CPU speed) vs time, in Example 7.9.

Exercises

7.1 (*mammals* data). The mammals data set in the MASS package records brain size and body size for 62 different mammals. Fit a regression model to describe the relation between brain size and body size. Display a residual plot using the plot method for the result of the lm function. Which observation (which mammal) has the largest residual in your fitted model?

7.2 (*mammals*, continued). Refer to the mammals data in package MASS. Display a scatterplot of log(brain) vs log(body). Fit a simple linear regression model to the transformed data. What is the equation of the fitted model? Display a fitted line plot and comment on the fit. Compare your results with results of Exercise 7.1.

7.3 (*mammals* residuals). Refer to Exercise 7.2. Display a plot of residuals vs fitted values and a normal-QQ plot of residuals. Do the residuals appear to be approximately normally distributed with constant variance?

7.4 (*mammals* summary statistics). Refer to Exercise 7.2. Use the summary function on the result of lm to display the summary statistics for the model. What is the estimate of the error variance? Find the coefficient of determination (R^2) and compare it to the square of the correlation between the response and predictor. Interpret the value of (R^2) as a measure of fit.

7.5 (Hubble's Law). In 1929 Edwin Hubble investigated the relationship between distance and velocity of celestial objects. Knowledge of this relationship might give clues as to how the universe was formed and what may happen in the future. Hubble's Law is is

$$\text{Recession Velocity} = H_0 \times \text{Distance},$$

where H_0 is Hubble's constant. This model is a straight line through the origin with slope H_0. Data that Hubble used to estimate the constant H_0 are given on the DASL web at http://lib.stat.cmu.edu/DASL/Datafiles/Hubble.html. Use the data to estimate Hubble's constant by simple linear regression.

7.6 (*peanuts* data). The data file "peanuts.txt" (Hand et al. [21]) records levels of a toxin in batches of peanuts. The data are the average level of aflatoxin X in parts per billion, in 120 pounds of peanuts, and percentage of non-contaminated peanuts Y in the batch. Use a simple linear regression model to predict Y from X. Display a fitted line plot. Plot residuals, and comment on the adequacy of the model. Obtain a prediction of percentage of non-contaminated peanuts at levels 20, 40, 60, and 80 of aflatoxin.

7.7 (*cars* data). For the cars data in Example 7.1, compare the coefficient of determination R^2 for the two models (with and without intercept term in the model). Hint: Save the fitted model as L and use summary(L) to display R^2. Interpret the value of R^2 as a measure of the fit.

7.8 (*cars* data, continued). Refer to the `cars` data in Example 7.1. Create a new variable `speed2` equal to the square of `speed`. Then use `lm` to fit a quadratic model

$$dist = \beta_0 + \beta_1 speed + \beta_2 (speed)^2 + \varepsilon.$$

The corresponding model formula would be `dist ~ speed + speed2`. Use `curve` to add the estimated quadratic curve to the scatterplot of the data and comment on the fit. How does the fit of the model compare with the simple linear regression model of Example 7.1 and Exercise 7.7?

7.9 (Cherry Tree data, quadratic regression model). Refer to the Cherry Tree data in Example 7.3. Fit and analyze a quadratic regression model $y = b_0 + b_1 x + b_2 x^2$ for predicting volume y given diameter x. Check the residual plots and summarize the results.

7.10 (*lunatics* data). Refer to the "lunatics" data in Example 7.8. Repeat the analysis, after deleting the two counties that are offshore islands, `NANTUCKET` and `DUKES` counties. Compare the estimates of slope and intercept with those obtained in Example 7.8. Construct the plots and analyze the residuals as in Example 7.8.

7.11 (*twins* data). Import the data file "twins.txt" using `read.table`. (The commands to read this data file are shown in the twins example in Section 3.3, page 85.) The variable `DLHRWAGE` is the difference (twin 1 minus twin 2) in the logarithm of hourly wage, given in dollars. The variable `HRWAGEL` is the hourly wage of twin 1. Fit and analyze a simple linear regression model to predict the difference `DLHRWAGE` given the *logarithm* of the hourly wage of twin 1.