Regression

13

Classification tries to predict a class from a data item. **Regression** tries to predict a value. For example, we know the zip code of a house, the square footage of its lot, the number of rooms and the square footage of the house, and we wish to predict its likely sale price. As another example, we know the cost and condition of a trading card for sale, and we wish to predict a likely profit in buying it and then reselling it. As yet another example, we have a picture with some missing pixels—perhaps there was text covering them, and we want to replace it—and we want to fill in the missing values. As a final example, you can think of classification as a special case of regression, where we want to predict either $+1$ or -1; this isn't usually the best way to classify, however. Predicting values is very useful, and so there are many examples like this.

Some formalities are helpful here. In the simplest case, we have a dataset consisting of a set of N pairs $(\mathbf{x}_i, y_i)$. We want to use the examples we have—the **training examples**—to build a model of the dependence between y and $\mathbf{x}$. This model will be used to predict values of y for new values of $\mathbf{x}$, which are usually called **test examples**. We think of y_i as the value of some function evaluated at $\mathbf{x}_i$, but with some random component. This means there might be two data items where the $\mathbf{x}_i$ are the same, and the y_i are different. We refer to the $\mathbf{x}_i$ as **explanatory variables** and the y_i as a **dependent variable**. We regularly say that we are regressing the dependent variable against the explanatory variables.

13.1 Regression to Make Predictions

Now imagine that we have one independent variable. An appropriate choice of $\mathbf{x}$ and of model (details below) will mean that the predictions made by this model will lie on a straight line. Figure 13.1 shows two regressions. The data are plotted with a scatter plot, and the line gives the prediction of the model for each value on the x axis.

We cannot guarantee that different values of $\mathbf{x}$ produce different values of y. Data just isn't like this (see the crickets example Fig. 13.1). This means you can't think of a regression as predicting the true value of y from $\mathbf{x}$ because usually there isn't one. Instead, you should think of a regression as predicting the expected value of y conditioned on $\mathbf{x}$. Some regression models can produce more information about the probability distribution for y conditioned on $\mathbf{x}$. For example, it might be very valuable to get both the mean and variance of the distribution of the likely sale value of a house from independent variables.

It should be clear that none of this will work if there is not some relationship between the training examples and the test examples. If I collect training data on the height and weight of children, I'm unlikely to get good predictions of the weight of adults from their height. We can be more precise with a probabilistic framework. We think of $\mathbf{x}_i$ as IID samples from some (usually unknown) probability distribution $P(X)$. Then the test examples should also be IID samples from $P(X)$, or, at least, rather like them—you usually can't check this point with any certainty.

A probabilistic formalism can help be precise about the y_i, too. Assume another random variable Y has joint distribution with X given by $P(Y, X)$. We think of each y_i as a sample from $P(Y|\{X = \mathbf{x}_i\})$. Then our modelling problem would be: given the training data, build a model that takes a test example $\mathbf{x}$ and yields $\mathbb{E}[Y|\{X = \mathbf{x}_i\}]$.

D. Forsyth, *Probability and Statistics for Computer Science*,
https://doi.org/10.1007/978-3-319-64410-3_13

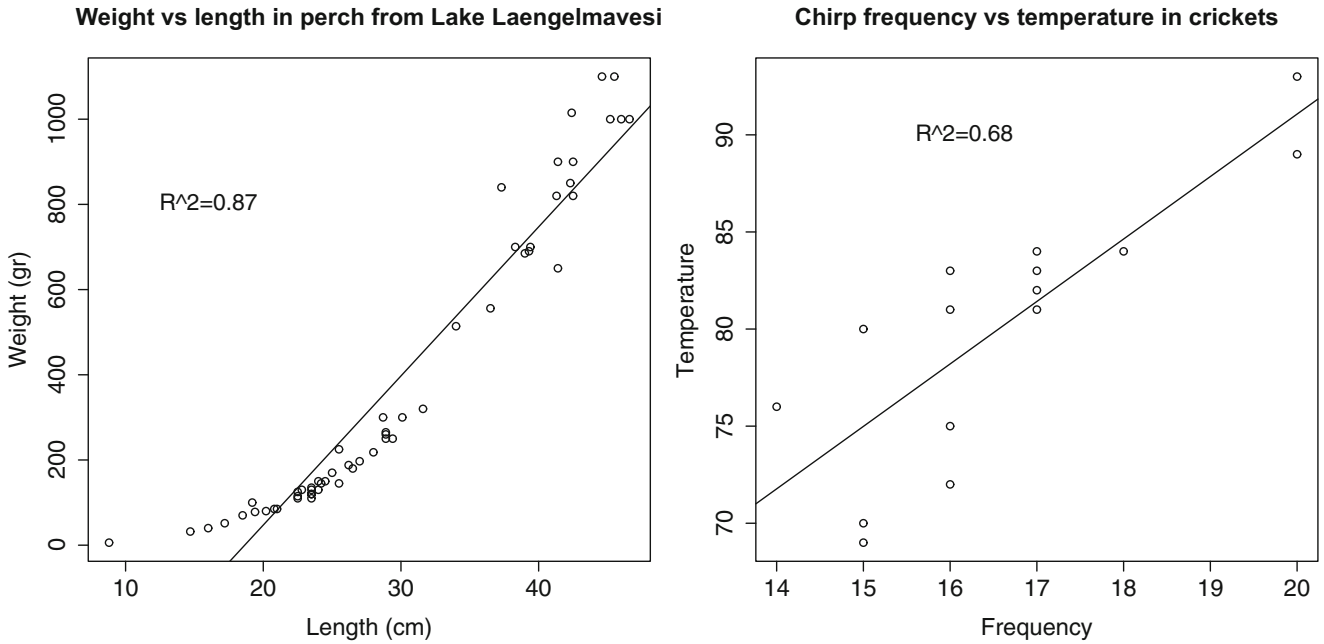

Fig. 13.1 On the *left*, a regression of weight against length for perch from a Finnish lake (you can find this dataset, and the back story at http://www.amstat.org/publications/jse/jse_data_archive.htm; look for "fishcatch" on that page). Notice that the linear regression fits the data fairly well, meaning that you should be able to predict the weight of a perch from its length fairly well. On the *right*, a regression of air temperature against chirp frequency for crickets. The data is fairly close to the line, meaning that you should be able to tell the temperature from the pitch of cricket's chirp fairly well. This data is from http://mste.illinois.edu/patel/amar430/keyprob1.html. The R^2 you see on each figure is a measure of the goodness of fit of the regression (Sect. 13.3.5)

Thinking about the problem this way should make it clear that we're not relying on any exact, physical, or causal relationship between Y and X. It's enough that their joint probability makes useful predictions possible, something we will test by experiment. This means that you can build regressions that work in somewhat surprising circumstances. For example, regressing childrens' reading ability against their foot size can be quite successful. This isn't because having big feet somehow helps you read. It's because on the whole, older children read better, and also have bigger feet. Regression isn't magic. Figure 13.2 shows two regressions where the predictions aren't particularly accurate.

13.2 Regression to Spot Trends

Regression isn't only used to predict values. Another reason to build a regression model is to compare trends in data. Doing so can make it clear what is really happening. Here is an example from Efron ("Computer-Intensive methods in statistical regression", B. Efron, SIAM Review, 1988). The table in the appendix shows some data from medical devices, which sit in the body and release a hormone. The data shows the amount of hormone currently in a device after it has spent some time in service, and the time the device spent in service. The data describes devices from three production lots (A, B, and C). Each device, from each lot, is supposed to have the same behavior. The important question is: Are the lots the same? The amount of hormone changes over time, so we can't just compare the amounts currently in each device. Instead, we need to determine the relationship between time in service and hormone, and see if this relationship is different between batches. We can do so by regressing hormone against time.

Figure 13.3 shows how a regression can help. In this case, we have modelled the amount of hormone in the device as

$$a \times (\text{time in service}) + b$$

for a, b chosen to get the best fit (much more on this point later!). This means we can plot each data point on a scatter plot, together with the best fitting line. This plot allows us to ask whether any particular batch behaves differently from the overall model in any interesting way.

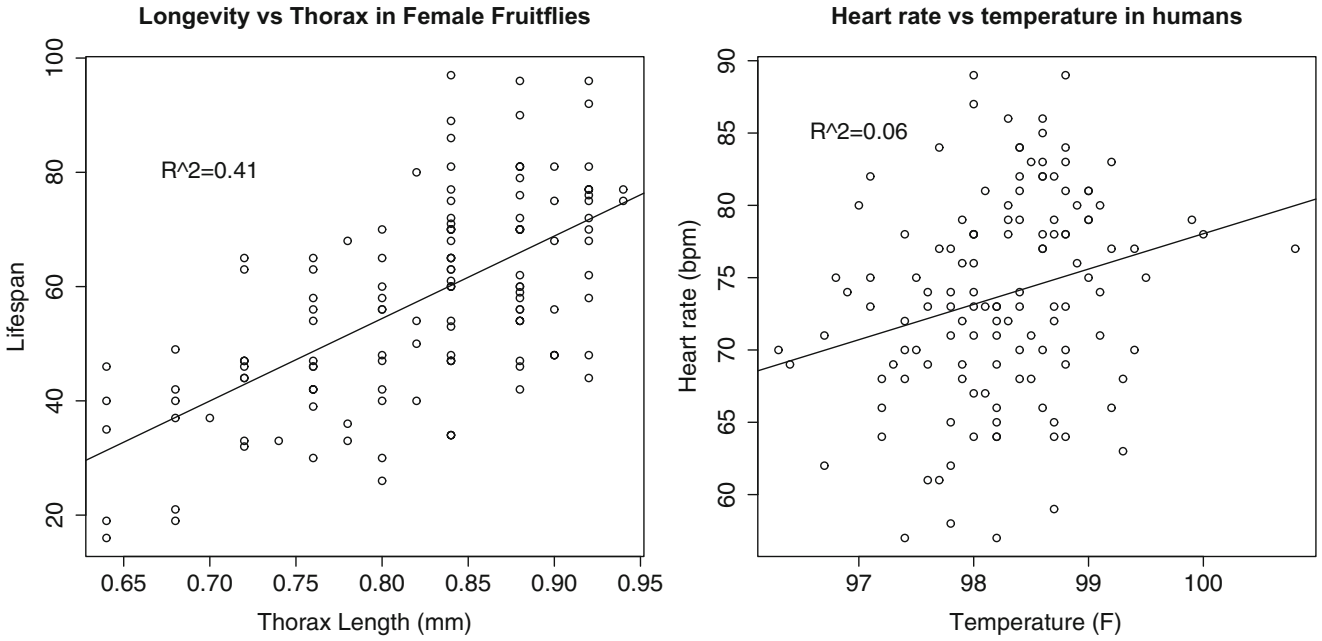

Fig. 13.2 Regressions do not necessarily yield good predictions or good model fits. On the *left*, a regression of the lifespan of female fruitflies against the length of their torso as adults (apparently, this doesn't change as a fruitfly ages; you can find this dataset, and the back story at http://www.amstat.org/publications/jse/jse_data_archive.htm; look for "fruitfly" on that page). The figure suggests you can make some prediction of how long your fruitfly will last by measuring its torso, but not a particularly accurate one. On the *right*, a regression of heart rate against body temperature for adults. You can find the data at http://www.amstat.org/publications/jse/jse_data_archive.htm as well; look for "temperature" on that page. Notice that predicting heart rate from body temperature isn't going to work that well, either

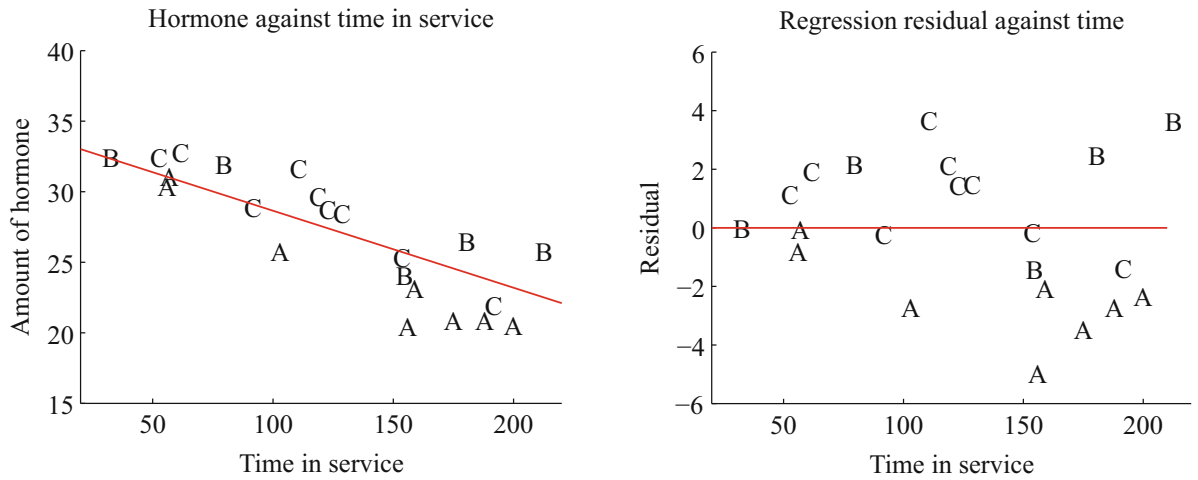

Fig. 13.3 On the *left*, a scatter plot of hormone against time for devices from Tables 13.1 and 13.1. Notice that there is a pretty clear relationship between time and amount of hormone (the longer the device has been in service the less hormone there is). The issue now is to understand that relationship so that we can tell whether lots A, B and C are the same or different. The best fit line to all the data is shown as well, fitted using the methods of Sect. 13.3. On the *right*, a scatter plot of residual—the distance between each data point and the best fit line—against time for the devices from Tables 13.1 and 13.1. Now you should notice a clear difference; some devices from lots B and C have positive and some negative residuals, but all lot A devices have negative residuals. This means that, when we account for loss of hormone over time, lot A devices still have less hormone in them. This is pretty good evidence that there is a problem with this lot

However, it is hard to evaluate the distances between data points and the best fitting line by eye. A sensible alternative is to subtract the amount of hormone predicted by the model from the amount that was measured. Doing so yields a residual—the difference between a measurement and a prediction. We can then plot those residuals (Fig. 13.3). In this case, the plot suggests that lot A is special—all devices from this lot contain less hormone than our model predicts.

Definition 13.2 (Regression) Regression accepts a feature vector and produces a prediction, which is usually a number, but can sometimes have other forms. You can use these predictions as predictions, or to study trends in data. It is possible, but not usually particularly helpful, to see classification as a form of regression.

13.3 Linear Regression and Least Squares

Assume we have a dataset consisting of a set of N pairs $(\mathbf{x}_i, y_i)$. We want to use the examples we have—the training examples—to build a model of the dependence between y and $\mathbf{x}$. This model will be used to predict values of y for new values of $\mathbf{x}$, which are usually called test examples. The model needs to have some probabilistic component; we do not expect that y is a function of $\mathbf{x}$, and there is likely some error in evaluating y anyhow.

13.3.1 Linear Regression

We cannot expect that our model makes perfect predictions. Furthermore, y may not be a function of $\mathbf{x}$—it is quite possible that the same value of $\mathbf{x}$ could lead to different y's. One way that this could occur is that y is a measurement (and so subject to some measurement noise). Another is that there is some randomness in y. For example, we expect that two houses with the same set of features (the $\mathbf{x}$) might still sell for different prices (the y's).

A good, simple model is to assume that the dependent variable (i.e. y) is obtained by evaluating a linear function of the explanatory variables (i.e. $\mathbf{x}$), then adding a zero-mean normal random variable. We can write this model as

$$y = \mathbf{x}^T\beta + \xi$$

where ξ represents random (or at least, unmodelled) effects. In this expression, β is a vector of weights, which we must estimate. We will always assume that ξ has zero mean, so that

$$\mathbb{E}[Y|\{X = \mathbf{x}_i\}] = \mathbf{x}_i\beta.$$

When we use this model to predict a value of y for a particular set of explanatory variables $\mathbf{x}^*$, we cannot predict the value that ξ will take. Our best available prediction is the mean value (which is zero). Notice that if $\mathbf{x} = 0$, the model predicts $y = 0$. This may seem like a problem to you—you might be concerned that we can fit only lines through the origin—but remember that $\mathbf{x}$ contains explanatory variables, and we can choose what appears in $\mathbf{x}$. The two examples show how a sensible choice of $\mathbf{x}$ allows us to fit a line with an arbitrary y-intercept.

Definition 13.3 (Linear Regression) A linear regression takes the feature vector $\mathbf{x}$ and predicts $\mathbf{x}^T\beta$, for some vector of coefficients β. The coefficients are adjusted, using data, to produce the best predictions.

Example 13.3 (A Linear Model Fitted to a Single Explanatory Variable) Assume we fit a linear model to a single explanatory variable. Then the model has the form $y = x\beta + \xi$, where ξ is a zero mean random variable. For any value x^* of the explanatory variable, our best estimate of y is βx^*. In particular, if $x^* = 0$, the model predicts $y = 0$, which is unfortunate. We can draw the model by drawing a line through the origin with slope β in the x, y plane. The y-intercept of this line must be zero.

Example 13.4 (A Linear Model with a Non-Zero y-Intercept) Assume we have a single explanatory variable, which we write u. We can then create a *vector* $\mathbf{x} = [u, 1]^T$ from the explanatory variable. We now fit a linear model to this vector. Then the model has the form $y = \mathbf{x}^T\beta + \xi$, where ξ is a zero mean random variable. For any value $\mathbf{x}^* = [u^*, 1]^T$ of the explanatory variable, our best estimate of y is $(\mathbf{x}^*)^T\beta$, which can be written as $y = \beta_1 u^* + \beta_2$. If $x^* = 0$, the model predicts $y = \beta_2$. We can draw the model by drawing a line through the origin with slope β_1 and y-intercept β_2 in the x, y plane.

13.3.2 Choosing β

We must determine β. We can proceed in two ways. I show both because different people find different lines of reasoning more compelling. Each will get us to the same solution. One is probabilistic, the other isn't. Generally, I'll proceed as if they're interchangeable, although at least in principle they're different.

Probabilistic approach: we could assume that ξ is a zero mean normal random variable with unknown variance. Then $P(y|x, \beta)$ is normal, with mean $\mathbf{x}^T\beta$, and so we can write out the log-likelihood of the data. Write σ^2 for the variance of ξ, which we don't know, but will not worry about right now. We have that

$$\begin{aligned}\log \mathcal{L}(\beta) &= -\sum_i \log P(y_i|\mathbf{x}_i, \beta) \\ &= \frac{1}{2\sigma^2}\sum_i (y_i - \mathbf{x}_i^T\beta)^2 \\ &\quad + \text{term not depending on } \beta\end{aligned}$$

Maximizing the log-likelihood of the data is equivalent to minimizing the negative log-likelihood of the data. Furthermore, the term $\frac{1}{2\sigma^2}$ does not affect the location of the minimum, so we must have that β minimizes $\sum_i (y_i - \mathbf{x}_i^T\beta)^2$, or anything proportional to it. It is helpful to minimize an expression that is an average of squared errors, because (hopefully) this doesn't grow much when we add data. We therefore minimize

$$\left(\frac{1}{N}\right)\left(\sum_i (y_i - \mathbf{x}_i^T\beta)^2\right).$$

Direct approach: notice that, if we have an estimate of β, we have an estimate of the values of the unmodelled effects ξ_i for each example. We just take $\xi_i = y_i - \mathbf{x}_i^T\beta$. It is quite natural to make the unmodelled effects "small". A good measure of size is the mean of the squared values, which means we want to minimize

$$\left(\frac{1}{N}\right)\left(\sum_i (y_i - \mathbf{x}_i^T\beta)^2\right).$$

13.3.3 Solving the Least Squares Problem

We can write all this more conveniently using vectors and matrices. Write $\mathbf{y}$ for the vector

$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

and $\mathcal{X}$ for the matrix

$$\begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \dots \mathbf{x}_n^T \end{pmatrix}.$$

Then we want to minimize

$$\left(\frac{1}{N}\right)(\mathbf{y} - \mathcal{X}\beta)^T(\mathbf{y} - \mathcal{X}\beta)$$

which means that we must have

$$\mathcal{X}^T\mathcal{X}\beta - \mathcal{X}^T\mathbf{y} = 0.$$

For reasonable choices of features, we could expect that $\mathcal{X}^T\mathcal{X}$—which should strike you as being a lot like a covariance matrix—has full rank. If it does, which is the usual case, this equation is easy to solve. If it does not, there is more to do, which we will do in Sect. 13.4.4.

Remember this: *The vector of coefficients β for a linear regression is usually estimated using a least-squares procedure.*

13.3.4 Residuals

Assume we have produced a regression by solving

$$\mathcal{X}^T\mathcal{X}\hat{\beta} - \mathcal{X}^T\mathbf{y} = 0$$

for the value of $\hat{\beta}$. I write $\hat{\beta}$ because this is an *estimate*; we likely don't have the true value of the β that generated the data (the model might be wrong; etc.). We cannot expect that $\mathcal{X}\hat{\beta}$ is the same as $\mathbf{y}$. Instead, there is likely to be some error. The **residual** is the vector

$$\mathbf{e} = \mathbf{y} - \mathcal{X}\hat{\beta}$$

which gives the difference between the true value and the model's prediction at each point. Each component of the residual is an estimate of the unmodelled effects for that data point. The **mean square error** is

$$m = \frac{\mathbf{e}^T\mathbf{e}}{N}$$

and this gives the average of the squared error of prediction on the training examples.

Notice that the mean squared error is not a great measure of how good the regression is. This is because the value depends on the units in which the dependent variable is measured. So, for example, if you measure y in meters you will get a different mean squared error than if you measure y in kilometers *for the same dataset*. This is a serious nuisance, because it means that the value of the mean squared error cannot tell you how good a regression is. There is an alternative measure of the accuracy of a regression which does not depends on the units of y.

13.3.5 R-Squared

Unless the dependent variable is a constant (which would make prediction easy), it has some variance. If our model is of any use, it should explain some aspects of the value of the dependent variable. This means that the variance of the residual should be smaller than the variance of the dependent variable. If the model made perfect predictions, then the variance of the residual should be zero.

We can formalize all this in a relatively straightforward way. We will ensure that $\mathcal{X}$ always has a column of ones in it, so that the regression can have a non-zero y-intercept. We now fit a model

$$\mathbf{y} = \mathcal{X}\beta + \mathbf{e}$$

(where $\mathbf{e}$ is the vector of residual values) by choosing β such that $\mathbf{e}^T\mathbf{e}$ is minimized. Then we get some useful technical results.

Useful Facts 13.1 (Regression)
We write $\mathbf{y} = \mathcal{X}\hat{\beta} + \mathbf{e}$, where $\mathbf{e}$ is the residual. For a vector $\mathbf{v}$ of N components, we write $\overline{\mathbf{v}} = (1/N)\mathbf{1}^T\mathbf{v}$. Assume $\mathcal{X}$ has a column of ones, and $\hat{\beta}$ is chosen to minimize $\mathbf{e}^T\mathbf{e}$. Then we have

1. $\mathbf{e}^T\mathcal{X} = \mathbf{0}$, i.e. that $\mathbf{e}$ is orthogonal to any column of $\mathcal{X}$. If $\mathbf{e}$ is not orthogonal to some column of $\mathbf{e}$, we can increase or decrease the $\hat{\beta}$ term corresponding to that column to make the error smaller. Another way to see this is to notice that $\hat{\beta}$ is chosen to minimize $\frac{1}{N}\mathbf{e}^T\mathbf{e}$, which is $\frac{1}{N}(\mathbf{y} - \mathcal{X}\hat{\beta})^T(\mathbf{y} - \mathcal{X}\hat{\beta})$. Now because this is a minimum, the gradient with respect to $\hat{\beta}$ is zero, so $(\mathbf{y} - \mathcal{X}\hat{\beta})^T(-\mathcal{X}) = -\mathbf{e}^T\mathcal{X} = 0$.
2. $\mathbf{e}^T\mathbf{1} = 0$ (recall that $\mathcal{X}$ has a column of all ones, and apply the previous result).
3. $\mathbf{e}^T\mathcal{X}\hat{\beta} = 0$ (first result means that this is true).
4. $\mathbf{1}^T(\mathbf{y} - \mathcal{X}\hat{\beta}) = 0$ (same as previous result).
5. $\overline{\mathbf{y}} = \overline{\mathcal{X}\hat{\beta}}$ (same as previous result).

Now $\mathbf{y}$ is a one dimensional dataset arranged into a vector, so we can compute $\mathsf{mean}\,(\{y\})$ and $\mathsf{var}[y]$. Similarly, $\mathcal{X}\hat{\beta}$ is a one dimensional dataset arranged into a vector (its elements are $\mathbf{x}_i^T\hat{\beta}$), as is $\mathbf{e}$, so we know the meaning of mean and variance for each. We have a particularly important result:

$$\mathsf{var}[y] = \mathsf{var}\left[\mathcal{X}\hat{\beta}\right] + \mathsf{var}[e].$$

This is quite easy to show, with a little more notation. Write $\overline{\mathbf{y}} = (1/N)(\mathbf{1}^T\mathbf{y})\mathbf{1}$ for the vector whose entries are all $\mathsf{mean}\,(\{y\})$; similarly for $\overline{\mathbf{e}}$ and for $\overline{\mathcal{X}\hat{\beta}}$. We have

$$\mathsf{var}[y] = (1/N)(\mathbf{y} - \overline{\mathbf{y}})^T(\mathbf{y} - \overline{\mathbf{y}})$$

and so on for $\mathsf{var}[e_i]$, etc. Notice from the facts that $\overline{\mathbf{y}} = \overline{\mathcal{X}\hat{\beta}}$. Now

$$\begin{aligned}
\mathsf{var}[y] &= (1/N)\left(\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right] + [\mathbf{e} - \overline{\mathbf{e}}]\right)^T\left(\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right] + [\mathbf{e} - \overline{\mathbf{e}}]\right) \\
&= (1/N)\left(\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right]^T\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right] + 2\,[\mathbf{e} - \overline{\mathbf{e}}]^T\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right] + [\mathbf{e} - \overline{\mathbf{e}}]^T[\mathbf{e} - \overline{\mathbf{e}}]\right) \\
&= (1/N)\left(\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right]^T\left[\mathcal{X}\hat{\beta} - \overline{\mathcal{X}\hat{\beta}}\right] + [\mathbf{e} - \overline{\mathbf{e}}]^T[\mathbf{e} - \overline{\mathbf{e}}]\right) \\
&\quad \text{because } \overline{\mathbf{e}} = 0 \text{ and } \mathbf{e}^T\mathcal{X}\hat{\beta} = 0 \text{ and } \mathbf{e}^T\mathbf{1} = 0 \\
&= \mathsf{var}\left[\mathcal{X}\hat{\beta}\right] + \mathsf{var}[e].
\end{aligned}$$

This is extremely important, because us allows us to think about a regression as explaining variance in $\mathbf{y}$. As we are better at explaining $\mathbf{y}$, $\mathsf{var}[e]$ goes down. In turn, a natural measure of the goodness of a regression is what percentage of the variance of $\mathbf{y}$ it explains. This is known as R^2 (the r-squared measure). We have

$$R^2 = \frac{\mathsf{var}\left[\mathbf{x}_i^T\hat{\beta}\right]}{\mathsf{var}[y_i]}$$

which gives some sense of how well the regression explains the training data. Notice that the value of R^2 is not affected by the units of $\mathbf{y}$ (exercises)

Good predictions result in high values of R^2, and a perfect model will have $R^2 = 1$ (which doesn't usually happen). For example, the regression of Fig. 13.3 has an R^2 value of 0.87. Figures 13.1 and 13.2 show the R^2 values for the regressions plotted there; notice how better models yield larger values of R^2. Notice that if you look at the summary that R provides for a linear regression, it will offer you *two* estimates of the value for R^2. These estimates are obtained in ways that try to account for (a) the amount of data in the regression, and (b) the number of variables in the regression. For our purposes, the differences between these numbers and the R^2 I defined are not significant. For the figures, I computed R^2 as I described in the text above, but if you substitute one of R's numbers nothing terrible will happen.

Remember this: *The quality of predictions made by a regression can be evaluated by looking at the fraction of the variance in the dependent variable that is explained by the regression. This number is called R^2, and lies between zero and one; regressions with larger values make better predictions.*

Procedure 13.1 (Linear Regression Using Least Squares) We have a dataset containing N pairs $(\mathbf{x}_i, y_i)$. Each x_i is a d-dimensional explanatory vector, and each y_i is a single dependent variable. We assume that each data point conforms to the model

$$y_i = \mathbf{x}_i^T \beta + \xi_i$$

where ξ_i represents unmodelled effects. We assume that ξ_i are samples of a random variable with 0 mean and unknown variance. Sometimes, we assume the random variable is normal. Write

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} \text{ and } \mathcal{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \dots \\ \mathbf{x}_n^T \end{pmatrix}.$$

We estimate $\hat{\beta}$ (the value of β) by solving the linear system

$$\mathcal{X}^T \mathcal{X} \hat{\beta} - \mathcal{X}^T \mathbf{y} = 0.$$

For a data point $\mathbf{x}$, our model predicts $\mathbf{x}^T \hat{\beta}$. The residuals are

$$\mathbf{e} = \mathbf{y} - \mathcal{X} \hat{\beta}.$$

We have that $\mathbf{e}^T \mathbf{1} = 0$. The mean square error is given by

$$m = \frac{\mathbf{e}^T \mathbf{e}}{N}.$$

The R^2 is given by

$$\frac{\text{var}\left(\left\{\mathbf{x}_i^T \hat{\beta}\right\}\right)}{\text{var}\left(\{\mathbf{y}\}\right)}.$$

Values of R^2 range from 0 to 1; a larger value means the regression is better at explaining the data.

13.4 Producing Good Linear Regressions

Linear regression is useful, but it isn't magic. Some regressions make poor predictions (recall the regressions of Figure 13.2). As another example, regressing the first digit of your telephone number against the length of your foot won't work.

We have some straightforward tests to tell whether a regression is working. You can **look at a plot** for a dataset with one explanatory variable and one dependent variable. You plot the data on a scatter plot, then plot the model as a line on that scatterplot. Just looking at the picture can be informative (compare Figs. 13.1 and 13.2).

You can check if the regression **predicts a constant**. This is usually a bad sign. You can check this by looking at the predictions for each of the training data items. If the variance of these predictions is small compared to the variance of the independent variable, the regression isn't working well. If you have only one explanatory variable, then you can plot the regression line. If the line is horizontal, or close, then the value of the explanatory variable makes very little contribution to the prediction. This suggests that there is no particular relationship between the explanatory variable and the independent variable.

You can also check, by eye, if **the residual isn't random**. If $y - \mathbf{x}^T\beta$ is a zero mean normal random variable, then the value of the residual vector should not depend on the corresponding y-value. Similarly, if $y - \mathbf{x}^T\beta$ is just a zero mean collection of unmodelled effects, we want the value of the residual vector to not depend on the corresponding y-value either. If it does, that means there is some phenomenon we are not modelling. Looking at a scatter plot of **e** against **y** will often reveal trouble in a regression (Fig. 13.7). In the case of Fig. 13.7, the trouble is caused by a few data points that are very different from the others severely affecting the regression. We will discuss such points in more detail below. Once they have been removed, the regression improves markedly (Fig. 13.8).

Remember this: *Linear regressions can make bad predictions. You can check for trouble by: evaluating R^2; looking at a plot; looking to see if the regression makes a constant prediction; or checking whether the residual is random. Other strategies exist, but are beyond the scope of this book.*

13.4.1 Transforming Variables

Sometimes the data isn't in a form that leads to a good linear regression. In this case, transforming explanatory variables, the dependent variable, or both can lead to big improvements. Figure 13.4 shows one example, based on the idea of word frequencies. Some words are used very often in text; most are used seldom. The dataset for this figure consists of counts of the number of time a word occurred for the 100 most common words in Shakespeare's printed works. It was originally collected from a concordance, and has been used to attack a variety of interesting questions, including an attempt to assess how many words Shakespeare knew. This is hard, because he likely knew many words that he didn't use in his works, so one can't just count. If you look at the plot of Fig. 13.4, you can see that a linear regression of count (the number of times a word is used) against rank (how common a word is, 1–100) is not really useful. The most common words are used very often, and the number of times a word is used falls off very sharply as one looks at less common words. You can see this effect in the scatter plot of residual against dependent variable in Fig. 13.4—the residual depends rather strongly on the dependent variable. This is an extreme example that illustrates how poor linear regressions can be.

However, if we regress log-count against log-rank, we get a very good fit indeed. This suggests that Shakespeare's word usage (at least for the 100 most common words) is consistent with **Zipf's law**. This gives the relation between frequency f and rank r for a word as

$$f \propto \frac{1}{r}^s$$

where s is a constant characterizing the distribution. Our linear regression suggests that s is approximately 1.67 for this data.

In some cases, the natural logic of the problem will suggest variable transformations that improve regression performance. For example, one could argue that humans have approximately the same density, and so that weight should scale as the cube of height; in turn, this suggests that one regress weight against the cube root of height. Figure 13.5 shows the result of this transformation on the fish data, where it appears to help a lot. Generally, shorter people tend not to be scaled versions of taller

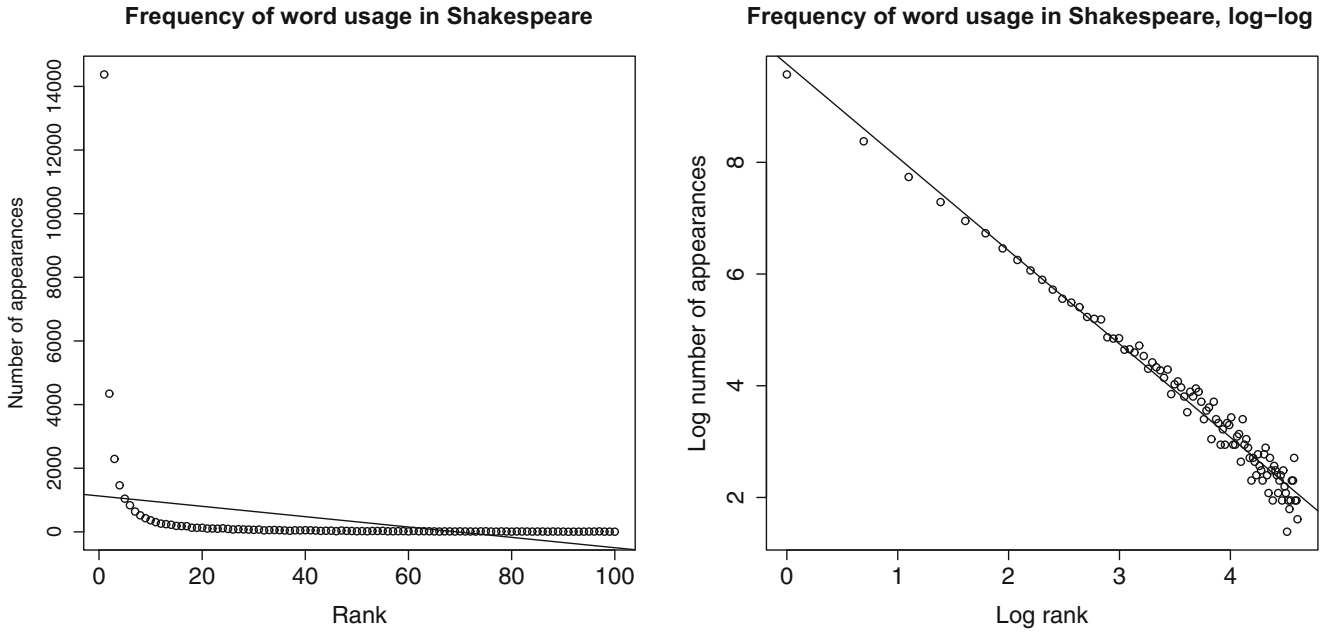

Fig. 13.4 On the *left*, word count plotted against rank for the 100 most common words in Shakespeare, using a dataset that comes with R (called "bard", and quite likely originating in an unpublished report by J. Gani and I. Saunders). I show a regression line too. This is a poor fit by eye, and the R^2 is poor, too ($R^2 = 0.1$). On the *right*, log word count plotted against log rank for the 100 most common words in Shakespeare, using a dataset that comes with R (called "bard", and quite likely originating in an unpublished report by J. Gani and I. Saunders). The regression line is very close to the data

people, so the cube root might be too aggressive. The body mass index (BMI: a controversial but not completely pointless measure of the relationship between weight and height) uses the square root.

Remember this: *The performance of a regression can be improved by transforming variables. Transformations can follow from looking at plots, or thinking about the logic of the problem*

13.4.2 Problem Data Points Have Significant Impact

Outlying data points can significantly weaken the usefulness of a regression. For some regression problems, we can identify data points that might be a problem, and then resolve how to deal with them. One possibility is that they are true outliers—someone recorded a data item wrong, or they represent an effect that just doesn't occur all that often. Another is that they are important data, and our linear model may not be good enough. If the data points really are outliers, we can drop them from the data set. If they aren't, we may be able to improve the regression by transforming features or by finding a new explanatory variable.

When we construct a regression, we are solving for the β that minimizes $\sum_i (y_i - \mathbf{x}_i^T \beta)^2$, equivalently for the β that produces the smallest value of $\sum_i e_i^2$. This means that residuals with large value can have a very strong influence on the outcome—we are squaring that large value, resulting in an enormous value. Generally, many residuals of medium size will have a smaller cost than one large residual and the rest tiny. As Fig. 13.6 illustrates, this means that a data point that lies far from the others can swing the regression line significantly (which affects the residual, Fig. 13.7).

This creates a problem, because data points that are very different from most others (sometimes called **outliers**) can also have the highest influence on the outcome of the regression. Figure 13.8 shows this effect for a simple case. When we have only one explanatory variable, there's an easy method to spot problem data points. We produce a scatter plot and a regression line, and the difficulty is usually obvious. In particularly tricky cases, printing the plot and using a see-through ruler to draw a line by eye can help (if you use an opaque ruler, you may not see some errors).

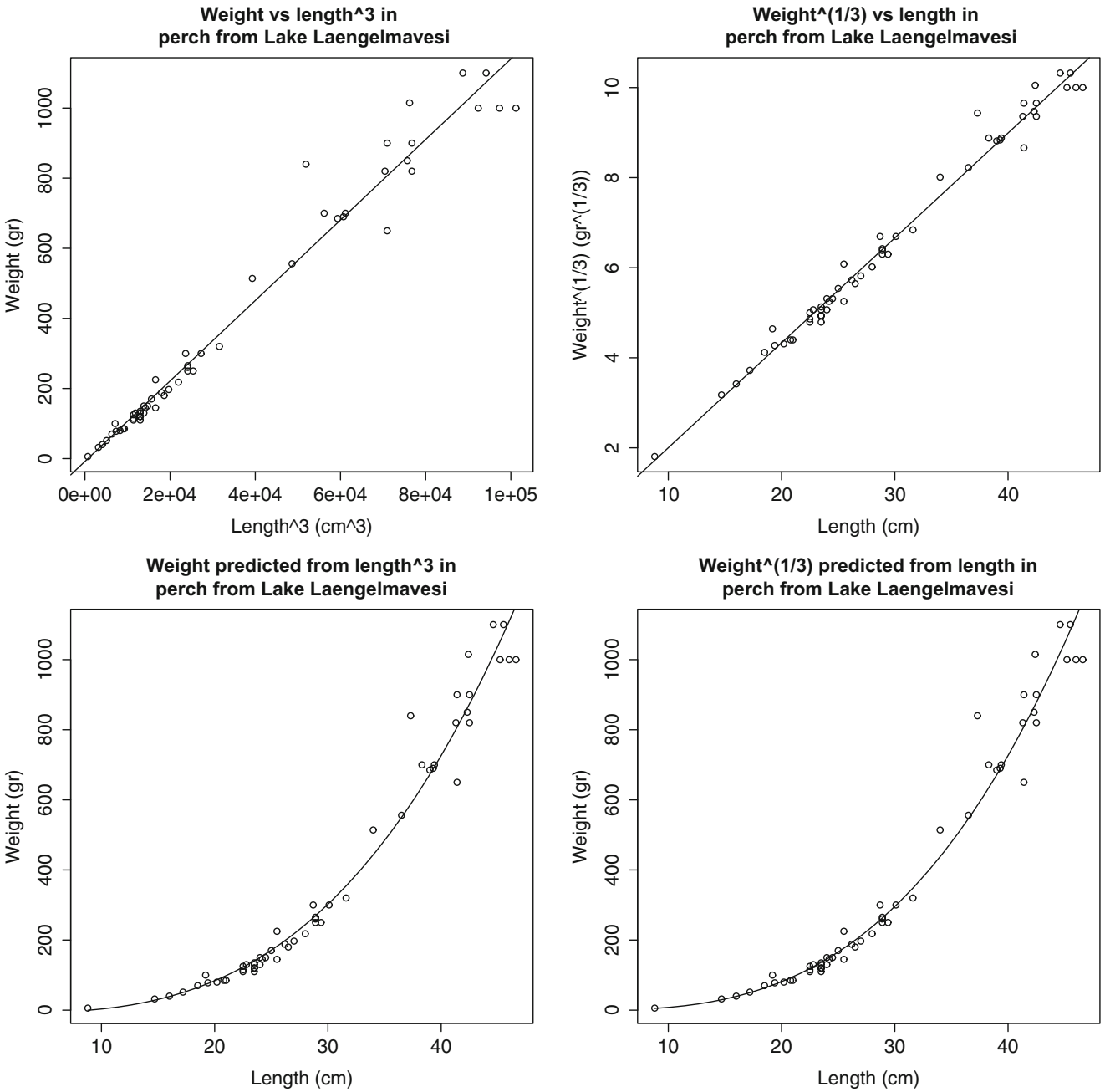

Fig. 13.5 Two variable transformations on the perch dataset. On the *top left*, weight predicted from length cubed; on the *top right*, cube root of weight predicted from length. On the *bottom* corresponding plots transformed to weight-length coordinates (you need to look very closely to see the differences). The non-linear transformation helps significantly

These data points can come from many sources. They may simply be errors. Failures of equipment, transcription errors, someone guessing a value to replace lost data, and so on are some methods that might produce outliers. Another possibility is your understanding of the problem is wrong. If there are some rare effects that are very different than the most common case, you might see outliers. Major scientific discoveries have resulted from investigators taking outliers seriously, and trying to find out what caused them (though you shouldn't see a Nobel prize lurking behind every outlier).

What to do about outliers is even more fraught. The simplest strategy is to find them, then remove them from the data. For low dimensional models, you can do this by plotting data and predictions, then looking for problems. There are other methods, but they are too complicated for us. You should be aware that this strategy can get dangerous fairly quickly, whether you use a simple or a sophisticated method. First, you might find that each time you remove a few problematic data points,

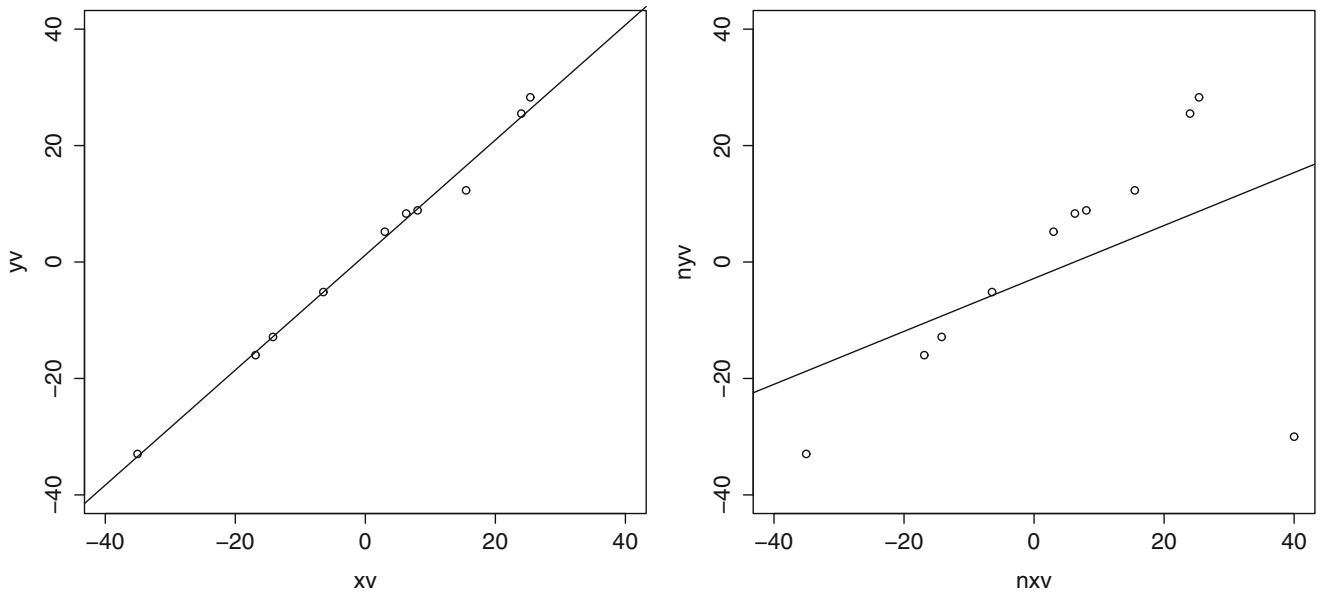

Fig. 13.6 On the *left*, a synthetic dataset with one independent and one explanatory variable, with the regression line plotted. Notice the line is close to the data points, and its predictions seem likely to be reliable. On the *right*, the result of adding a single outlying datapoint to that dataset. The regression line has changed significantly, because the regression line tries to minimize the sum of squared vertical distances between the data points and the line. Because the outlying datapoint is far from the line, the squared vertical distance to this point is enormous. The line has moved to reduce this distance, at the cost of making the other points further from the line

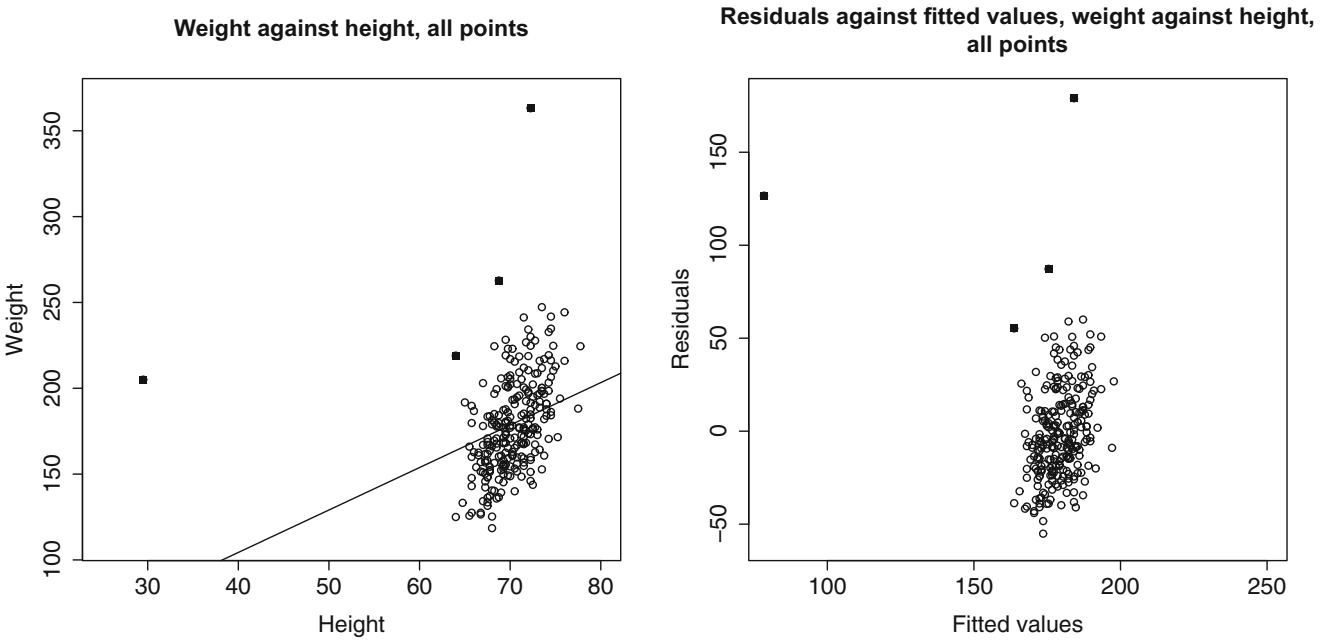

Fig. 13.7 On the *left*, weight regressed against height for the bodyfat dataset. The line doesn't describe the data particularly well, because it has been strongly affected by a few data points (filled-in markers). On the *right*, a scatter plot of the residual against the value predicted by the regression. This doesn't look like noise, which is a sign of trouble

some more data points look strange to you. This process is unlikely to end well. Second, you should be aware that throwing out outliers can *increase* your future prediction error, particularly if they're caused by real effects. An alternative strategy is to build methods that can either discount or model the effects of outliers.

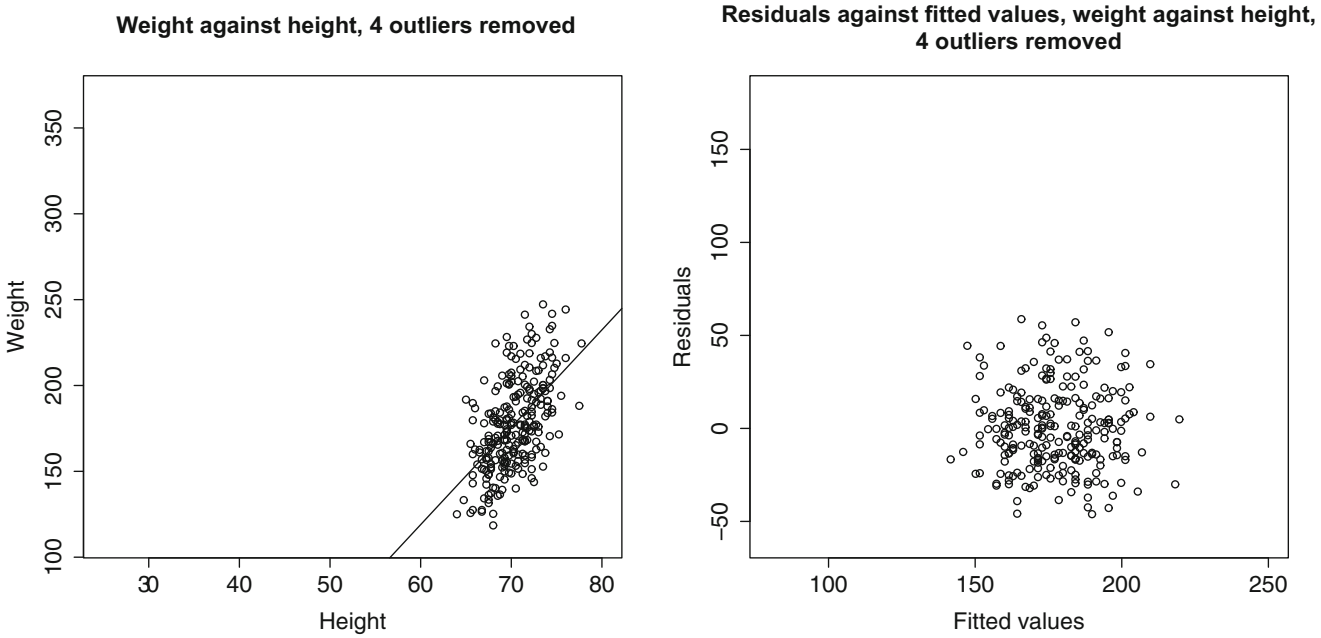

Fig. 13.8 On the *left*, weight regressed against height for the bodyfat dataset. I have now removed the four suspicious looking data points, identified in Fig. 13.7 with filled-in markers; these seemed the most likely to be outliers. On the *right*, a scatter plot of the residual against the value predicted by the regression. Notice that the residual looks like noise. The residual seems to be uncorrelated to the predicted value; the mean of the residual seems to be zero; and the variance of the residual doesn't depend on the predicted value. All these are good signs, consistent with our model, and suggest the regression will yield good predictions

Remember this: *Outliers can affect linear regressions significantly. Usually, if you can plot the regression, you can look for outliers by eyeballing the plot. Other methods exist, but are beyond the scope of this text.*

13.4.3 Functions of One Explanatory Variable

Imagine we have only one measurement to form explanatory variables. For example, in the perch data of Fig. 13.1, we have only the length of the fish. If we evaluate functions of that measurement, and insert them into the vector of explanatory variables, the resulting regression is still easy to plot. It may also offer better predictions. The fitted line of Fig. 13.1 looks quite good, but the data points look as though they might be willing to follow a curve. We can get a curve quite easily. Our current model gives the weight as a linear function of the length with a noise term (which we wrote $y_i = \beta_1 x_i + \beta_0 + \xi_i$). But we could expand this model to incorporate other functions of the length. In fact, it's quite surprising that the weight of a fish should be predicted by its length. If the fish doubled in each direction, say, its weight should go up by a factor of eight. The success of our regression suggests that fish do not just scale in each direction as they grow. But we might try the model $y_i = \beta_2 x_i^2 + \beta_1 x_i + \beta_0 + \xi_i$. This is easy to do. The i'th row of the matrix $\mathcal{X}$ currently looks like $[x_i, 1]$. We build a new matrix $\mathcal{X}^{(b)}$, where the i'th row is $[x_i^2, x_i, 1]$, and proceed as before. This gets us a new model. The nice thing about this model is that it is easy to plot—our predicted weight is still a function of the length, it's just not a linear function of the length. Several such models are plotted in Fig. 13.9.

You should notice that it can be quite easy to add a lot of functions like this (in the case of the fish, I tried x_i^3 as well). However, it's hard to decide whether the regression has actually gotten better. The least-squares error *on the training data* will *never* go up when you add new explanatory variables, so the R^2 will *never* get worse. This is easy to see, because you could always use a coefficient of zero with the new variables and get back the previous regression. However, the models that you choose are likely to produce worse and worse predictions as you add explanatory variables. Knowing when to stop can be tough, though it's sometimes obvious that the model is untrustworthy (Fig. 13.9).

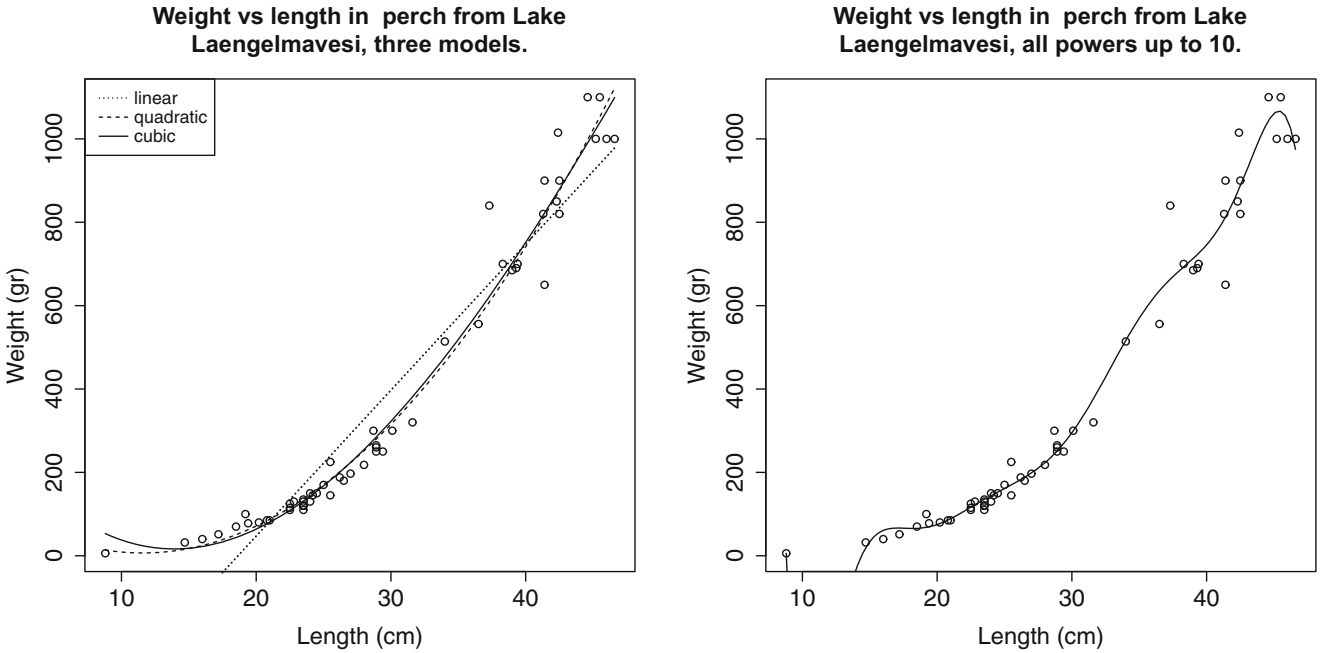

Fig. 13.9 On the *left*, several different models predicting fish weight from length. The line uses the explanatory variables 1 and x_i; and the curves use other monomials in x_i as well, as shown by the legend. This allows the models to predict curves that lie closer to the data. It is important to understand that, while you can make a curve go closer to the data by inserting monomials, that doesn't mean you necessarily have a better model. On the *right*, I have used monomials up to x_i^{10}. This curve lies very much closer to the data points than any on the other side, at the cost of some very odd looking wiggles in between data points (look at small lengths; the model goes quite strongly negative there, but I can't bring myself to change the axes and show predictions that are obvious nonsense). I can't think of any reason that these structures would come from true properties of fish, and it would be hard to trust predictions from this model

Remember this: *If you have only one measurement, you can construct a high dimensional* $\mathbf{x}$ *by using functions of that measurement. This produces a regression that has many explanatory variables, but is still easy to plot. Knowing when to stop is hard. An understanding of the problem is helpful.*

13.4.4 Regularizing Linear Regressions

When we have many explanatory variables, some might be significantly correlated. This means that we can predict, quite accurately, the value of one explanatory variable using the values of the other variables. This means there must be a vector $\mathbf{w}$ so that $\mathcal{X}\mathbf{w}$ is small (exercises). In turn, that $\mathbf{w}^T\mathcal{X}^T\mathcal{X}\mathbf{w}$ must be small, so that $\mathcal{X}^T\mathcal{X}$ has some small eigenvalues. These small eigenvalues lead to bad predictions, as follows. The vector $\mathbf{w}$ has the property that $\mathcal{X}^T\mathcal{X}\mathbf{w}$ is small. This means that $\mathcal{X}^T\mathcal{X}(\hat{\beta} + \mathbf{w})$ is not much different from $\mathcal{X}^T\mathcal{X}\hat{\beta}$ (equivalently, the matrix can turn large vectors into small ones). All this means that $(\mathcal{X}^T\mathcal{X})^{-1}$ will turn some small vectors into big ones. A small change in $\mathcal{X}^T\mathbf{Y}$ can lead to a large change in the estimate of $\hat{\beta}$.

This is a problem, because we can expect that different samples from the same data will have somewhat different values of $\mathcal{X}^T\mathbf{Y}$. For example, imagine the person recording fish measurements in Lake Laengelmavesi recorded a different set of fish; we expect changes in $\mathcal{X}$ and $\mathbf{Y}$. But, if $\mathcal{X}^T\mathcal{X}$ has small eigenvalues, these changes could produce large changes in our model (Figs. 13.10 and 13.11).

The problem is relatively easy to control. When there are small eigenvalues in $\mathcal{X}^T\mathcal{X}$, we expect that $\hat{\beta}$ will be large (because we can add components in the direction of $\mathbf{w}$ without changing all that much), and the largest components in $\hat{\beta}$ might be very inaccurately estimated. If we are trying to predict new y values, we expect that large components in $\hat{\beta}$ turn into large errors in prediction (exercises).

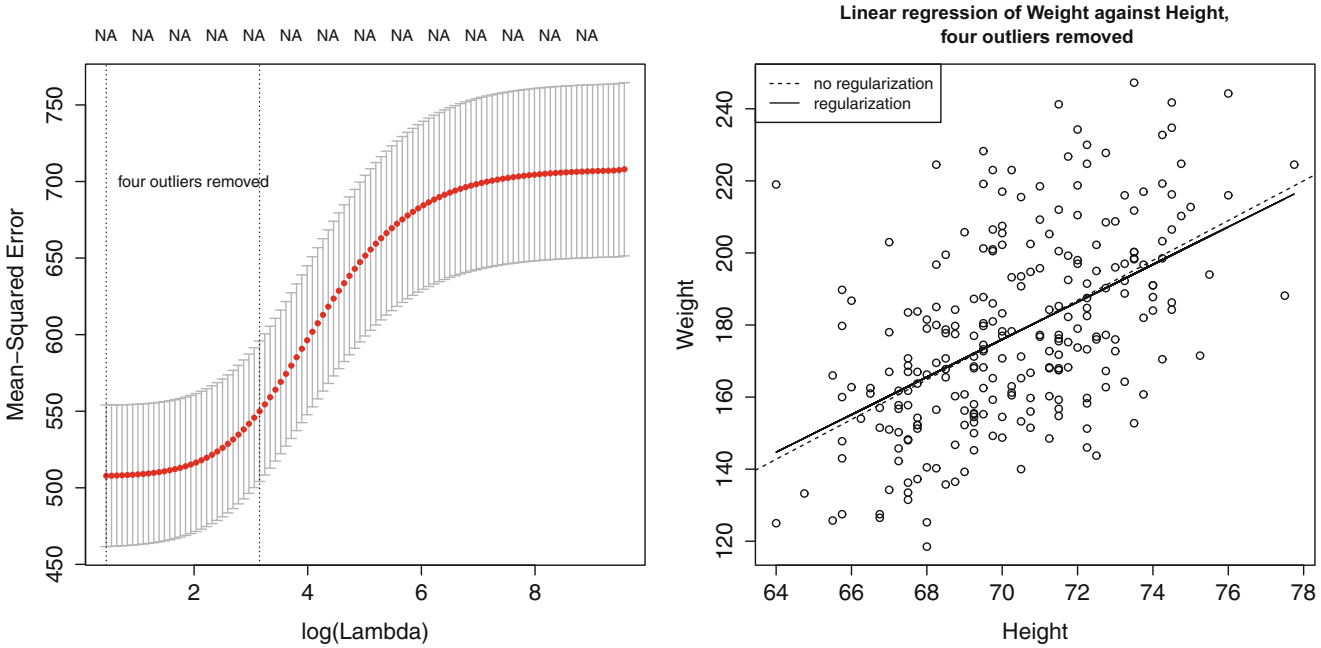

Fig. 13.10 On the *left*, cross-validated error estimated for different choices of regularization constant for a linear regression of weight against height for the bodyfat dataset, with four outliers removed. The horizontal axis is log regression constant; the vertical is cross-validated error. The mean of the error is shown as a spot, with vertical error bars. The vertical lines show a range of reasonable choices of regularization constant (*left* yields the lowest observed error, *right* the error whose mean is within one standard error of the minimum). On the *right*, two regression lines on a scatter plot of this dataset; one is the line computed without regularization, the other is obtained using the regularization parameter that yields the lowest observed error. In this case, the regularizer doesn't change the line much, but may produce improved values on new data (notice how the cross-validated error is fairly flat with low values of the regularization constant)

An important and useful way to suppress these errors is to try to find a $\hat{\beta}$ that isn't large, and also gives a low error. We can do this by regularizing, using the same trick we saw in the case of classification. Instead of choosing the value of β that minimizes

$$\left(\frac{1}{N}\right)(\mathbf{y}-\mathcal{X}\beta)^T(\mathbf{y}-\mathcal{X}\beta)$$

we minimize

$$\left(\frac{1}{N}\right)(\mathbf{y}-\mathcal{X}\beta)^T(\mathbf{y}-\mathcal{X}\beta)+\lambda\beta^T\beta$$

$$\text{Error} + \text{Regularizer}$$

Here $\lambda > 0$ is a constant (the **regularization weight**, though it's pretty widely known as λ) that weights the two requirements (small error; small $\hat{\beta}$) relative to one another. Notice also that dividing the total error by the number of data points means that our choice of λ shouldn't be affected by changes in the size of the data set.

Regularization helps deal with the small eigenvalue, because to solve for β we must solve the equation

$$\left[\left(\frac{1}{N}\right)\mathcal{X}^T\mathcal{X}+\lambda\mathcal{I}\right]\hat{\beta}=\left(\frac{1}{N}\right)\mathcal{X}^T\mathbf{y}$$

(obtained by differentiating with respect to β and setting to zero) and the smallest eigenvalue of the matrix $\left(\left(\frac{1}{N}\right)(\mathcal{X}^T\mathcal{X}+\lambda\mathcal{I})\right.$ will be at least λ (exercises). Penalizing a regression with the size of β in this way is sometimes known as **ridge regression**. The value of λ that is most helpful depends on the dataset. Typically, one sets up a range of values, then searches, using cross-validation to estimate the error.

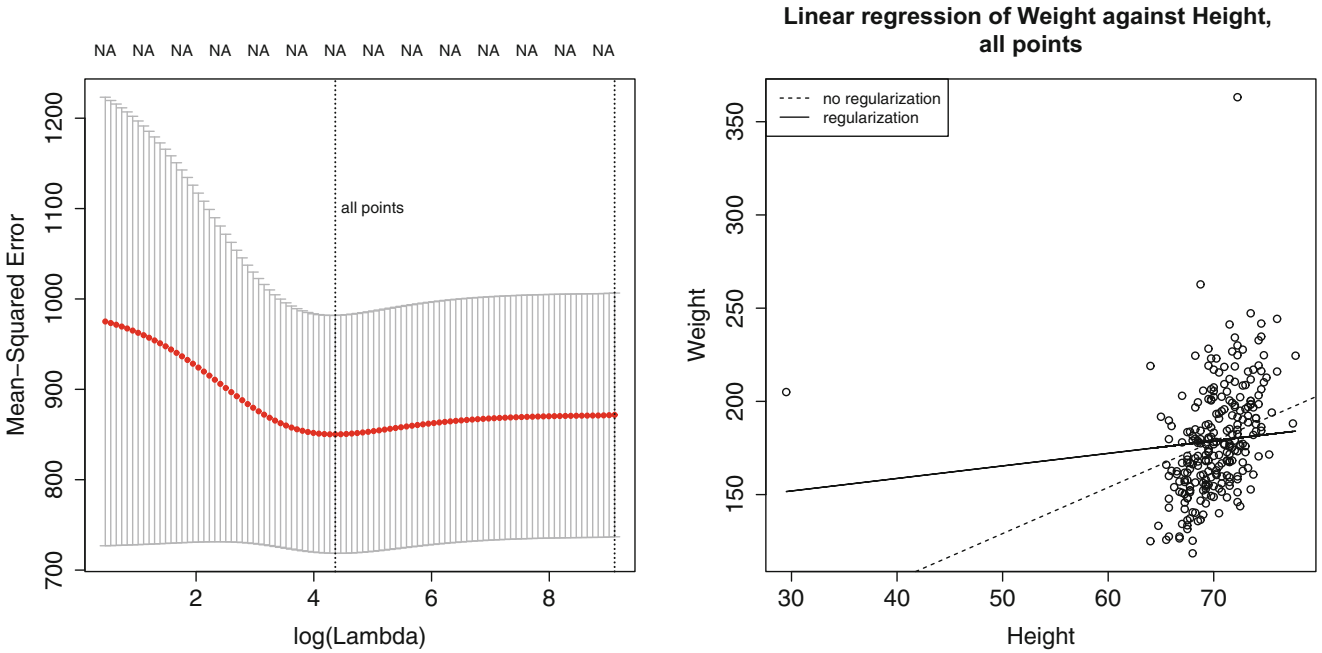

Fig. 13.11 Regularization doesn't make outliers go away. On the *left*, cross-validated error estimated for different choices of regularization constant for a linear regression of weight against height for the bodyfat dataset, with all points. The horizontal axis is log regression constant; the vertical is cross-validated error. The mean of the error is shown as a spot, with vertical error bars. The vertical lines show a range of reasonable choices of regularization constant (*left* yields the lowest observed error, *right* the error whose mean is within one standard error of the minimum). On the *right*, two regression lines on a scatter plot of this dataset; one is the line computed without regularization, the other is obtained using the regularization parameter that yields the lowest observed error. In this case, the regularizer doesn't change the line much, but may produce improved values on new data (notice how the cross-validated error is fairly flat with low values of the regularization constant)

Worked example 13.1 (Predicting the Weight of a Fish with Regularized Linear Regression) We have already seen how to predict the weight of a fish using different powers of its length (Sects. 13.4.1 and 13.4.3; Fig. 13.9). Section 13.4.3 showed that using too many powers would likely lead to poor predictions on test data. Show that regularization can be used to control this problem.

Solution The main point of this example is how useful good statistical software can be, and to draw your attention to an excellent package. The package I use for regressions, `glmnet`, will choose a good range of regularization weights (λ's) and compute estimates of the mean and standard deviation of the squared cross-validated error for various values in that range. It then prepares a nice plot of this information, which makes the impact of the regularization clear. I've shown such a plot in Fig. 13.12. In this problem, quite a large value of the regularization constant produces the best result. I've also show a plot of the predictions made, with the coefficients of each power of length used in the regression for the best value of the regularization constant. You should notice that the coefficient of length and its square are fairly high, there's a small value of the coefficient for the cube of length, and for higher powers the coefficients are pretty tiny. If you're careful, you'll check that the coefficients are small compared to the scale of the numbers (because the 10'th power of 20, say, is big). The curve has no wiggles in it, because these coefficients mean that high powers make almost no contribution to its shape.

We choose λ in the same way we used for classification; split the training set into a training piece and a validation piece, train for different values of λ, and test the resulting regressions on the validation piece. The error is a random variable, random because of the random split. It is a fair model of the error that would occur on a randomly chosen test example

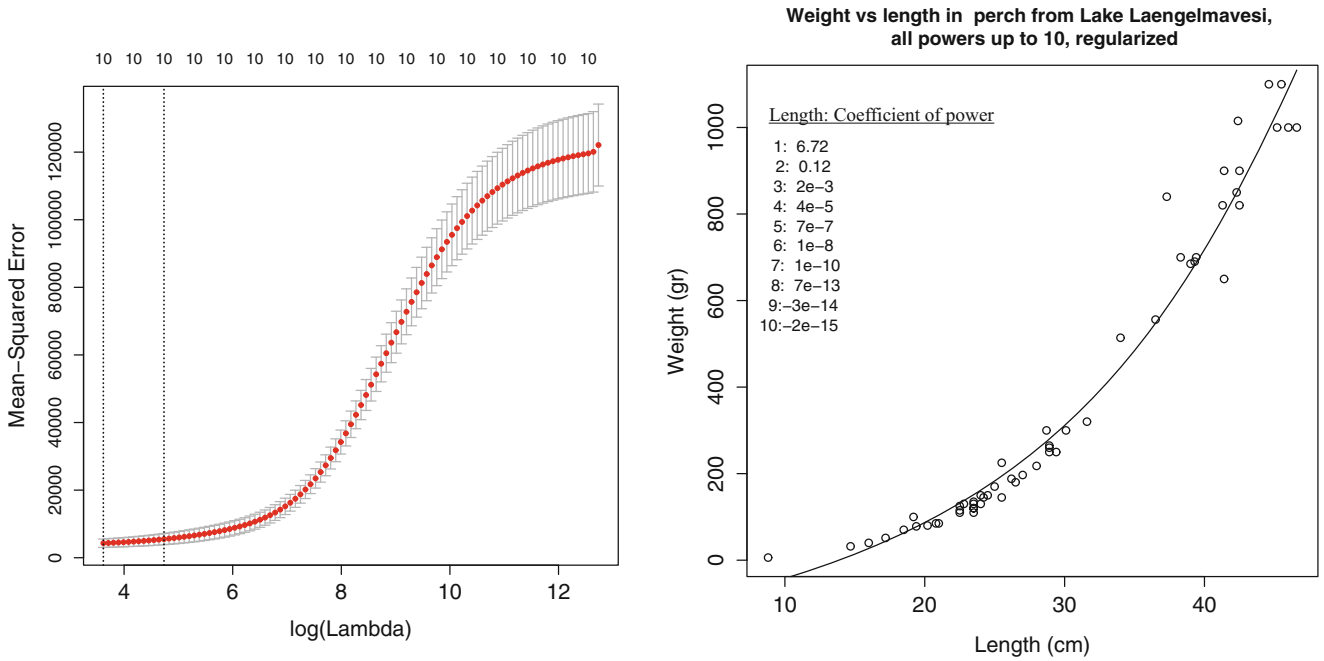

Fig. 13.12 Regularization can be a significant help when there are many predictors. On the *left*, the glmnet plot of cross-validated prediction error against log regularization coefficient for the perch data of Fig. 13.9. The set of independent variables includes all powers of length up to 10 (as in the wiggly graph on the right of Fig. 13.9). Notice that the regularization coefficient that yields the smallest error is quite large (the horizontal axis is on a logarithmic scale). On the *right*, the curve of predicted values. The cross-validated error chooses a regularization constant that discourages wiggles; inspecting the coefficients, shown in the inset, shows that high powers of length are firmly suppressed

(assuming that the training set is "like" the test set, in a way that I do not wish to make precise yet). We could use multiple splits, and average over the splits. Doing so yields both an average error for a value of λ and an estimate of the standard deviation of error.

Statistical software will do all the work for you. I used the `glmnet` package in R; this package is available in Matlab, too. There are likely other such packages. Figure 13.10 shows an example, for weight regressed against height. Notice the regularization doesn't change the model (plotted in the figure) all that much. For each value of λ (horizontal axis), the method has computed the mean error and standard deviation of error using cross-validation splits, and displays these with error bars. Notice that $\lambda = 0$ yields poorer predictions than a larger value; large $\hat{\beta}$ really are unreliable. Notice that now there is now no λ that yields the smallest validation error, because the value of error depends on the random splits used in cross-validation. A reasonable choice of λ lies between the one that yields the smallest error encountered (one vertical line in the plot) and the largest value whose mean error is within one standard deviation of the minimum (the other vertical line in the plot).

All this is quite similar to regularizing a classification problem. We started with a cost function that evaluated the errors caused by a choice of β, then added a term that penalized β for being "large". This term is the squared length of β, as a vector. It is sometimes known as the L_2 **norm** of the vector.

Remember this: *The performance of a regression can be improved by regularizing, particularly if some explanatory variables are correlated. The procedure is similar to that used for classification.*

13.5 Exploiting Your Neighbors for Regression

Nearest neighbors can clearly predict a number for a query example—you find the closest training example, and report its number. This would be one way to use nearest neighbors for regression, but it isn't terribly effective. One important difficulty is that the regression prediction is piecewise constant (Fig. 13.13). If there is an immense amount of data, this may not present major problems, because the steps in the prediction will be small and close together. But it's not generally an effective use of data.

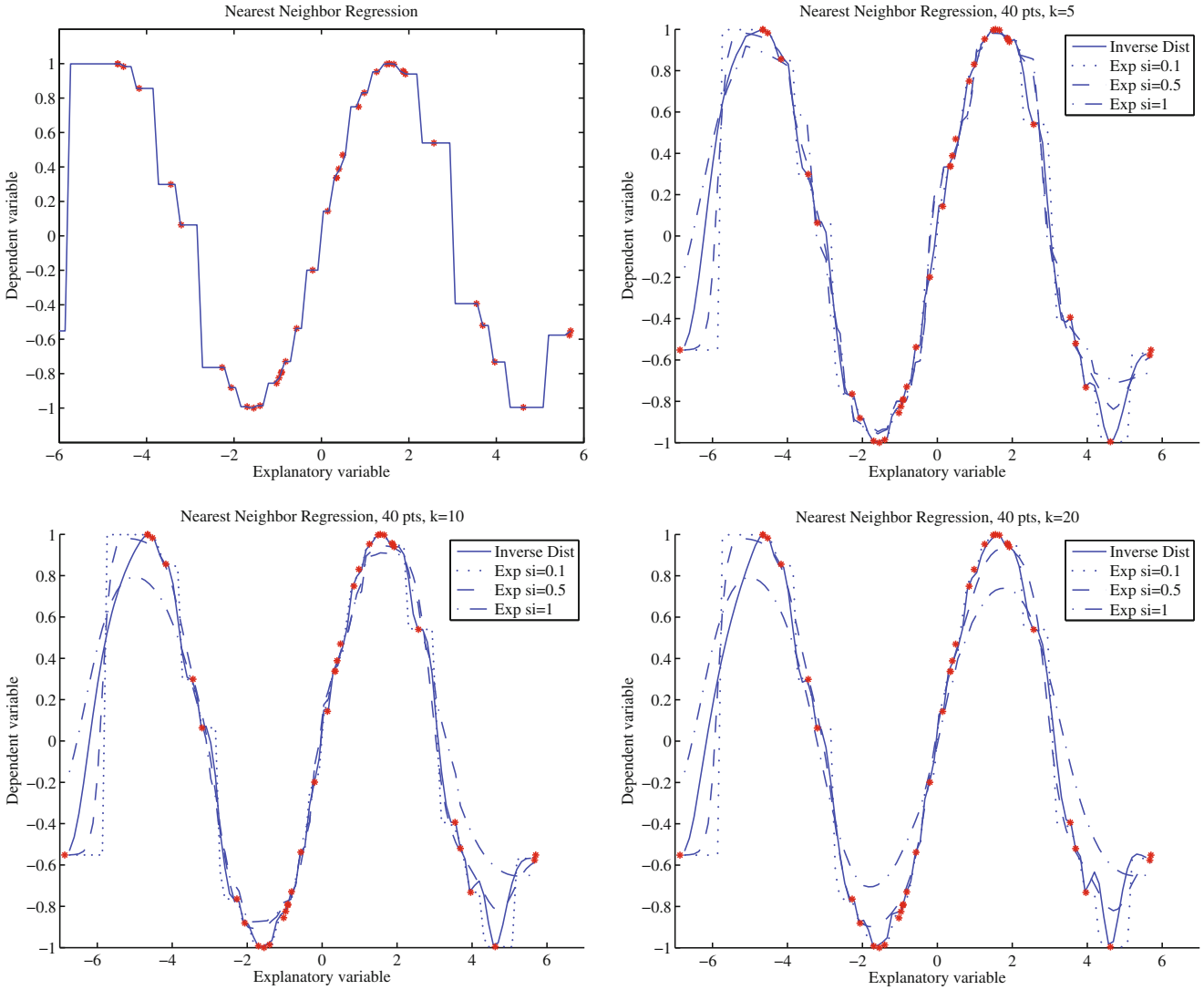

Fig. 13.13 Different forms of nearest neighbors regression, predicting y from a one-dimensional x, using a total of 40 training points. *Top left:* reporting the nearest neighbor leads to a piecewise constant function. *Top right:* improvements are available by forming a weighted average of the five nearest neighbors, using inverse distance weighting or exponential weighting with three different scales. Notice if the scale is small, then the regression looks a lot like nearest neighbors, and if it is too large, all the weights in the average are nearly the same (which leads to a piecewise constant structure in the regression). *Bottom left* and *bottom right* show that using more neighbors leads to a smoother regression

A more effective strategy is to find several nearby training examples, and use them to produce an estimate. This approach can produce very good regression estimates, because every prediction is made by training examples that are near to the query example. However, producing a regression estimate is expensive, because for every query one must find the nearby training examples.

Write $\mathbf{x}$ for the query point, and assume that we have already collected the N nearest neighbors, which we write $\mathbf{x}_i$. Write y_i for the value of the dependent variable for the i'th of these points. Notice that some of these neighbors could be quite far from the query point. We don't want distant points to make as much contribution to the model as nearby points. This suggests forming a weighted average of the predictions of each point. Write w_i for the weight at the i'th point. Then the estimate is

$$y_{pred} = \frac{\sum_i w_i y_i}{\sum_i w_i}.$$

A variety of weightings are reasonable choices. Write $d_i = \|(\mathbf{x} - \mathbf{x}_i)\|$ for the distance between the query point and the i'th nearest neighbor. Then inverse distance weighting uses $w_i = 1/d_i$. Alternatively, we could use an exponential function to strongly weight down more distant points, using

$$w_i = \exp\left(\frac{-d_i^2}{2\sigma^2}\right).$$

We will need to choose a scale σ, which can be done by cross-validation. Hold out some examples, make predictions at the held out examples using a variety of different scales, and choose the scale that gives the best held-out error. Alternatively, if there are enough nearest neighbors, we could form a distance weighted linear regression, then predict the value at the query point from that regression.

Each of these strategies presents some difficulties when $\mathbf{x}$ has high dimension. In that case, it is usual that the nearest neighbor is a lot closer than the second nearest neighbor. If this happens, then each of these weighted averages will boil down to evaluating the dependent variable at the nearest neighbor (because all the others will have very small weight in the average).

Remember this: *Nearest neighbors can be used for regression. In the simplest approach, you find the nearest neighbor to your feature vector, and take that neighbor's number as your prediction. More complex approaches smooth predictions over multiple neighbors.*

13.5.1 Using Your Neighbors to Predict More than a Number

Linear regression takes some features and predicts a number. But in practice, one often wants to predict something more complex than a number. For example, I might want to predict a parse tree (which has combinatorial structure) from a sentence (the explanatory variables). As another example, I might want to predict a map of the shadows in an image (which has spatial structure) against an image (the explanatory variables). As yet another example, I might want to predict which direction to move the controls on a radio-controlled helicopter (which have to be moved together) against a path plan and the current state of the helicopter (the explanatory variables).

Looking at neighbors is a very good way to solve such problems. The general strategy is relatively simple. We find a large collection of pairs of training data. Write $\mathbf{x}_i$ for the explanatory variables for the i'th example, and $\mathbf{y}_i$ for the dependent variable in the i'th example. This dependent variable could be anything—it doesn't need to be a single number. It might be a tree, or a shadow map, or a word, or anything at all. I wrote it as a vector because I needed to choose some notation.

In the simplest, and most general, approach, we obtain a prediction for a new set of explanatory variables $\mathbf{x}$ by (a) finding the nearest neighbor and then (b) producing the dependent variable for that neighbor. We might vary the strategy slightly by using an approximate nearest neighbor. If the dependent variables have enough structure that it is possible to summarize a collection of different dependent variables, then we might recover the k nearest neighbors and summarize their dependent variables. How we summarize rather depends on the dependent variables. For example, it is a bit difficult to imagine the average of a set of trees, but quite straightforward to average images. If the dependent variable was a word, we might not be able to average words, but we can vote and choose the most popular word. If the dependent variable is a vector, we can compute either distance weighted averages or a distance weighted linear regression.

Remember this: *Nearest neighbors can be used to predict more than numbers.*

13.6 You Should

13.6.1 Remember These Definitions

13.6.2 Remember These Terms

13.6.3 Remember These Facts

13.6.4 Remember These Procedures

Appendix: Data

Table 13.1 A table showing the amount of hormone remaining and the time in service for devices from lot A, lot B and lot C

Batch A		Batch B		Batch C	
Amount of hormone	Time in service	Amount of hormone	Time in service	Amount of hormone	Time in service
25.8	99	16.3	376	28.8	119
20.5	152	11.6	385	22.0	188
14.3	293	11.8	402	29.7	115
23.2	155	32.5	29	28.9	88
20.6	196	32.0	76	32.8	58
31.1	53	18.0	296	32.5	49
20.9	184	24.1	151	25.4	150
20.9	171	26.5	177	31.7	107
30.4	52	25.8	209	28.5	125

The numbering is arbitrary (i.e. there's no relationship between device 3 in lot A and device 3 in lot B). We expect that the amount of hormone goes down as the device spends more time in service, so cannot compare batches just by comparing numbers. This data appeared in "Computer-Intensive methods in statistical regression", B. Efron, SIAM Review, 1988, and is used for Fig. 13.3

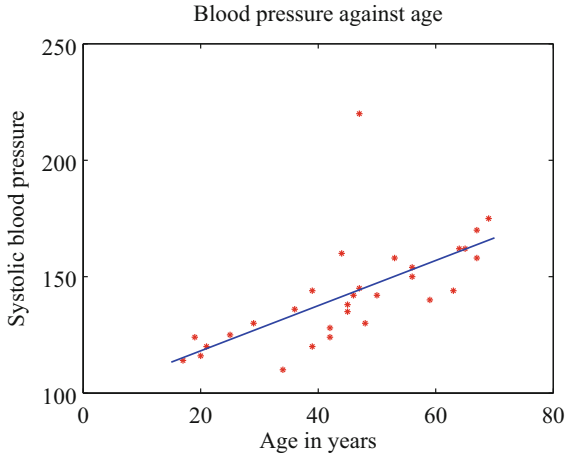

Fig. 13.14 A regression of blood pressure against age, for 30 data points

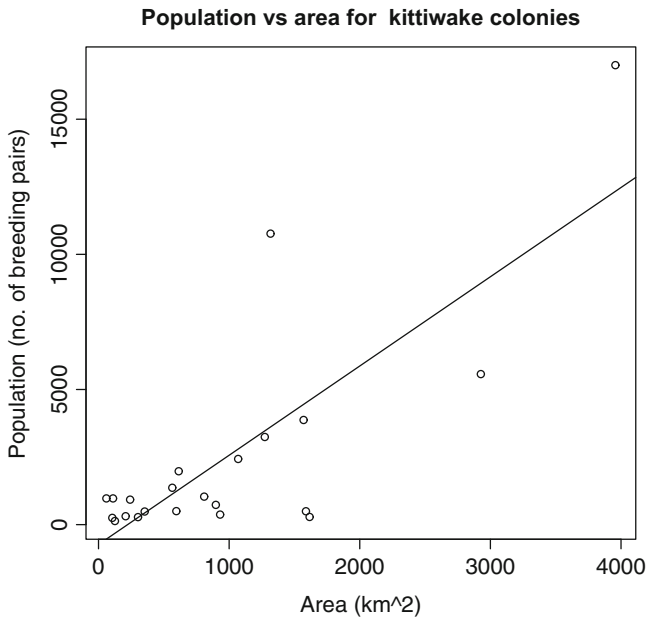

Fig. 13.15 A regression of the number of breeding pairs of kittiwakes against the area of an island, for 22 data points

Problems

13.1 Figure 13.14 shows a linear regression of systolic blood pressure against age. There are 30 data points.

(a) Write $e_i = y_i - \mathbf{x}_i^T \beta$ for the residual. What is the mean $(\{e\})$ for this regression?
(b) For this regression, var $(\{y\}) = 509$ and the R^2 is 0.4324. What is var $(\{e\})$ for this regression?
(c) How well does the regression explain the data?
(d) What could you do to produce better predictions of blood pressure (without actually measuring blood pressure)?

13.2 At http://www.statsci.org/data/general/kittiwak.html, you can find a dataset collected by D.K. Cairns in 1988 measuring the area available for a seabird (black-legged kittiwake) colony and the number of breeding pairs for a variety of different colonies. Figure 13.15 shows a linear regression of the number of breeding pairs against the area. There are 22 data points.

(a) Write $e_i = y_i - \mathbf{x}_i^T \beta$ for the residual. What is the mean $(\{e\})$ for this regression?
(b) For this regression, var $(\{y\}) = 16{,}491{,}357$ and the R^2 is 0.62. What is var $(\{e\})$ for this regression?
(c) How well does the regression explain the data? If you had a large island, to what extent would you trust the prediction for the number of kittiwakes produced by this regression? If you had a small island, would you trust the answer more?

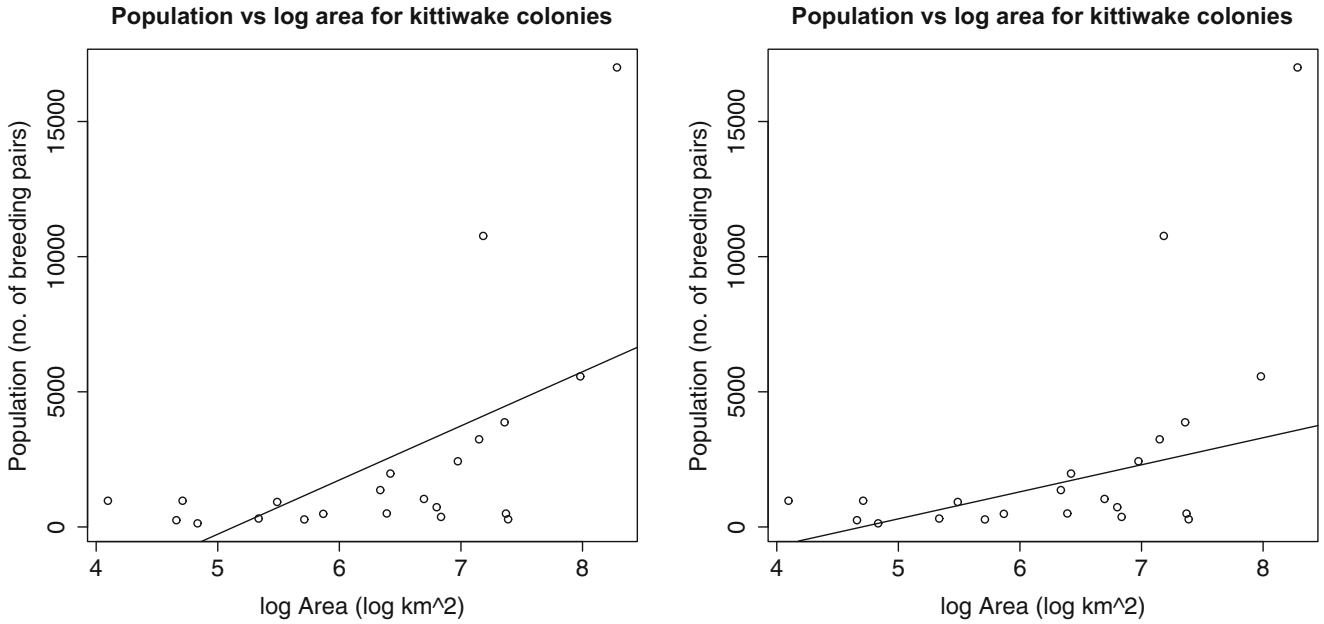

Fig. 13.16 *Left:* A regression of the number of breeding pairs of kittiwakes against the log of area of an island, for 22 data points. *Right:* A regression of the number of breeding pairs of kittiwakes against the log of area of an island, for 22 data points, using a method that ignores two likely outliers

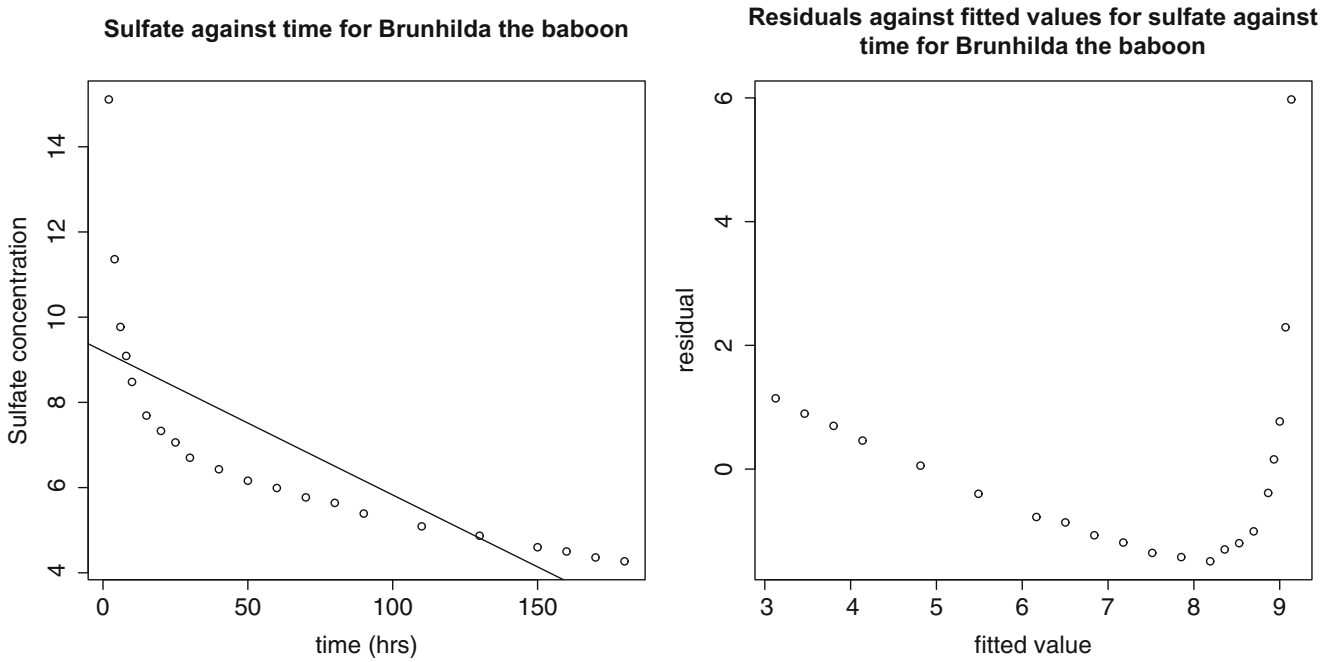

Fig. 13.17 *Left:* A regression of the concentration of sulfate in the blood of Brunhilda the baboon against time. *Right:* For this regression, a plot of residual against fitted value

13.3 At http://www.statsci.org/data/general/kittiwak.html, you can find a dataset collected by D.K. Cairns in 1988 measuring the area available for a seabird (black-legged kittiwake) colony and the number of breeding pairs for a variety of different colonies. Figure 13.16 shows a linear regression of the number of breeding pairs against the log of area. There are 22 data points.

(a) Write $e_i = y_i - \mathbf{x}_i^T \beta$ for the residual. What is the mean $(\{e\})$ for this regression?

(b) For this regression, $\mathsf{var}(\{y\}) = 16{,}491{,}357$ and the R^2 is 0.31. What is $\mathsf{var}(\{e\})$ for this regression?
(c) How well does the regression explain the data? If you had a large island, to what extent would you trust the prediction for the number of kittiwakes produced by this regression? If you had a small island, would you trust the answer more? Why?
(d) Figure 13.16 shows the result of a linear regression that ignores two likely outliers. Would you trust the predictions of this regression more? Why?

13.4 At http://www.statsci.org/data/general/brunhild.html, you will find a dataset that measures the concentration of a sulfate in the blood of a baboon named Brunhilda as a function of time. Figure 13.17 plots this data, with a linear regression of the concentration against time. I have shown the data, and also a plot of the residual against the predicted value. The regression appears to be unsuccessful.

(a) What suggests the regression has problems?
(b) What is the cause of the problem, and why?
(c) What could you do to improve the problems?

13.5 Assume we have a dataset where $\mathbf{Y} = \mathcal{X}\beta + \xi$, for some unknown β and ξ. The term ξ is a normal random variable with zero mean, and covariance $\sigma^2\mathcal{I}$ (i.e. this data really does follow our model).

(a) Write $\hat{\beta}$ for the estimate of β recovered by least squares, and $\hat{\mathbf{Y}}$ for the values predicted by our model for the training data points. Show that

$$\hat{\mathbf{Y}} = \mathcal{X}\left(\mathcal{X}^T\mathcal{X}\right)^{-1}\mathcal{X}^T\mathbf{Y}$$

(b) Show that

$$\mathbb{E}[\hat{y}_i - y_i] = 0$$

for each training data point y_i, where the expectation is over the probability distribution of ξ.
(c) Show that

$$\mathbb{E}\left[(\hat{\beta} - \beta)\right] = 0$$

where the expectation is over the probability distribution of ξ.

13.6 In this exercise, I will show that the prediction process of Chap. 2 (see page 42) is a linear regression with two independent variables. Assume we have N data items which are 2-vectors $(x_1, y_1), \ldots, (x_N, y_N)$, where $N > 1$. These could be obtained, for example, by extracting components from larger vectors. As usual, we will write $\hat{x}_i$ for x_i in normalized coordinates, and so on. The correlation coefficient is r (this is an important, traditional notation).

(a) Assume that we have an x_o, for which we wish to predict a y value. Show that the value of the prediction obtained using the method of page 43 is

$$\begin{aligned} y_{\text{pred}} &= \frac{\mathsf{std}(y)}{\mathsf{std}(x)} r(x_0 - \mathsf{mean}(\{x\})) + \mathsf{mean}(\{y\}) \\ &= \left(\frac{\mathsf{std}(y)}{\mathsf{std}(x)} r\right) x_0 + \left(\mathsf{mean}(\{y\}) - \frac{\mathsf{std}(x)}{\mathsf{std}(y)}\mathsf{mean}(\{x\})\right). \end{aligned}$$

(b) Show that

$$\begin{aligned} r &= \frac{\mathsf{mean}(\{(x - \mathsf{mean}(\{x\}))(y - \mathsf{mean}(\{y\}))\})}{\mathsf{std}(x)\mathsf{std}(y)} \\ &= \frac{\mathsf{mean}(\{xy\}) - \mathsf{mean}(\{x\})\mathsf{mean}(\{y\})}{\mathsf{std}(x)\mathsf{std}(y)}. \end{aligned}$$

(c) Now write

$$\mathcal{X} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} \text{ and } \mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}.$$

The coefficients of the linear regression will be $\hat{\beta}$, where $\mathcal{X}^T\mathcal{X}\hat{\beta} = \mathcal{X}^T\mathbf{Y}$. Show that

$$\begin{aligned}\mathcal{X}^T\mathcal{X} &= N\begin{pmatrix} \text{mean}\left(\{x^2\}\right) & \text{mean}\left(\{x\}\right) \\ \text{mean}\left(\{x\}\right) & 1 \end{pmatrix} \\ &= N\begin{pmatrix} \text{std}\left(x\right)^2 + \text{mean}\left(\{x\}\right)^2 & \text{mean}\left(\{x\}\right) \\ \text{mean}\left(\{x\}\right) & 1 \end{pmatrix}\end{aligned}$$

(d) Now show that

$$\begin{aligned}\mathcal{X}^T\mathbf{Y} &= N\begin{pmatrix} \text{mean}\left(\{xy\}\right) \\ \text{mean}\left(\{y\}\right) \end{pmatrix} \\ &= N\begin{pmatrix} \text{std}\left(x\right)\text{std}\left(y\right)r + \text{mean}\left(\{x\}\right)\text{mean}\left(\{y\}\right) \\ \text{mean}\left(\{y\}\right) \end{pmatrix}.\end{aligned}$$

(e) Now show that

$$\left(\mathcal{X}^T\mathcal{X}\right)^{-1} = \frac{1}{N}\frac{1}{\text{std}\left(x\right)^2}\begin{pmatrix} 1 & -\text{mean}\left(\{x\}\right) \\ -\text{mean}\left(\{x\}\right) & \text{std}\left(x\right)^2 + \text{mean}\left(\{x\}\right)^2 \end{pmatrix}$$

(f) Now (finally!) show that if $\hat{\beta}$ is the solution to $\mathcal{X}^T\mathcal{X}\hat{\beta} - \mathcal{X}^T\mathbf{Y} = 0$, then

$$\hat{\beta} = \begin{pmatrix} r\frac{\text{std}(y)}{\text{std}(x)} \\ \text{mean}\left(\{y\}\right) - \left(r\frac{\text{std}(y)}{\text{std}(x)}\right)\text{mean}\left(\{x\}\right) \end{pmatrix}$$

and use this to argue that the process of page 42 is a linear regression with two independent variables.

13.7 This exercise investigates the effect of correlation on a regression. Assume we have N data items, $(\mathbf{x}_i, y_i)$. We will investigate what happens when the data have the property that the first component is relatively accurately predicted by the other components. Write x_{i1} for the first component of $\mathbf{x}_i$, and $\mathbf{x}_{i,\hat{1}}$ for the vector obtained by deleting the first component of $\mathbf{x}_i$. Choose $\mathbf{u}$ to predict the first component of the data from the rest *with minimum error*, so that $x_{i1} = \mathbf{x}_{i\hat{1}}^T\mathbf{u} + w_i$. The error of prediction is w_i. Write $\mathbf{w}$ for the vector of errors (i.e. the i'th component of $\mathbf{w}$ is w_i). Because $\mathbf{w}^T\mathbf{w}$ is minimized by choice of $\mathbf{u}$, we have $\mathbf{w}^T\mathbf{1} = 0$ (i.e. the average of the w_i's is zero). Assume that these predictions are very good, so that there is some small positive number ϵ so that $\mathbf{w}^T\mathbf{w} \leq \epsilon$.

(a) Write $\mathbf{a} = [-1, \mathbf{u}]^T$. Show that

$$\mathbf{a}^T\mathcal{X}^T\mathcal{X}\mathbf{a} \leq \epsilon.$$

(b) Now show that the smallest eigenvalue of $\mathcal{X}^T\mathcal{X}$ is less than or equal to ϵ.

(c) Assume that $\hat{\beta}$ is the solution to $\mathcal{X}^T\mathcal{X}\hat{\beta} = \mathcal{X}^T\mathbf{Y}$. Show that there is a unit vector $\mathbf{v}$ such that

$$(\mathcal{X}^T\mathbf{Y} - \mathcal{X}^T\mathcal{X}(\hat{\beta} + \mathbf{v}))^T(\mathcal{X}^T\mathbf{Y} - \mathcal{X}^T\mathcal{X}(\hat{\beta} + \mathbf{v}))$$

is bounded above by

$$\epsilon^2$$

(d) Use the last sub exercises to explain why correlated data will lead to a poor estimate of $\hat{\beta}$.

13.8 This exercise explores the effect of regularization on a regression. Assume we have N data items, $(\mathbf{x}_i, y_i)$. We will investigate what happens when the data have the property that the first component is relatively accurately predicted by the other components. Write x_{i1} for the first component of $\mathbf{x}_i$, and $\mathbf{x}_{i,\hat{1}}$ for the vector obtained by deleting the first component of $\mathbf{x}_i$. Choose $\mathbf{u}$ to predict the first component of the data from the rest *with minimum error*, so that $x_{i1} = \mathbf{x}_{i\hat{1}}^T\mathbf{u} + w_i$. The error of prediction is w_i. Write $\mathbf{w}$ for the vector of errors (i.e. the i'th component of $\mathbf{w}$ is w_i). Because $\mathbf{w}^T\mathbf{w}$ is minimized by choice of $\mathbf{u}$, we have $\mathbf{w}^T\mathbf{1} = 0$ (i.e. the average of the w_i's is zero). Assume that these predictions are very good, so that there is some small positive number ϵ so that $\mathbf{w}^T\mathbf{w} \leq \epsilon$.

(a) Show that $\mathcal{X}^T\mathcal{X}$ is positive semi-definite, and so all its eigenvalues are non-negative.
(b) Show that, *for any vector* $\mathbf{v}$,

$$\mathbf{v}^T\left(\mathcal{X}^T\mathcal{X} + \lambda\mathcal{I}\right)\mathbf{v} \geq \lambda\mathbf{v}^T\mathbf{v}$$

and use this to argue that the smallest eigenvalue of $\left(\mathcal{X}^T\mathcal{X} + \lambda\mathcal{I}\right)$ is greater than λ.
(c) Write $\mathbf{b}$ for an eigenvector of $\mathcal{X}^T\mathcal{X}$ with eigenvalue $\lambda_{\mathbf{b}}$. Show that $\mathbf{b}$ is an eigenvector of $\left(\mathcal{X}^T\mathcal{X} + \lambda\mathcal{I}\right)$ with eigenvalue $\lambda_{\mathbf{b}} + \lambda$.
(d) Assume that $\mathcal{X}^T\mathcal{X}$ is positive definite and has no repeated eigenvalues (this doesn't affect the point of this exercise, but it greatly simplifies the reasoning). Recall $\mathcal{X}^T\mathcal{X}$ is a $d \times d$ matrix which is symmetric, and so has d orthonormal eigenvectors. Write $\mathbf{b}_i$ for the i'th such vector, and $\lambda_{\mathbf{b}_i}$ for the corresponding eigenvalue. Show that

$$\mathcal{X}^T\mathcal{X}\beta - \mathcal{X}^T\mathbf{Y} = 0$$

is solved by

$$\beta = \sum_{i=1}^{d} \frac{\left(\mathbf{Y}^T\mathcal{X}\mathbf{b}_i\right)\mathbf{b}_i}{\lambda_{\mathbf{b}_i}}.$$

(Hint: an easy way to do this is to show that the eigenvectors are an orthonormal basis for d dimensional space, and that $\left(\mathcal{X}^T\mathcal{X}\beta - \mathcal{X}^T\mathbf{Y}\right)\mathbf{b}_i = 0$ for any i.)
(e) Using the notation of the previous sub exercise, show that

$$(\mathcal{X}^T\mathcal{X} + \lambda\mathcal{I})\beta - \mathcal{X}^T\mathbf{Y} = 0$$

is solved by

$$\beta = \sum_{i=1}^{d} \frac{\left(\mathbf{Y}^T\mathcal{X}\mathbf{b}_i\right)\mathbf{b}_i}{\lambda_{\mathbf{b}_i} + \lambda}.$$

Use this expression to explain why a regularized regression may produce better results on test data than an unregularized regression.

Programming Exercises

13.9 At http://www.statsci.org/data/general/brunhild.html, you will find a dataset that measures the concentration of a sulfate in the blood of a baboon named Brunhilda as a function of time. Build a linear regression of the log of the concentration against the log of time.

(a) Prepare a plot showing (a) the data points and (b) the regression line in log-log coordinates.
(b) Prepare a plot showing (a) the data points and (b) the regression curve in the original coordinates.
(c) Plot the residual against the fitted values in log-log and in original coordinates.
(d) Use your plots to explain whether your regression is good or bad and why.

13.10 At http://www.statsci.org/data/oz/physical.html, you will find a dataset of measurements by M. Larner, made in 1996. These measurements include body mass, and various diameters. Build a linear regression of predicting the body mass from these diameters.

(a) Plot the residual against the fitted values for your regression.
(b) Now regress the cube root of mass against these diameters. Plot the residual against the fitted values in both these cube root coordinates and in the original coordinates.
(c) Use your plots to explain which regression is better.

13.11 At https://archive.ics.uci.edu/ml/datasets/Abalone, you will find a dataset of measurements by W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn and W. B. Ford, made in 1992. These are a variety of measurements of blacklip abalone (*Haliotis rubra*; delicious by repute) of various ages and genders.

(a) Build a linear regression predicting the age from the measurements, ignoring gender. Plot the residual against the fitted values.
(b) Build a linear regression predicting the age from the measurements, including gender. There are three levels for gender; I'm not sure whether this has to do with abalone biology or difficulty in determining gender. You can represent gender numerically by choosing one for one level, 0 for another, and -1 for the third. Plot the residual against the fitted values.
(c) Now build a linear regression predicting the log of age from the measurements, ignoring gender. Plot the residual against the fitted values.
(d) Now build a linear regression predicting the log age from the measurements, including gender, represented as above. Plot the residual against the fitted values.
(e) It turns out that determining the age of an abalone is possible, but difficult (you section the shell, and count rings). Use your plots to explain which regression you would use to replace this procedure, and why.
(f) Can you improve these regressions by using a regularizer? Use glmnet to obtain plots of the cross-validated prediction error.