02:14 Friday 13th November, 2015
See updates and corrections at http://www.stat.cmu.edu/~cshalizi/mreg/

Lecture 20: Outliers and Influential Points

36-401, Fall 2015, Section B

5 November 2015

Contents

1 Outliers Are Data Points Which Break a Pattern 2
1.1 Examples with Simple Linear Regression 5

2 Influence of Individual Data Points on Estimates 7
2.1 Leverage 7

3 Studentized Residuals 10

4 Leave-One-Out 12
4.1 Fitted Values and Cross-Validated Residuals 12
4.2 Cook's Distance 13
4.3 Coefficients 13
4.4 Leave-More-Than-One-Out 15

5 Practically, and with R 15
5.1 In R 16
5.2 `plot` 18

6 Responses to Outliers 20
6.1 Deletion 20
6.2 Changing the Model 21
6.3 Robust Linear Regression 21

7 Exercises 23

An **outlier** is a data point which is very far, somehow, from the rest of the data. They are often worrisome, but not always a problem. When we are doing regression modeling, in fact, we don't really care about whether some data point is far from the rest of the data, but whether it breaks a pattern the rest of the data seems to follow. Here, we'll first try to build some intuition for when outliers cause trouble in linear regression models. Then we'll look at some ways of quantifying how much influence particular data points have on the model; consider the strategy of pretending that inconvenient data doesn't exist; and take a brief look at the **robust regression** strategy, of replacing least squares estimates with others which are less easily influenced.

1 Outliers Are Data Points Which Break a Pattern

Consider Figure 1. The points marked in red and blue are clearly not like the main cloud of the data points, even though their x and y coordinates are quite typical of the data as a whole: the x coordinates of those points aren't related to the y coordinates in the right way, they break a pattern. On the other hand, the point marked in green, while its coordinates are very weird on both axes, does not break that pattern — it was positioned to fall right on the regression line.

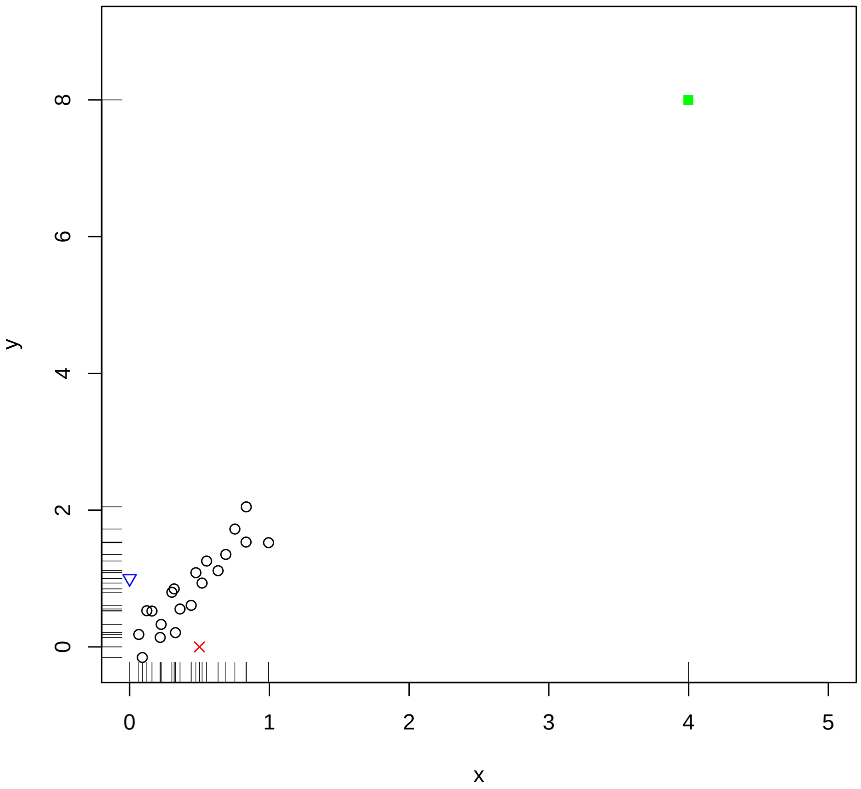


FIGURE 1: *Points marked with a red × and a blue triangle are outliers for the regression line through the main cloud of points, even though their x and y coordinates are quite typical of the marginal distributions (see rug plots along axes). The point marked by the green square, while an outlier along both axes, falls right along the regression line. (See the source file online for the figure-making code.)*

	(Intercept)	x
black only	-0.0174	1.96
black+blue	-0.0476	1.93
black+red	0.1450	1.69
black+green	-0.0359	2.00
all points	-0.0108	1.97

Table 1: *Estimates of the simple regression line from the black points in Figure 1, plus re-estimates adding in various outliers.*

What affect do these different outliers have on a simple linear model here? Table 1 shows the estimates we get from using just the black points, from adding only one of the three outlying points to the black points, and from using all the points. As promised, adding the red or blue points shifts the line, while adding the green point changes hardly anything at all.

If we are worried that outliers might be messing up our model, we would like to quantify how much the estimates change if we add or remove individual data points. Fortunately, we can quantify this using only quantities we estimated on the complete data, especially the hat matrix.

1.1 Examples with Simple Linear Regression

To further build intuition, let's think about what happens with simple linear regression for a moment; that is, our model is

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

with a single real-valued predictor variable X. When we estimate the coefficients by least squares, we know that

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x} \tag{2}$$

Let us turn this around. The fitted value at $X = \overline{x}$ is

$$\hat{\beta}_0 + \hat{\beta}_1 \overline{x} = \overline{y} \tag{3}$$

Suppose we had a data point, say the i^{th} point, where $X = \overline{x}$. Then the actual value of y_i almost wouldn't matter for the fitted value there — the regression line *has* to go through $\overline{y}$ at $\overline{x}$, never mind whether y_i there is close to $\overline{y}$ or far away. If $x_i = \overline{x}$, we say that y_i has little *leverage* over $\hat{m}_i$, or little *influence* on $\hat{m}_i$. It has *some* influence, because y_i is part of what we average to get $\overline{y}$, but that's not a lot of influence.

Again, with simple linear regression, we know that

$$\hat{\beta}_1 = \frac{c_{XY}}{s^2_X} \tag{4}$$

the ratio between the sample covariance of X and Y and the sample variance of X. How does y_i show up in this? It's

$$\hat{\beta}_1 = \frac{n^{-1}\sum_{i=1}^{n}{(x_i - \overline{x})(y_i - \overline{y})}}{s^2_X} \tag{5}$$

Notice that when $x_i = \overline{x}$, y_i doesn't actually matter at all to the slope. If x_i is far from $\overline{x}$, then $y_i - \overline{y}$ will contribute to the slope, and its contribution will get bigger (whether positive or negative) as $x_i - \overline{x}$ grows. y_i will also make a big contribution to the slope when $y_i - \overline{y}$ is big (unless, again, $x_i = \overline{x}$).

Let's write a general formula for the predicted value, at an arbitrary point $X = x$.

$$\begin{aligned}
\hat{m}(x) &= \hat{\beta}_0 + \hat{\beta}_1 x && (6)\\
&= \overline{y} - \hat{\beta}_1 \overline{x} + \hat{\beta}_1 x && (7)\\
&= \overline{y} + \hat{\beta}_1 (x - \overline{x}) && (8)\\
&= \overline{y} + \frac{1}{n}\frac{\sum_{i=1}^{n}{(x_i - \overline{x})(y_i - \overline{y})}}{s^2_X}(x - \overline{x}) && (9)
\end{aligned}$$

So, in words:

- The predicted value is always a weighted average of all the y_i.

- As x_i moves away from $\overline{x}$, y_i gets more weight (possibly a large negative weight). When $x_i = \overline{x}$, y_i only matters because it contributes to the global mean $\overline{y}$.
- The weights on all data points increase in magnitude when the point x where we're trying to predict is far from $\overline{x}$. If $x = \overline{x}$, only $\overline{y}$ matters.

All of this is still true of the fitted values at the original data points:

- If x_i is at $\overline{x}$, y_i only matters for the fit because it contributes to $\overline{y}$.
- As x_i moves away from $\overline{x}$, in either direction, it makes a bigger contribution to *all* the fitted values.

Why is this happening? We get the coefficient estimates by minimizing the mean squared error, and the MSE treats all data points equally:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{m}(x_i))^2 \qquad (10)$$

But we're not just using any old function $\hat{m}(x)$; we're using a linear function. This has only two parameters, so we can't change the predicted value to match each data point — altering the parameters to bring $\hat{m}(x_i)$ closer to y_i might actually increase the error elsewhere. By minimizing the over-all MSE with a linear function, we get two constraints,

$$\overline{y} = \hat{\beta}_0 + \hat{\beta}_1\overline{x} \qquad (11)$$

and

$$\sum_i e_i(x_i - \overline{x}) = 0 \qquad (12)$$

The first of these makes the regression line insensitive to y_i values when x_i is close to $\overline{x}$. The second makes the regression line *very* sensitive to residuals when $x_i - \overline{x}$ is big — when $x_i - \overline{x}$ is large, a big residual (e_i far from 0) is harder to balance out than if $x_i - \overline{x}$ were smaller.

So, let's sum this up.

- Least squares estimation tries to bring all the predicted values closer to y_i, but it can't match each data point at once, because the fitted values are all functions of the same coefficients.
- If x_i is close to $\overline{x}$, y_i makes little difference to the coefficients or fitted values — they're pinned down by needing to go through the mean of the data.
- As x_i moves away from $\overline{x}$, $y_i - \overline{y}$ makes a bigger and bigger impact on both the coefficients and on the fitted values.

If we worry that some point isn't falling on the same regression line as the others, we're really worrying that including it will throw off our estimate of the line. This is going to be a concern when x_i is far from $\overline{x}$, or when the combination of $x_i - \overline{x}$ and $y_i - \overline{y}$ makes that point has a disproportionate impact on the estimates. We should also be worried if the residual values are too big, but when asking what's "too big", we need to take into account the fact that the model will try harder to fit some points than others. A big residual at a point of high leverage is more of a red flag than an equal-sized residual at point with little influence.

All of this will carry over to multiple regression models, but with more algebra to keep track of the different dimensions.

2 Influence of Individual Data Points on Estimates

Recall that our least-squares coefficient estimator is

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{13}$$

from which we get our fitted values as

$$\widehat{\mathbf{m}} = \mathbf{x}\widehat{\beta} = \mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} = \mathbf{H}\mathbf{y} \tag{14}$$

with the hat matrix $\mathbf{H} \equiv \mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T$. This leads to a very natural sense in which one observation might be more or less influential than another:

$$\frac{\partial\hat{\beta}_k}{\partial y_i} = \left((\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\right)_{ki} \tag{15}$$

and

$$\frac{\partial\hat{m}_k}{\partial y_i} = H_{ii} \tag{16}$$

If y_i were different, it would change the estimates for all the coefficients and for all the fitted values. The rate at which the k^{th} coefficient or fitted value changes is given by the ki^{th} entry in these matrices — matrices which, notice, are completely defined by the design matrix $\mathbf{x}$.

2.1 Leverage

H_{ii} is the influence of y_i on its own fitted value; it tells us how much of $\hat{m}_i$ is just y_i. This turns out to be a key quantity in looking for outliers, so we'll give it a special name, the **leverage**. It is sometimes also written h_i. Once again, the leverage of the i^{th} data point doesn't depend on y_i, only on the design matrix.

Because the general linear regression model doesn't assume anything about the distribution of the predictors, other than that they're not collinear, we can't say definitely that some values of the leverage break model assumptions, or even are very unlikely under the model assumptions. But we can say some things about the leverage.

Average leverages We showed in the homework that the trace of the hat matrix equals the number of coefficients we estimate:

$$\operatorname{tr} \mathbf{H} = p + 1 \tag{17}$$

But the trace of any matrix is the sum of its diagonal entries,

$$\operatorname{tr} \mathbf{H} = \sum_{i=1}^{n} H_{ii} \tag{18}$$

so the trace of the hat matrix is the sum of each point's leverage. The average leverage is therefore $\frac{p+1}{n}$. We don't expect every point to have exactly the same leverage, but if some points have much more than others, the regression function is going to be pulled towards fitting the high-leverage points, and the function will tend to ignore the low-leverage points.

Leverage vs. geometry Let's center all the predictor variables, i.e., subtract off the mean of each predictor variable. Call this new vector of predictor variables Z, with the $n \times p$ matrix $\mathbf{z}$. This will not change any of the slopes, and will fix the intercept to be $\overline{y}$. The fitted values then come from

$$\hat{m}_i = \overline{y} + \frac{1}{n}(\mathbf{x}_i - \overline{\mathbf{x}})\mathrm{Var}\left[X\right]^{-1} \mathbf{z}^T \mathbf{y} \tag{19}$$

This tells us that y_i will have a lot of leverage if $(\mathbf{x}_i - \overline{\mathbf{x}})\mathrm{Var}\left[X\right]^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})^T$ is big[1]. If the data point falls exactly at the mean of the predictors, y_i matters only because it contributes to the over-all mean $\overline{y}$. If the data point moves away from the mean of the predictors, not all directions count equally. Remember the eigen-decomposition of $\mathrm{Var}\left[X\right]$:

$$\mathrm{Var}\left[X\right] = \mathbf{V}\mathbf{U}\mathbf{V}^T \tag{20}$$

where $\mathbf{V}$ is the matrix whose columns are the eigenvectors of $\mathrm{Var}\left[X\right]$, $\mathbf{V}^T = \mathbf{V}^{-1}$, and $\mathbf{U}$ is the diagonal matrix of the eigenvalues of $\mathrm{Var}\left[X\right]$. Each eigenvalue gives the variance of the predictors along the direction of the corresponding eigenvector. It follows that

$$\mathrm{Var}\left[X\right]^{-1} = \mathbf{V}\mathbf{U}^{-1}\mathbf{V} \tag{21}$$

So if the data point is far from the center of the predictors along a high-variance direction, that doesn't count as much as being equally far along a low-variance direction[2]. Figure 2 shows a distribution for two predictor variables we're very familiar with, together with the two eigenvectors from the variance matrix, and the corresponding surface of leverages.

[1] This sort of thing — take the difference between two vectors, multiply by an inverse variance matrix, and multiply by the difference vector again — is called a **Mahalanobis distance**. As we will see in a moment, it gives more attention to differences along coordinates where the variance is small, and less attention to differences along coordinates where the variance is high.

[2] I have an unfortunate feeling that I said this backwards throughout the afternoon.

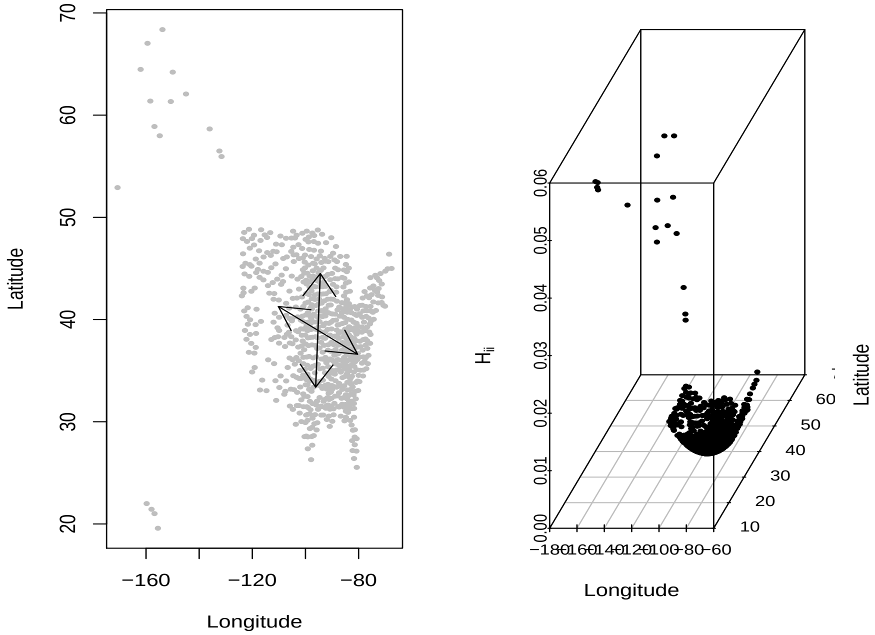


Figure 2: *Left: The geographic coordinates of the communities from the* `mobility` *data, along with their mean, and arrows marking the eigenvectors of the variance-covariance matrix (lengths scaled by the eigenvalues). Right: leverages for each point when regressing rates of economic mobility (or anything else) on latitude and longitude. See online for the code.*

You may convince yourself that with one predictor variable, all of this collapses down to just $1/n + (x_i - \overline{x})^2/ns_X^2$ (Exercise 1). This leads to plots which may be easier to grasp (Figure 3).

One curious feature of the leverage is, and of the hat matrix in general, is that it doesn't care *what* we are regressing on the predictor variables; it could be economic mobility or sightings of Bigfoot, and the same design matrix will give us the same hat matrix and leverages.

To sum up: The leverage of a data point just depends on the value of the predictors there; it increases as the point moves away from the mean of the predictors. It increases more if the difference is along low-variance coordinates, and less for differences along high-variance coordinates.

3 Studentized Residuals

We return once more to the hat matrix, the source of all knowledge.

$$\widehat{\mathbf{m}} = \mathbf{H}\mathbf{y} \tag{22}$$

The residuals, too, depend only on the hat matrix:

$$\mathbf{e} = \mathbf{y} - \widehat{\mathbf{m}} = (\mathbf{I} - \mathbf{H})\mathbf{y} \tag{23}$$

We know that the residuals vary randomly with the noise, so let's re-write this in terms of the noise (Exercise 2).

$$\mathbf{e} = (\mathbf{I} - \mathbf{H})\epsilon \tag{24}$$

Since $\mathbb{E}\left[\epsilon\right] = \mathbf{0}$ and $\mathrm{Var}\left[\epsilon\right] = \sigma^2\mathbf{I}$, we have

$$\mathbb{E}\left[\mathbf{e}\right] = \mathbf{0} \tag{25}$$

and

$$\mathrm{Var}\left[\mathbf{e}\right] = \sigma^2(\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})^T = \sigma^2(\mathbf{I} - \mathbf{H}) \tag{26}$$

If we also assume that the noise is Gaussian, the residuals are Gaussian, with the stated mean and variance.

What does this imply for the residual at the i^{th} data point? It has expectation 0,

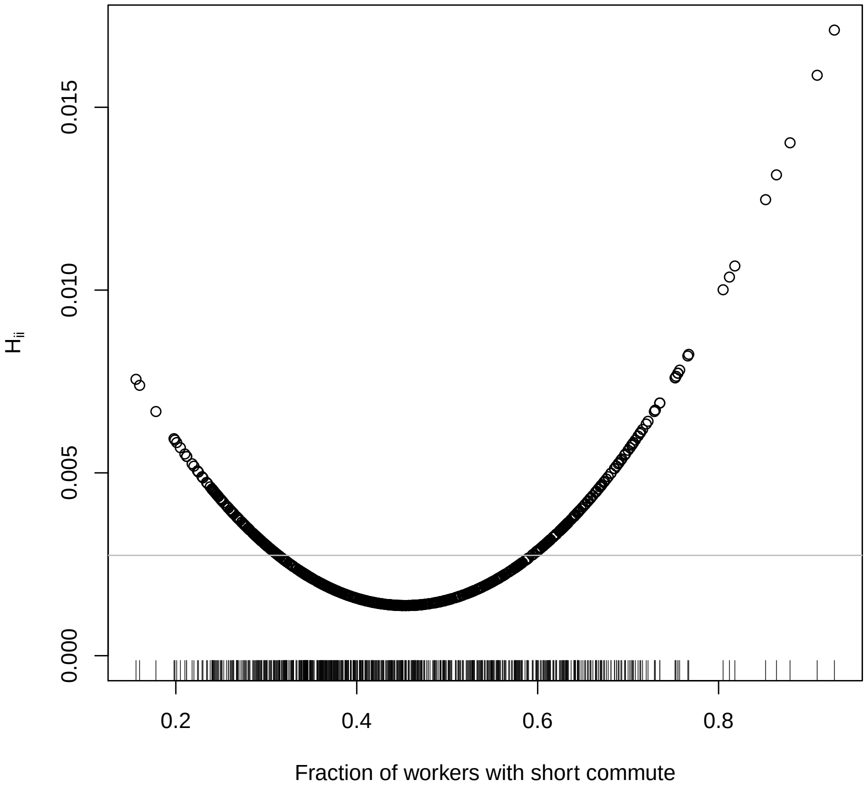
$$\mathbb{E}\left[e_i\right] = 0 \tag{27}$$

and it has a variance which depends on i through the hat matrix:

$$\mathrm{Var}\left[e_i\right] = \sigma^2(\mathbf{I} - \mathbf{H})_{ii} = \sigma^2(1 - H_{ii}) \tag{28}$$

In words: the bigger the leverage of i, the smaller the variance of the residual there. This is yet another sense in which points with high leverage are points which the model tries very hard to fit.

Previously, when we looked at the residuals, we expected them to all be of roughly the same magnitude. This rests on the leverages H_{ii} being all about the



```
H.mob.lm <- hatvalues(lm(Mobility~Commute,data=mobility))
plot(mobility$Commute, H.mob.lm, ylim=c(0,max(H.mob.lm)),
     xlab="Fraction of workers with short commute",
     ylab=expression(H[ii]))
abline(h=2/nrow(mobility),col="grey")
rug(mobility$Commute,side=1)
```

FIGURE 3: *Leverages (H_{ii}) for a simple regression of economic mobility (or anything else) against the fraction of workers with short commutes. The grey line marks the average we'd see if every point was exactly equally influential. Note how leverage increases automatically as* `Commute` *moves away from its mean in either direction. (See below for the* `hatvalues` *function.*

same size. If there are substantial variations in leverage across the data points, it's better to scale the residuals by their expected size.

The usual way to do this is through the **standardized** or **studentized residuals**

$$r_i \equiv \frac{e_i}{\hat{\sigma}\sqrt{1 - H_{ii}}} \tag{29}$$

Why "studentized"? Because we're dividing by an estimate of the standard error, just like in "Student's" t-test for differences in means[3]

All of the residual plots we've done before can also be done with the studentized residuals. In particular, the studentized residuals should look flat, with constant variance, when plotted against the fitted values or the predictors.

4 Leave-One-Out

Suppose we left out the i^{th} data point altogether. How much would that change the model?

4.1 Fitted Values and Cross-Validated Residuals

Let's take the fitted values first. The hat matrix, $\mathbf{H}$, is an $n \times n$ matrix. If we deleted the i^{th} observation when estimating the model, but still asked for a prediction at $\mathbf{x}_i$, we'd get a different, $n \times (n-1)$ matrix, say $\mathbf{H}^{(-i)}$. This in turn would lead to a new fitted value:

$$\hat{m}^{(-i)}(\mathbf{x}_i) = \frac{(\mathbf{H}\mathbf{y})_i - \mathbf{H}_{ii} y_i}{1 - \mathbf{H}_{ii}} \tag{30}$$

Basically, this is saying we can take the old fitted value, and then subtract off the part of it which came from having included the observation y_j in the first place. Because each row of the hat matrix has to add up to 1 (Exercise 3), we need to include the denominator (Exercise 4).

The **leave-one-out residual** is the difference between this and y_i:

$$e_i^{(-i)} \equiv y_i - \hat{m}^{(-i)}(\mathbf{x}_i) \tag{31}$$

That is, this is how far off the model's prediction of y_i would be if it didn't actually get to see y_i during the estimation, but had to honestly predict it.

Leaving out the data point i would give us an MSE of $\hat{\sigma}^2_{(-i)}$, and a little work says that

$$t_i \equiv \frac{e_i^{(-i)}}{\hat{\sigma}_{(-i)}\sqrt{1 + \mathbf{x}_i^T(\mathbf{x}_{(-i)}^T \mathbf{x}_{(-i)})^{-1}\mathbf{x}_i}} \; t_{n-p-2} \tag{32}$$

[3] The distribution here is however not quite a t-distribution, because, while e_i has a Gaussian distribution and $\hat{\sigma}$ is the square root of a χ^2-distributed variable, e_i is actually used in computing $\hat{\sigma}$, hence they're not statistically independent. Rather, $r_i^2/(n-p-1)$ has a $\beta(\frac{1}{2}, \frac{1}{2}(n-p-2))$ distribution (Seber and Lee, 2003, p. 267). This gives us studentized residuals which all have the same distribution, and that distribution does approach a Gaussian as $n \to \infty$ with p fixed.

(The -2 here is because these predictions are based on only $n-1$ data points.) These are called the **cross-validated**, or **jackknife**, or **externally studentized**, residuals. (Some people use the name "studentized residuals" only for these, calling the others the "standardized residuals".) Fortunately, we can compute this without having to actually re-run the regression:

$$\begin{aligned} t_i &= \frac{e_i^{(-i)}}{\hat{\sigma}_{(-i)}\sqrt{1+\mathbf{x}_i^T(\mathbf{x}_{(-i)}^T\mathbf{x}_{(-i)})^{-1}\mathbf{x}_i}} && (33)\\ &= \frac{e_i}{\hat{\sigma}_{(-i)}\sqrt{1-H_{ii}}} && (34)\\ &= r_i\sqrt{\frac{n-p-2}{n-p-1-r_i^2}} && (35) \end{aligned}$$

4.2 Cook's Distance

Omitting point i will generally change all of the fitted values, not just the fitted value at that point. We go from the vector of predictions $\widehat{\mathbf{m}}$ to $\widehat{\mathbf{m}}^{(-i)}$. How big a change is this? It's natural (by this point!) to use the squared length of the difference vector,

$$\|\widehat{\mathbf{m}}-\widehat{\mathbf{m}}^{(-i)}\|^2 = (\widehat{\mathbf{m}}-\widehat{\mathbf{m}}^{(-i)})^T(\widehat{\mathbf{m}}-\widehat{\mathbf{m}}^{(-i)}) \tag{36}$$

To make this more comparable across data sets, it's conventional to divide this by $(p+1)\hat{\sigma}^2$, since there are really only $p+1$ independent coordinates here, each of which might contribute something on the order of $\hat{\sigma}^2$. This is called the **Cook's distance** or **Cook's statistic** for point i:

$$D_i = \frac{(\widehat{\mathbf{m}}-\widehat{\mathbf{m}}^{(-i)})^T(\widehat{\mathbf{m}}-\widehat{\mathbf{m}}^{(-i)})}{(p+1)\hat{\sigma}^2} \tag{37}$$

As usual, there is a simplified formula, which evades having to re-fit the regression:

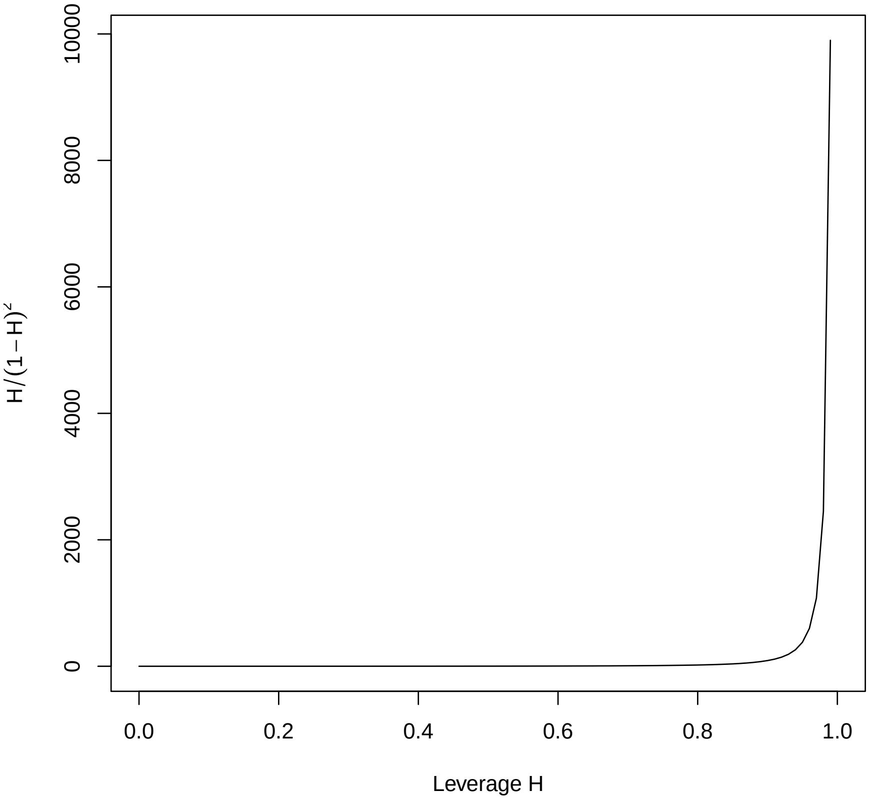
$$D_i = \frac{1}{p+1}e_i^2\frac{H_{ii}}{(1-H_{ii})^2} \tag{38}$$

Notice that $H_{ii}/(1-H_{ii})^2$ is a growing function of H_{ii} (Figure 4). So this says that the total influence of a point over all the fitted values grows with both its leverage (H_{ii}) and the size of its residual when it is included (e_i^2).

4.3 Coefficients

The leave-one-out idea can also be applied to the coefficients. Writing $\widehat{\beta}^{(-i)}$ for the vector of coefficients we get when we drop the i^{th} data point. One can show (Seber and Lee, 2003, p. 268) that

$$\widehat{\beta}^{(-i)} = \widehat{\beta} - \frac{(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}_i^T e_i}{1-H_{ii}} \tag{39}$$



```
curve(x/(1-x)^2, from=0, to=1, xlab="Leverage H",
      ylab=expression(H/(1-H)^2))
```

FIGURE 4: *Illustration of the function $H/(1-H)^2$ relating leverage H to Cook's distance. Notice that leverage must be ≥ 0 and ≤ 1, so this is the whole relevant range of the curve.*

Cook's distance can actually be computed from this, since the change in the vector of fitted values is $\mathbf{x}(\widehat{\beta}^{(-i)} - \widehat{\beta})$, so

$$D_i = \frac{((\widehat{\beta}^{(-i)} - \widehat{\beta})^T \mathbf{x}^T \mathbf{x}(\widehat{\beta}^{(-i)} - \widehat{\beta})}{(p+1)\hat{\sigma}^2} \tag{40}$$

4.4 Leave-More-Than-One-Out

Sometimes, whole clusters of nearby points might be potential outliers. In such cases, removing just one of them might change the model very little, while removing them all might change it a great deal. Unfortunately there are $\binom{n}{k} = O(n^k)$ groups of k points you could consider deleting at once, so while looking at all leave-one-out results is feasible, looking at all leave-two- or leave-ten- out results is not. Instead, you have to think.

5 Practically, and with R

We have three ways of looking at whether points are outliers:

1. We can look at their leverage, which depends only on the value of the predictors.
2. We can look at their studentized residuals, either ordinary or cross-validated, which depend on how far they are from the regression line.
3. We can look at their Cook's statistics, which say how much removing each point shifts all the fitted values; it depends on the product of leverage and residuals.

The model assumptions don't put any limit on how big the leverage can get (just that it's ≤ 1 at each point) or on how its distributed across the points (just that it's got to add up to $p + 1$). Having most of the leverage in a few super-inferential points doesn't break the model, exactly, but it should make us worry.

The model assumptions *do* say how the studentized residuals should be distributed. In particular, the cross-validated studentized residuals should follow a t distribution. This is something we can test, either for specific points which we're worried about (say because they showed up on our diagnostic plots), or across all the points[4].

Because Cook's distance is related to how much the parameters change, the theory of confidence ellipsoids (Lecture 18) can be used to get some idea of how

[4] Be careful about testing all the points. If you use a size α test and everything is fine, you'd see about αn rejections. A good, if not necessarily optimal, way to deal with this is to lower the threshold to α/n for each test — another example of the Bonferroni correction from Lecture 18.

big a D_i is worrying[5]. Cook's original rule-of-thumb translates into worrying when $(p+1)D_i$ is bigger than about $\chi^2_{p+1}(0.1)$, though the 0.1 is arbitrary[6]. However, this is not really a hypothesis test.

5.1 In R

Almost everything we've talked — leverages, studentized residuals, Cook's statistics — can be calculated using the `influence` function. However, there are more user-friendly functions which call that in turn, and are probably better to use.

Leverages come from the ‘hatvalues‘ function, or from the ‘hat‘ component of what ‘influence‘ returns:

```
mob.lm <- lm(Mobility~Commute, data=mobility)
hatvalues(mob.lm)
influence(mob.lm)$hat  # Same as previous line
```

The standardized, or internally-studentized, residuals r_i are available with `rstandard`:

```
rstandard(mob.lm)
residuals(mob.lm)/sqrt(1-hatvalues(mob.lm)) # Same as previous line
```

The cross-validated or externally-studentized residuals t_i are available with `rstudent`:

```
rstudent(mob.lm) # Too tedious to calculate from rstandard though you could
```

Cook's statistic is calculated with `cooks.distance`:

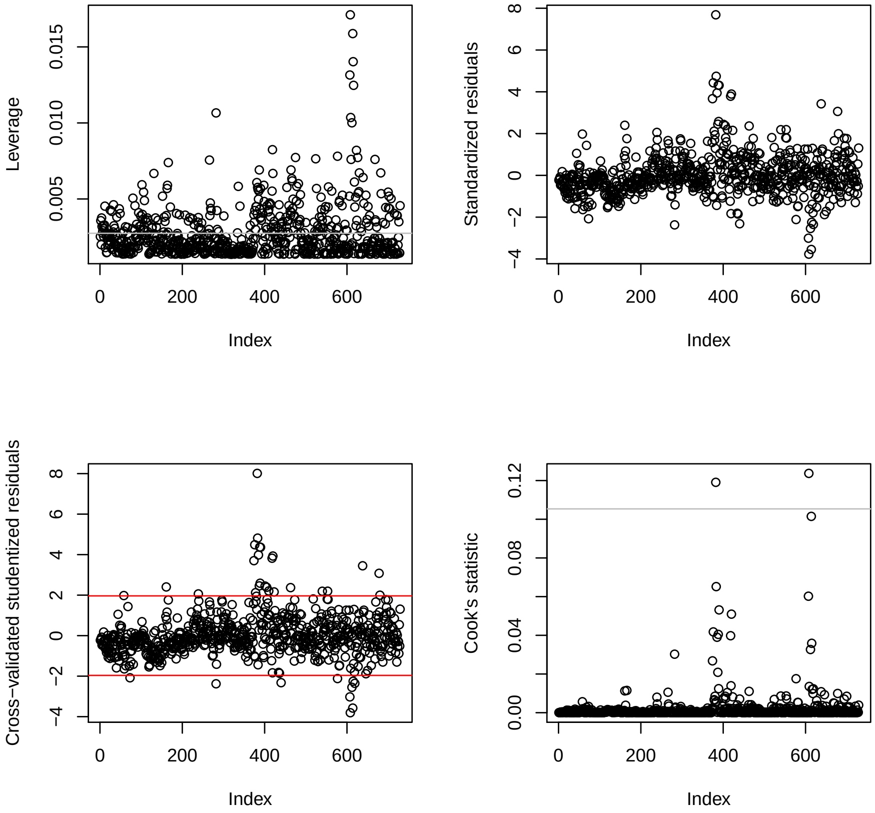
```
cooks.distance(mob.lm)
```

Often the most useful thing to do with these is to plot them, and look at the most extreme points. (One might also rank them, and plot them against ranks.) Figure 5 does so. The standardized and studentized residuals can also be put into our usual diagnostic plots, since they should average to zero and have constant variance when plotted against the fitted values or the predictors. (I omit that here because in this case, $1/\sqrt{1-H_{ii}}$ is sufficiently close to 1 that it makes no visual difference.)

We can now look at exactly which points have the extreme values, say the 10 most extreme residuals, or largest Cook's statistics:

[5] Remember we saw that for large n, $(\widehat{\beta}-\beta)^T\mathbf{\Sigma}^{-1}(\widehat{\beta}-\beta) \sim \chi^2_{p+1}$, where $\mathbf{\Sigma}$ is the variance matrix of the coefficient estimates. But that's $\sigma^2(\mathbf{x}^T\mathbf{x})^{-1}$, so we get $\sigma^{-2}(\widehat{\beta}-\beta)^T\mathbf{x}^T\mathbf{x}(\widehat{\beta}-\beta) \sim \chi^2_{p+1}$. Now compare with Eq. 40.

[6] More exactly, he used an F distribution to take account of small-n uncertainties in $\hat{\sigma}^2$, and suggested worrying when D_i was bigger than $F_{p+1,n-p-1}(0.1)$. This will come to the same thing for large n.



```
par(mfrow=c(2,2))
mob.lm <- lm(Mobility~Commute,data=mobility)
plot(hatvalues(mob.lm), ylab="Leverage")
abline(h=2/nrow(mobility), col="grey")
plot(rstandard(mob.lm), ylab="Standardized residuals")
plot(rstudent(mob.lm), ylab="Cross-validated studentized residuals")
abline(h=qt(0.025,df=nrow(mobility)-2),col="red")
abline(h=qt(1-0.025,df=nrow(mobility)-2),col="red")
plot(cooks.distance(mob.lm), ylab="Cook's statistic")
abline(h=qchisq(0.1,2)/2,col="grey")
```

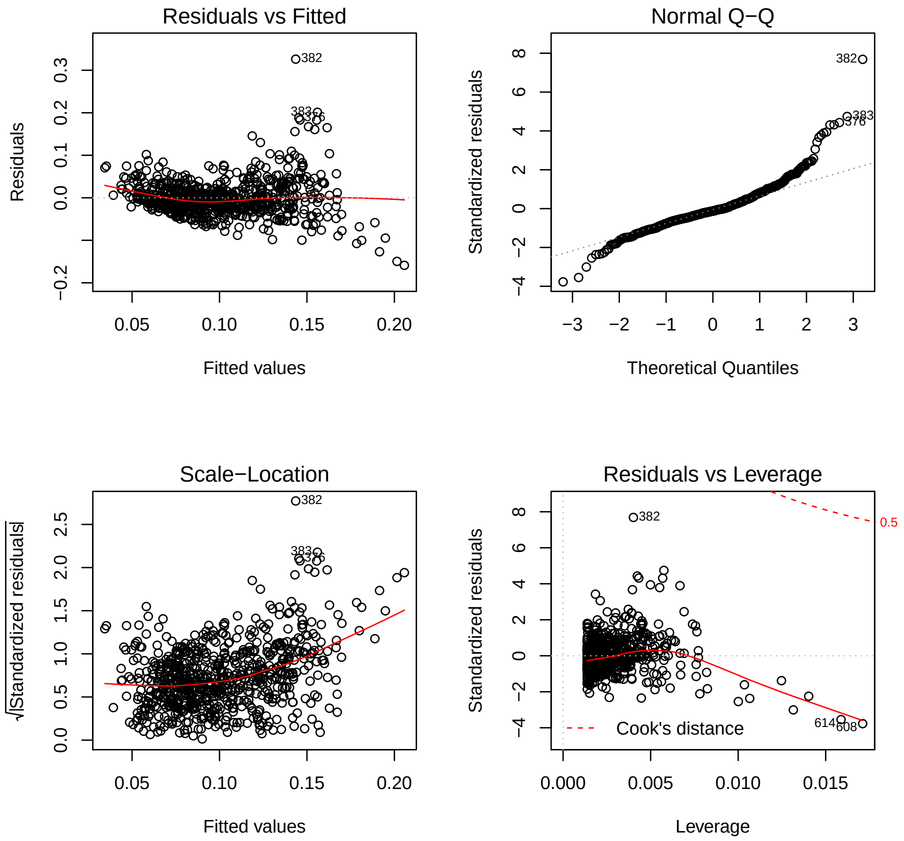
FIGURE 5: *Leverages, two sorts of standardized residuals, and Cook's distance statistic for each point in a basic linear model of economic mobility as a function of the fraction of workers with short commutes. The horizontal line in the plot of leverages shows the average leverage. The lines in studentized residual plot shows a 95% t-distribution sampling interval. (What is the grey line in the plot of Cook's distances?) Note the clustering of extreme residuals* and *leverage around row 600, and another cluster of points with extreme residuals around row 400.*

```
mobility[rank(-abs(rstudent(mob.lm)),)<=10,]

##       X       Name   Mobility State Commute  Longitude Latitude
## 374 375     Linton 0.29891303    ND   0.646 -100.16075 46.31258
## 376 378 Carrington 0.33333334    ND   0.656  -98.86684 47.59698
## 382 384     Bowman 0.46969697    ND   0.648 -103.42526 46.33993
## 383 385     Lemmon 0.35714287    ND   0.704 -102.42011 45.96558
## 385 388 Plentywood 0.31818181    MT   0.681 -104.65381 48.64743
## 388 391  Dickinson 0.32920793    ND   0.659 -102.61354 47.32696
## 390 393  Williston 0.33830845    ND   0.702 -103.33987 48.25441
## 418 422     Miller 0.31506848    SD   0.697  -99.27758 44.53313
## 420 424 Gettysburg 0.32653061    SD   0.729 -100.19547 45.05100
## 608 618       Nome 0.04678363    AK   0.928 -162.03012 64.47514
```

```
mobility[rank(-abs(cooks.distance(mob.lm)))<=10,]

##       X       Name   Mobility State Commute  Longitude Latitude
## 376 378 Carrington 0.33333334    ND   0.656  -98.86684 47.59698
## 382 384     Bowman 0.46969697    ND   0.648 -103.42526 46.33993
## 383 385     Lemmon 0.35714287    ND   0.704 -102.42011 45.96558
## 388 391  Dickinson 0.32920793    ND   0.659 -102.61354 47.32696
## 390 393  Williston 0.33830845    ND   0.702 -103.33987 48.25441
## 418 422     Miller 0.31506848    SD   0.697  -99.27758 44.53313
## 420 424 Gettysburg 0.32653061    SD   0.729 -100.19547 45.05100
## 607 617   Kotzebue 0.06451613    AK   0.864 -159.43781 67.02818
## 608 618       Nome 0.04678363    AK   0.928 -162.03012 64.47514
## 614 624     Bethel 0.05186386    AK   0.909 -158.38213 61.37712
```

5.2 plot

We have not used the **plot** function on an **lm** object yet. This is because most of what it gives us is in fact related to residuals (Figure 6). The first plot is of residuals versus fitted values, plus a smoothing line, with extreme residuals marked by row number. The second is a Q-Q plot of the standardized residuals, again with extremes marked by row number. The third shows the square root of the absolute standardized residuals against fitted values (ideally, flat); the fourth plots standardized residuals against leverage, with contour lines showing equal values of Cook's distance. There are many options, described in **help(plot.lm)**.



```
par(mfrow=c(2,2))
plot(mob.lm)
par(mfrow=c(1,1))
```

FIGURE 6: *The basic* `plot` *function applied to our running example model.*

6 Responses to Outliers

There are essentially three things to do when we're convinced there are outliers: delete them; change the model; or change how we estimate.

6.1 Deletion

Deleting data points should never be done lightly, but it is sometimes the right thing to do.

The best case for removing a data point is when you have good reasons to think it's just wrong (and you have no way to fix it). Medical records which give a patient's blood pressure as 0, or their temperature as 200 degrees, are just impossible and have to be errors[7]. Those points aren't giving you useful information about the process you're studying[8], so getting rid of them makes sense.

The next best case is if you have good reasons to think that the data point isn't *wrong*, exactly, but belongs to a different phenomenon or population from the one you're studying. (You're trying to see if a new drug helps cancer patients, but you discover the hospital has included some burn patients and influenza cases as well.) Or the data point does belong to the right population, but also somehow to another one which isn't what you're interested in right now. (All of the data is on cancer patients, but some of them were also sick with the flu.) You should be careful about that last, though. (After all, some proportion of future cancer patients are also going to have the flu.)

The next best scenario after that is that there's nothing quite so definitely wrong about the data point, but it just looks really weird compared to all the others. Here you are really making a judgment call that either the data really are mistaken, or not from the right population, but you can't put your finger on a concrete reason why. The rules-of-thumb used to identify outliers, like "Cook's distance shouldn't be too big", or "Tukey's rule"[9], are at best of this sort. It is always more satisfying, and more reliable, if investigating how the data were gathered lets you turn cases of this sort into one of the two previous kinds.

The least good case for getting rid of data points which isn't just bogus is that you've got a model which almost works, and would work a lot better if you just get rid of a few stubborn points. This is really a sub-case of the previous one, with added special pleading on behalf of your favorite model. You are here basically trusting your model more than your data, so it had better be either a really good model or really bad data.

Beyond this, we get into what can only be called ignoring inconvenient facts so that you get the answer you want.

[7] This is true whether the temperature is in degrees Fahrenheit, degrees centigrade, or kelvins.

[8] Unless it's the very process of making errors of measurement and recording.

[9] Which flags any point more than 1.5 times the inter-quartile range above the third quartile, or below the first quartile, on any dimension.

6.2 Changing the Model

Outliers are points that break a pattern. This can be because the points are bad, or because we made a bad guess about the pattern. Figure 7 shows data where the cloud of points on the right are definite outliers for any linear model. But I drew those points following a quadratic model, and they fall perfectly along it (as they should). Deleting them, in order to make a linear model work better, would have been short-sighted at best.

The moral of Figure 7 is that data points can look like outliers because we're looking for the wrong pattern. If when we find apparent outliers and we can't convince ourselves that data is erroneous or irrelevant, we should consider changing our model, before, or as well as, deleting them.

6.3 Robust Linear Regression

A final alternative is to change how we estimate our model. Everything we've done has been based on ordinary least-squares (OLS) estimation. Because the squared error grows very rapidly with the error, OLS can be very strongly influenced by a few large "vertical" errors[10]. We might, therefore, consider using not a different statistical model, but a different method of estimating its parameters. Estimation techniques which are less influenced by outliers in the residuals than OLS are called **robust estimators**, or (for regression models) **robust regression**.

Usually (though not always), robust estimation, like OLS, tries to minimize[11] some average of a function of the errors:

$$\tilde{\beta} = \underset{\mathbf{b}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \rho(y_i - \mathbf{x}_i \mathbf{b}) \tag{41}$$

Different choices of ρ, the **loss function**, yield different estimators. $\rho(u) = |u|$ is **least absolute deviation** (LAD) estimation[12]. $\rho(u) = u^2$ is OLS again. A popular compromise is to use **Huber's** loss function[13]

$$\rho(u) = \begin{cases} u^2 & |u| \leq c \\ 2c|u| - c^2 & |u| \geq c \end{cases} \tag{42}$$

Notice that Huber's loss looks like squared error for small errors, but like absolute error for large errors[14]. Huber's loss is designed to be continuous at c, and

[10] Suppose there are 100 data points, and we start with parameter values where $e_1 > 10$, while e_2 through $e_100 = 0$. Changing to a new parameter value where $e_i = 1$ for all i actually reduces the MSE, even though it moves us away from perfectly fitting 99% of the data points.

[11] Hence the name "M-estimators".

[12] For minimizing absolute error, the scenario suggested in the previous footnote seems like a horrible idea, the average loss function goes from 0.1 to 1.0.

[13] Often written ψ, since that's the symbol Huber used when he introduced it. Also, some people define it as 1/2 of the way I have here; this way, though, it's identical to squared error for small u.

[14] If we set $c = 1$ in our little scenario, the average loss would go from 0.19 to 1.0, a definite worsening.

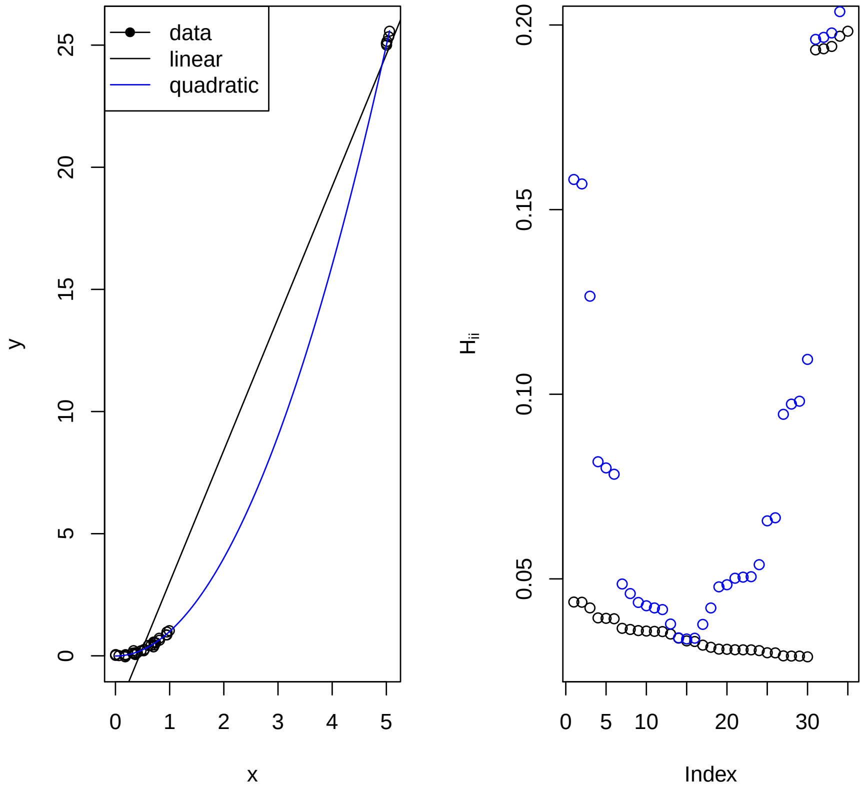


FIGURE 7: *The points in the upper-right are outliers for any linear model fit through the main body of points, but dominate the line because of their very high leverage; they'd be identified as outliers. But all points were generated from a quadratic model.*

have a continuous first derivative there as well (which helps with optimization). We need to pick the scale c at which it switches over from acting like squared error to acting like absolute error; this is usually done using a robust estimate of the noise standard deviation σ.

Robust estimation with Huber's loss can be conveniently done with the `rlm` function in the `MASS` package, which, as the name suggests, is designed to work very much like `lm`.

```
library(MASS)
summary(rlm(Mobility~Commute,data=mobility))
```

```
##
## Call: rlm(formula = Mobility ~ Commute, data = mobility)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.148719 -0.019461 -0.002341  0.021093  0.332347
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept)  0.0028  0.0043     0.6398
## Commute      0.2077  0.0091    22.7939
##
## Residual standard error: 0.0293 on 727 degrees of freedom
```

Robust linear regression is designed for the situation where it's still true that $Y = X\beta + \epsilon$, but the noise ϵ is not very close to Gaussian, and indeed is sometimes "contaminated" by wildly larger values. It does nothing to deal with non-linearity, or correlated noise, or even some points having excessive leverage because we're insisting on a linear model.

7 Exercises

1. Prove that in a simple linear regression

$$H_{ii} = \frac{1}{n}\left(1 + \frac{(x_i - \overline{x})^2}{s^2_X}\right) \tag{43}$$

2. Show that $(\mathbf{I} - \mathbf{H})\mathbf{x}\mathbf{c} = 0$ for any matrix $\mathbf{c}$.

3. Every row of the hat matrix has entries that sum to 1.

   (a) Show that if all of the y_i are equal, say c, then $\hat{\beta}_0 = c$ and all the estimated slopes are 0.

   (b) Using the previous part, show that $\mathbf{1}$, the $n \times 1$ matrix of all 1s, must be an eigenvector of the hat matrix with eigenvalue 1, $\mathbf{H}\mathbf{1} = \mathbf{1}$.

   (c) Using the previous part, show that the sum of each row of $\mathbf{H}$ must be 1, $\sum_{j=1}^{n} H_{ij} = 1$ for all i.

4. *Fitted values after deleting a point*

   (a) (Easier) Presume that $\mathbf{H}^{(-i)}$ can be found by setting $\mathbf{H}_{jk}^{(-i)} = \mathbf{H}_{jk}/(1-\mathbf{H}_{ji})$. Prove Eq. 30.

   (b) (Challenging) Let $\mathbf{x}^{(-i)}$ be $\mathbf{x}$ with its i^{th} row removed. By construction, $\mathbf{H}^{(-i)}$, the $n \times (n-1)$ matrix which gives predictions at all of the original data points, is

   $$\mathbf{H}^{(-i)} = \mathbf{x}((\mathbf{x}^{(-i)})^T \mathbf{x}^{(-i)})^{-1} (\mathbf{x}^{(-i)})^T \tag{44}$$

   Show that this matrix has the form claimed in the previous problem.

5. (Challenging) Derive Eq. 38 for Cook's statistic from the definition. *Hint:* First, derive a formula for $\widehat{\mathbf{m}}_j^{(-i)}$ in terms of the hat matrix. Next, substitute in to the definition of D_i. Finally, you will need to use properties of the hat matrix to simplify.

References

Seber, George A. F. and Alan J. Lee (2003). *Linear Regression Analysis*. New York: Wiley, 2nd edn.