Robust Regression[*]

John Fox & Sanford Weisberg

October 8, 2013

All estimation methods rely on assumptions for their validity. We say that an estimator or statistical procedure is *robust* if it provides useful information even if some of the assumptions used to justify the estimation method are not applicable. Most of this appendix concerns *robust regression*, estimation methods typically for the linear regression model that are insensitive to outliers and possibly high leverage points. Other types of robustness, for example to model misspecification, are not discussed here. These methods were developed beginning in the mid-1960s. With the exception of the L_1 methods described in Section 5, the are not widely used today. A recent mathematical treatment is given by **?**.

1 Breakdown and Robustness

The finite sample breakdown of an estimator/procedure is the smallest fraction α of data points such that if $[n\alpha]$ points $\to \infty$ then the estimator/procuedure also becomes infinite.

The sample mean of $x_1, \ldots, x_n$ is

$$\begin{aligned}
\bar{x}_n &= \frac{1}{n}\sum_{i=1}^{n} x_i \\
&= \frac{1}{n}\left[\sum_{i=1}^{n-1} x_i + x_n\right] \\
&= \frac{n-1}{n}\bar{x}_{n-1} + \frac{1}{n}x_n
\end{aligned}$$

and so if x_n is large enough then $\bar{x}_n$ can be made as large as desired regardless of the other $n-1$ values.

Unlike the mean, the sample median, as an estimate of a population median, can tolerate up to 50% bad values. In general, breakdown cannot exceed 50%. (Why is that?)

2 M-Estimation

Linear least-squares estimates can behave badly when the error distribution is not normal, particularly when the errors are heavy-tailed. One remedy is to remove influential observations from the least-squares fit. Another approach, termed *robust regression*, is to use a fitting criterion that is not as vulnerable as least squares to unusual data.

The most common general method of robust regression is *M-estimation*, introduced by **?**. This class of estimators can be regarded as a generalization of maximum-likelihood estimation, hence the term "M"-estimation.

[*] An appendix to **?**.

We consider only the linear model

$$\begin{aligned} y_i &= \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i \\ &= \mathbf{x}_i' \beta + \varepsilon_i \end{aligned}$$

for the ith of n observations. Given an estimator $\mathbf{b}$ for β, the fitted model is

$$\hat{y}_i = a + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_k x_{ik} + e_i = \mathbf{x}_i' \mathbf{b}$$

and the residuals are given by

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

With M-estimation, the estimates $\mathbf{b}$ are determined by minimizing a particular *objective function* over all $\mathbf{b}$,

$$\sum_{i=1}^{n} \rho(e_i) = \sum_{i=1}^{n} \rho(y_i - \mathbf{x}_i' \mathbf{b})$$

where the function ρ gives the contribution of each residual to the objective function. A reasonable ρ should have the following properties:

- Always nonnegative, $\rho(e) \geq 0$
- Equal to zero when its argument is zero, $\rho(0) = 0$
- Symmetric, $\rho(e) = \rho(-e)$
- Monotone in $|e_i|$, $\rho(e_i) \geq \rho(e_{i'})$ for $|e_i| > |e_{i'}|$

For example, the least-squares ρ-function $\rho(e_i) = e_i^2$ satisfies these requirements, as do many other functions.

Let $\psi = \rho'$ be the derivative of ρ. ψ is called the *influence curve.* Differentiating the objective function with respect to the coefficients $\mathbf{b}$ and setting the partial derivatives to 0, produces a system of $k+1$ estimating equations for the coefficients:

$$\sum_{i=1}^{n} \psi(y_i - \mathbf{x}_i' \mathbf{b}) \mathbf{x}_i' = \mathbf{0}$$

Define the *weight function* $w(e) = \psi(e)/e$, and let $w_i = w(e_i)$.

2.1 Computing

The estimating equations may be written as

$$\sum_{i=1}^{n} w_i (y_i - \mathbf{x}_i' \mathbf{b}) \mathbf{x}_i' = \mathbf{0}$$

Solving these estimating equations is equivalent to a weighted least-squares problem, minimizing $\sum w_i^2 e_i^2$. The weights, however, depend upon the residuals, the residuals depend upon the estimated coefficients, and the estimated coefficients depend upon the weights. An iterative solution (called *iteratively reweighted least-squares*, *IRLS*) is therefore required:

1. Select initial estimates $\mathbf{b}^{(0)}$, such as the least-squares estimates.

2. At each iteration t, calculate residuals $e_i^{(t-1)}$ and associated weights $w_i^{(t-1)} = w\left[e_i^{(t-1)}\right]$ from the previous iteration.

3. Solve for new weighted-least-squares estimates

$$\mathbf{b}^{(t)} = \left[\mathbf{X}'\mathbf{W}^{(t-1)}\mathbf{X}\right]^{-1}\mathbf{X}'\mathbf{W}^{(t-1)}\mathbf{y}$$

   where $\mathbf{X}$ is the model matrix, with $\mathbf{x}_i'$ as its ith row, and $\mathbf{W}^{(t-1)} = \text{diag}\left\{w_i^{(t-1)}\right\}$ is the current weight matrix.

Steps 2 and 3 are repeated until the estimated coefficients converge.

The asymptotic covariance matrix of $\mathbf{b}$ is

$$\mathcal{V}(\mathbf{b}) = \frac{E(\psi^2)}{[E(\psi')]^2}(\mathbf{X}'\mathbf{X})^{-1}$$

Using $\sum[\psi(e_i)]^2$ to estimate $E(\psi^2)$, and $\left[\sum\psi'(e_i)/n\right]^2$ to estimate $[E(\psi')]^2$ produces the *estimated* asymptotic covariance matrix, $\widehat{\mathcal{V}}(\mathbf{b})$ (which is not reliable in small samples).

2.2 Objective Functions

Figure 1 compares the objective functions, and the corresponding ψ and weight functions for three M-estimators: the familiar least-squares estimator; the *Huber* estimator; and the Tukey *bisquare* (or *biweight*) estimator. The objective and weight functions for the three estimators are also given in Table 1.

Both the least-squares and Huber objective functions increase without bound as the residual e departs from 0, but the least-squares objective function increases more rapidly. In contrast, the bisquare objective function levels eventually levels off (for $|e| > k$). Least-squares assigns equal weight to each observation; the weights for the Huber estimator decline when $|e| > k$; and the weights for the bisquare decline as soon as e departs from 0, and are 0 for $|e| > k$.

The value k for the Huber and bisquare estimators is called a *tuning constant*; smaller values of k produce more resistance to outliers, but at the expense of lower efficiency when the errors are normally distributed. The tuning constant is generally picked to give reasonably high efficiency in the normal case; in particular, $k = 1.345\sigma$ for the Huber and $k = 4.685\sigma$ for the bisquare (where σ is the standard deviation of the errors) produce 95-percent efficiency when the errors are normal, and still offer protection against outliers.

In an application, we need an estimate of the standard deviation of the errors to use these results. Usually a robust measure of spread is used in preference to the standard deviation of the residuals. For example, a common approach is to take $\widehat{\sigma} = \text{MAR}/0.6745$, where MAR is the median absolute residual.

Method	*Objective Function*	*Weight Function*
Least-Squares	$\rho_{\text{LS}}(e) = e^2$	$w_{\text{LS}}(e) = 1$
Huber	$\rho_{\text{H}}(e) = \begin{cases} \frac{1}{2}e^2 & \text{for } \lvert e\rvert \le k \\ k\lvert e\rvert - \frac{1}{2}k^2 & \text{for } \lvert e\rvert > k \end{cases}$	$w_{\text{H}}(e) = \begin{cases} 1 & \text{for } \lvert e\rvert \le k \\ k/\lvert e\rvert & \text{for } \lvert e\rvert > k \end{cases}$
Bisquare	$\rho_{\text{B}}(e) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{e}{k}\right)^2\right]^3\right\} & \text{for } \lvert e\rvert \le k \\ k^2/6 & \text{for } \lvert e\rvert > k \end{cases}$	$w_{\text{B}}(e) = \begin{cases} \left[1 - \left(\frac{e}{k}\right)^2\right]^2 & \text{for } \lvert e\rvert \le k \\ 0 & \text{for } \lvert e\rvert > k \end{cases}$

Table 1: Objective function and weight function for least-squares, Huber, and bisquare estimators.

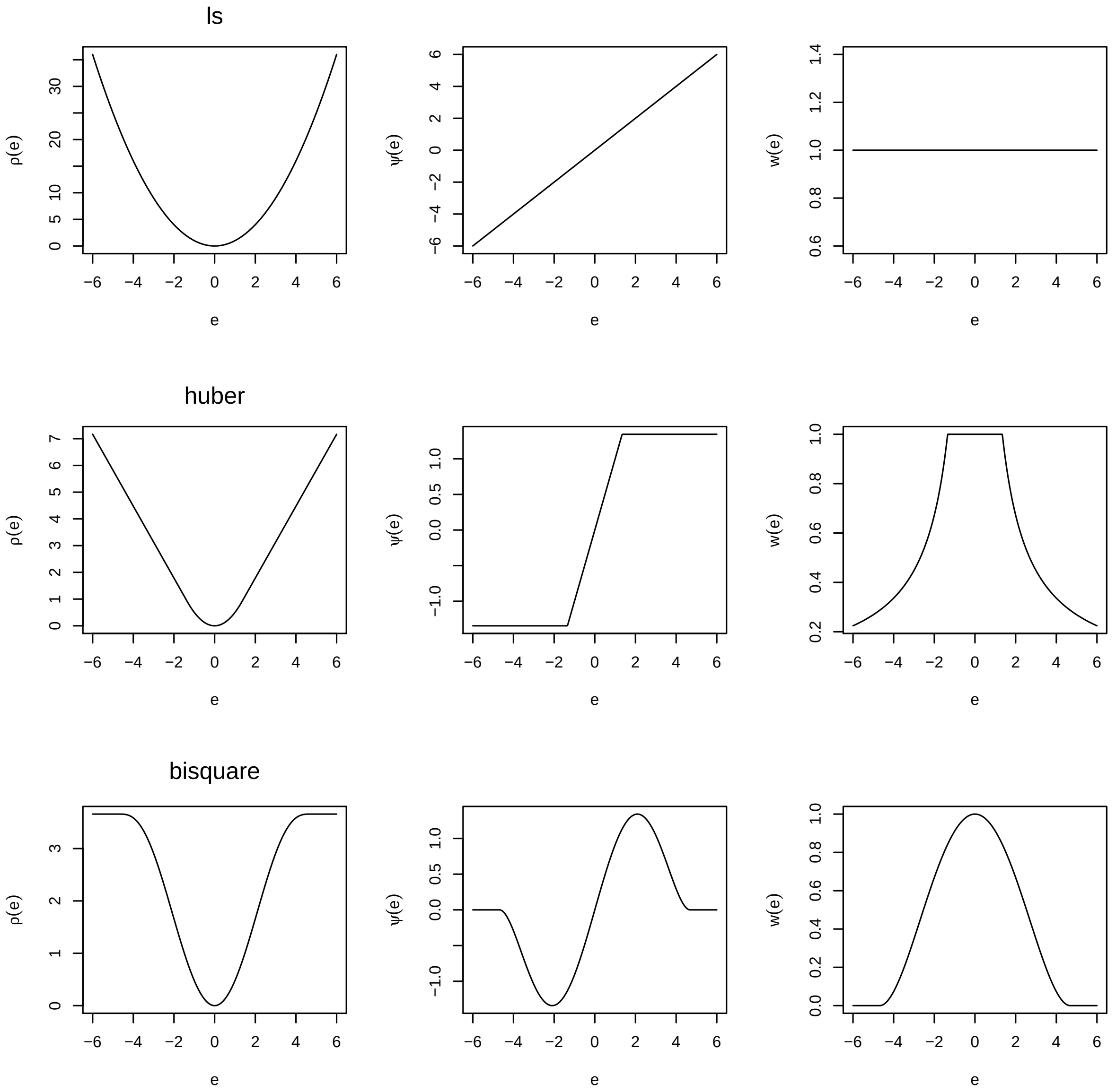


Figure 1: Objective, ψ, and weight functions for the least-squares (top), Huber (middle), and bisquare (bottom) estimators. The tuning constants for these graphs are $k = 1.345$ for the Huber estimator and $k = 4.685$ for the bisquare. (One way to think about this scaling is that the standard deviation of the errors, σ, is taken as 1.)

3 Bounded-Influence Regression

Under certain circumstances, M-estimators can be vulnerable to high-leverage observations. A key concept in assessing influence is the *breakdown point* of an estimator: The breakdown point is the fraction of 'bad' data that the estimator can tolerate without being affected to an arbitrarily large extent. For example, in the context of estimating the center of a distribution, the mean has a breakdown point of 0, because even one bad observation can change the mean by an arbitrary amount; in contrast the median has a breakdown point of 50 percent. Very high breakdown estimators for regression have been proposed and R functions for them are presented here. However, very high breakdown estimates should be avoided unless you have faith that the model you are fitting is correct, as the very high breakdown estimates do not allow for diagnosis of model misspecification, **?**.

One bounded-influence estimator is *least-trimmed squares* (*LTS*) regression. Order the squared residuals from smallest to largest:

$$(e^2)_{(1)}, (e^2)_{(2)}, \ldots, (e^2)_{(n)}$$

The LTS estimator chooses the regression coefficients $\mathbf{b}$ to minimize the sum of the smallest m of the squared residuals,

$$\text{LTS}(\mathbf{b}) = \sum_{i=1}^{m} (e^2)_{(i)}$$

where, typically, $m = \lfloor n/2 \rfloor + \lfloor (k+2)/2 \rfloor$, a little more than half of the observations, and the "floor" brackets, $\lfloor \;\; \rfloor$, denote rounding down to the next smallest integer.

While the LTS criterion is easily described, the mechanics of fitting the LTS estimator are complicated (**?**). Moreover, bounded-influence estimators can produce unreasonable results in certain circumstances, **?**, and there is no simple formula for coefficient standard errors.[1]

4 An Illustration: Duncan's Occupational-Prestige Regression

Duncan's occupational-prestige regression was introduced in Chapter 1 of **?**. The least-squares regression of `prestige` on `income` and `education` produces the following results:

```
> library(car)
> mod.ls <- lm(prestige ~ income + education, data=Duncan)
> summary(mod.ls)

Call:
lm(formula = prestige ~ income + education, data = Duncan)

Residuals:
   Min     1Q Median     3Q    Max
-29.54  -6.42   0.65   6.61  34.64

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -6.0647     4.2719   -1.42     0.16
income        0.5987     0.1197    5.00  1.1e-05
```

[1] Statistical inference for the LTS estimator can be performed by bootstrapping, however. See the Appendix on bootstrapping for an example.

```
education     0.5458     0.0983    5.56  1.7e-06

Residual standard error: 13.4 on 42 degrees of freedom
Multiple R-squared:  0.828,        Adjusted R-squared:  0.82
F-statistic:  101 on 2 and 42 DF,  p-value: <2e-16
```

Recall from the discussion of Duncan's data in **?** that two observations, ministers and railroad conductors, serve to decrease the `income` coefficient substantially and to increase the `education` coefficient, as we may verify by omitting these two observations from the regression:

```
> mod.ls.2 <- update(mod.ls, subset=-c(6,16))
> summary(mod.ls.2)

Call:
lm(formula = prestige ~ income + education, data = Duncan, subset = -c(6,
    16))

Residuals:
   Min     1Q Median     3Q    Max
-28.61  -5.90   1.94   5.62  21.55

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -6.4090     3.6526   -1.75   0.0870
income        0.8674     0.1220    7.11  1.3e-08
education     0.3322     0.0987    3.36   0.0017

Residual standard error: 11.4 on 40 degrees of freedom
Multiple R-squared:  0.876,        Adjusted R-squared:  0.87
F-statistic:  141 on 2 and 40 DF,  p-value: <2e-16
```

Alternatively, let us compute the Huber M-estimator for Duncan's regression model, using the `rlm` (robust linear model) function in the `MASS` library:

```
> library(MASS)
> mod.huber <- rlm(prestige ~ income + education, data=Duncan)
> summary(mod.huber)

Call: rlm(formula = prestige ~ income + education, data = Duncan)
Residuals:
   Min     1Q Median     3Q    Max
-30.12  -6.89   1.29   4.59  38.60

Coefficients:
            Value  Std. Error t value
(Intercept) -7.111  3.881     -1.832
income       0.701  0.109      6.452
education    0.485  0.089      5.438

Residual standard error: 9.89 on 42 degrees of freedom
```

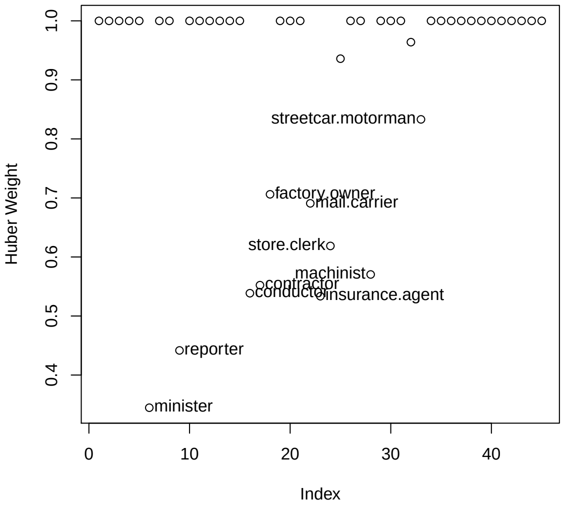


Figure 2: Weights from the robust Huber estimator for the regression of `prestige` on `income` and `education`.)

The `summary` method for `rlm` objects prints the correlations among the coefficients; to suppress this output, specify `correlation=FALSE`. The Huber regression coefficients are between those produced by the least-squares fit to the full data set and by the least-squares fit eliminating the occupations `minister` and `conductor`.

It is instructive to extract and plot (in Figure 2) the final weights used in the robust fit. The `showLabels` function from `car` is used to label all observations with weights less than 0.9.

```
> plot(mod.huber$w, ylab="Huber Weight")
> bigweights <- which(mod.huber$w < 0.9)
> showLabels(1:45, mod.huber$w, rownames(Duncan), id.method=bigweights, cex.=.6)
```

```
        minister            reporter           conductor          contractor
               6                   9                  16                  17
   factory.owner        mail.carrier     insurance.agent         store.clerk
              18                  22                  23                  24
       machinist streetcar.motorman
              28                  33
```

Ministers and conductors are among the observations that receive the smallest weight.

The function `rlm` can also fit the bisquare estimator for Duncan's regression. Starting values for the IRLS procedure are potentially more critical for the bisquare estimator; specifying the argument `method='MM'` to `rlm` requests bisquare estimates with start values determined by a preliminary bounded-influence regression.

```
> mod.bisq <- rlm(prestige ~ income + education, data=Duncan, method='MM')
> summary(mod.bisq, cor=F)

Call: rlm(formula = prestige ~ income + education, data = Duncan, method = "MM")
Residuals:
```

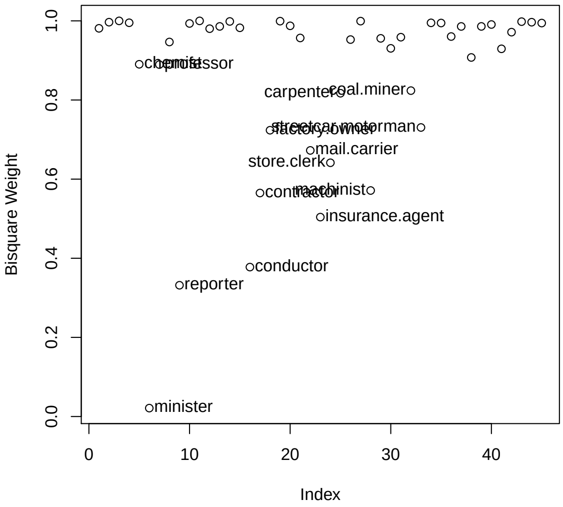


Figure 3: Weights from the robust bisquare estimator for the regression of `prestige` on `income` and `education`.)

```
   Min      1Q Median      3Q     Max
-29.87   -6.63   1.44    4.47   42.40

Coefficients:
            Value  Std. Error t value
(Intercept) -7.389  3.908     -1.891
income       0.783  0.109      7.149
education    0.423  0.090      4.710

Residual standard error: 9.79 on 42 degrees of freedom
```

Compared to the Huber estimates, the bisquare estimate of the `income` coefficient is larger, and the estimate of the `education` coefficient is smaller. Figure 3 shows a graph of the weights from the bisquare fit, interactively identifying the observations with the smallest weights:

```
> plot(mod.bisq$w, ylab="Bisquare Weight")
> showLabels(1:45, mod.bisq$w, rownames(Duncan),
+      id.method= which(mod.bisq$w < 0.9), cex.=0.6)
```

```
          chemist            minister           professor            reporter
                5                   6                   7                   9
        conductor          contractor       factory.owner        mail.carrier
               16                  17                  18                  22
  insurance.agent         store.clerk           carpenter           machinist
               23                  24                  25                  28
       coal.miner streetcar.motorman
               32                  33
```

Finally, the `ltsreg` function in the `lqs` library is used to fit Duncan's model by LTS regression:[2]

```
> (mod.lts <- ltsreg(prestige ~ income + education, data=Duncan))

Call:
lqs.formula(formula = prestige ~ income + education, data = Duncan,
    method = "lts")

Coefficients:
(Intercept)       income    education
     -5.503        0.768        0.432

Scale estimates 7.77 7.63
```

In this case, the results are similar to those produced by the M-estimators. The `print` method for bounded-influence regression gives the regression coefficients and two estimates of the variation or scale of the errors. There is no `summary` method for this class of models.

5 L_1 and Quantile Regression

This section follows **?** and the vignette for quantile regression in the `quantreg` package in R on the class website.

5.1 Sample and population quantiles

Given an distribution F, for any $0 < \tau < 1$ we define the τ-th quantile to be the solution to

$$\xi_\tau(x) = F^{-1}(\tau) = \inf\{x : F(x) \geq \tau\}$$

Sample quantiles $\hat{\xi}_\tau(x)$ are similarly defined, with the sample CDF $\hat{F}$ replacing F.

5.2 L_1 Regression

We start by assuming a model like this:

$$y_i = x_i'\beta + e_i \tag{1}$$

where the e are random variables. We will estimate β by soling the minimization problem

$$\tilde{\beta} = \arg\min \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i'\beta| = \frac{1}{n}\sum_{i=1}^{n} \rho_{.5}(y_i - x_i'\beta) \tag{2}$$

where the objective function $\rho_\tau(u)$ is called in this instance a *check function*,

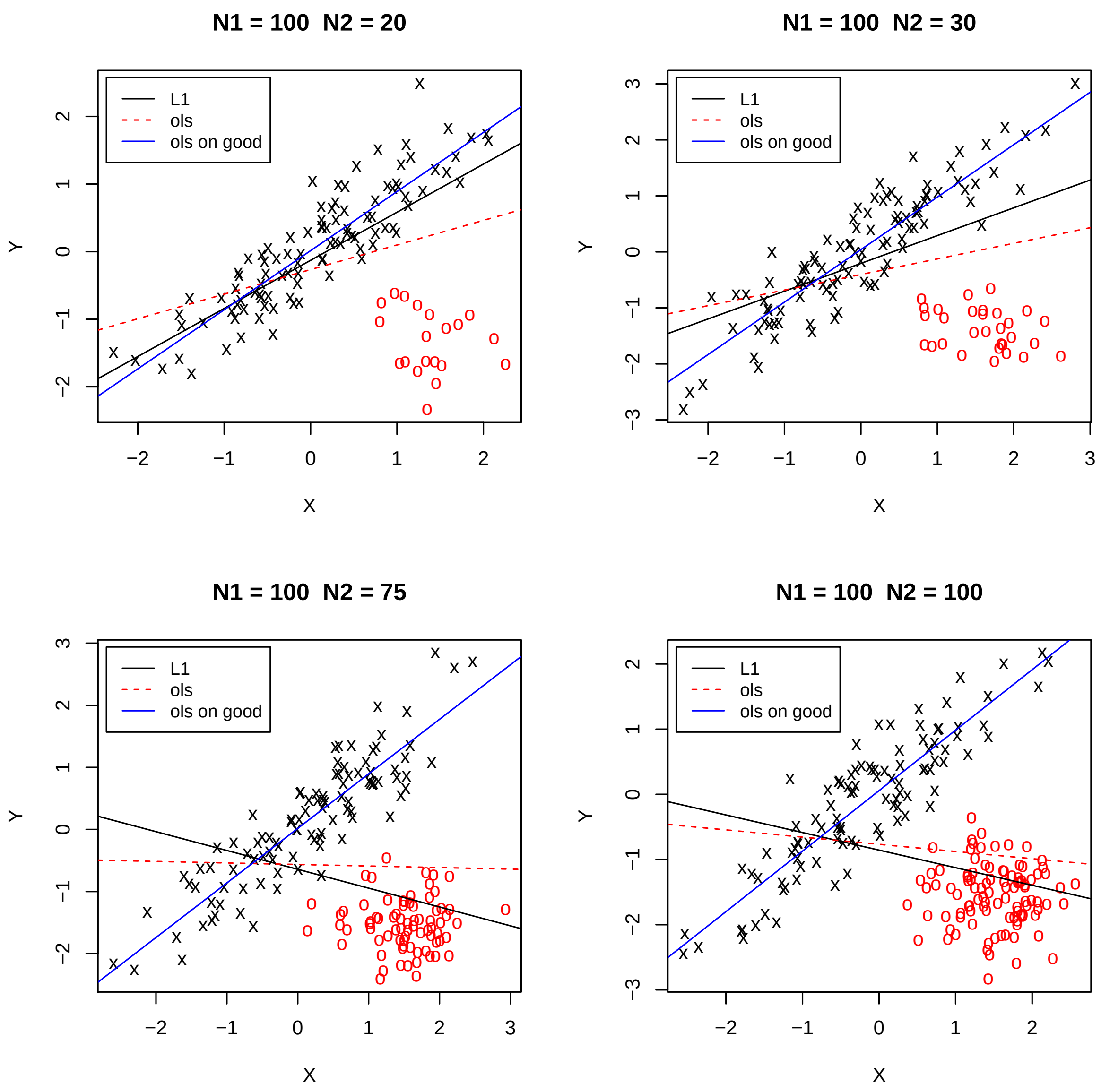
$$\rho_\tau(u) = u \times (\tau - I(u < 0)) \tag{3}$$

where I is the indicator function (more on check functions later). If the e are iid from a double exponential distribution, then $\tilde{\beta}$ will be the corresponding mle for β. In general, however, we will be estimating the *median* at $x_i'\beta$, so one can think of this as *median regression.*

Example We begin with a simple simulated example with n_1 "good" observations and n_2 "bad" ones.

[2] LTS regression is also the default method for the `lqs` function, which additionally can fit other bounded-influence estimators.

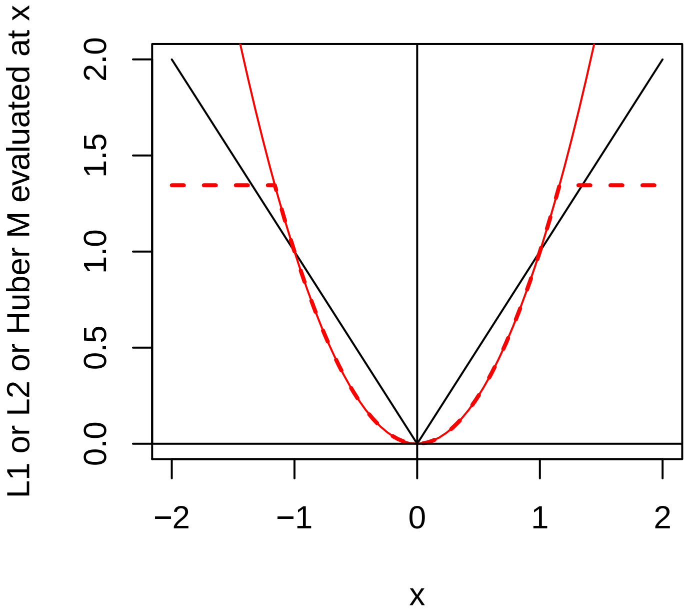
```
> set.seed(10131986)
> library(MASS)
> library(quantreg)
> l1.data <- function(n1=100,n2=20){
+   data <- mvrnorm(n=n1,mu=c(0, 0),
+                   Sigma=matrix(c(1, .9, .9, 1), ncol=2))
+ # generate 20 'bad' observations
+   data <- rbind(data, mvrnorm(n=n2,
+                   mu=c(1.5, -1.5), Sigma=.2*diag(c(1, 1))))
+   data <- data.frame(data)
+   names(data) <- c("X", "Y")
+   ind <- c(rep(1, n1),rep(2, n2))
+   plot(Y ~ X, data, pch=c("x", "o")[ind],
+        col=c("black", "red")[ind], main=paste("N1 =",n1," N2 =", n2))
+   summary(r1 <-rq(Y ~ X, data=data, tau=0.5))
+   abline(r1)
+   abline(lm(Y ~ X, data),lty=2, col="red")
+   abline(lm(Y ~ X, data, subset=1:n1), lty=1, col="blue")
+   legend("topleft", c("L1","ols","ols on good"),
+          inset=0.02, lty=c(1, 2, 1), col=c("black", "red", "blue"),
+          cex=.9)}
> par(mfrow=c(2, 2))
> l1.data(100, 20)
> l1.data(100, 30)
> l1.data(100, 75)
> l1.data(100, 100)
```



5.3 Comparing L_1 and L_2

L_1 minimizes the sum of the absolute errors while L_2 minimizes squared errors. L_1 *gives much less weight to large deviations.* Here are the ρ-functions for L_1 and L_2.

```
> curve(abs(x),-2,2,ylab="L1 or L2 or Huber M evaluated at x" )
> curve(x^2,-3,3,add=T,col="red")
> abline(h=0)
> abline(v=0)
```



5.4 L_1 facts

1. The L_1 estimator is the mle if the errors are independent with a double-exponential distribution.

2. In (1) if x consists only of a "1", then the L_1 estimator is the median.

3. Computations are not nearly as easy as for ls, as a linear programming solution is required.

4. If $X = (x'_1, \dots, x'_n)'$ the $n \times p$ design matrix is of full rank p, then if h is a set at indexes exactly p of the rows of X, there is always an h such that the L_1 estimate $\tilde{\beta}$ fits these p points exactly, so $\tilde{\beta} = (X'_h X_h)^{-1} X'_h y_h = X_h^{-1} y_h$. Of course the number of potential subsets is large, so this may not help much in the computations.

5. L_1 is equivariant, meaning that replacing Y by $a + bY$ and X by $A + B^{-1}X$ will leave the solution essentially unchanged.

6. The breakdown points of the L_1 estimate can be shown to be $1 - 1/\sqrt{2} \approx 0.29$, so about 29% "bad" data can be tolerated.

7. In general we are estimating the median of $y|x$, not the mean.

8. Suppose we have (1) with the errors iid from a distribution F with density f. The population median is $\xi_\tau = F^{-1}(\tau)$ with $\tau = 0.5$, and the sample median is $\hat{\xi}_{.5} = \hat{F}^{-1}(\tau)$. We assume a standardized version of f so $f(u) = (1/\sigma) f_0(u/\sigma)$. Write $Q_n = n^{-1} \sum x_i x'_i$, and suppose that in large samples $Q_n \to Q_0$, a fixed matrix. We will then have:

$$\sqrt{n}(\tilde{\beta} - \beta) \sim \mathrm{N}(0, \omega Q_0^{-1})$$

where $\omega = \sigma^2\tau(1-\tau)/[f_0(F_0^{-1}(\tau))]^2$ and $\tau = 0.50$. For example, if f is the standard normal density, $f(F_0^{-1}(\tau)) = 1/\sqrt{2\pi} = .399$, and $\sqrt{\omega} = .5\sigma/.399 = 1.26\sigma$, so in the normal case the standard deviations of the L_1 estimators are 26% larger than the standard deviations of the ols estimators.

9. If f were known, asymptotic Wald inference/confidence statements can be based on percentiles of the normal distribution. In practice, $f(F^{-1}(\tau))$ must be estimated. One standard method due to Siddiqui is to estimate

$$f(\widehat{F^{-1}}(\tau)) = \left[\hat{F}^{-1}(\tau + h) - \hat{F}^{-1}(\tau - h)\right]/2h$$

for some bandwidth parameter h. This is closely related to density estimation, and so the value of h used in practice is selected using a method appropriate for density estimation. Alternatively, $f(F^{-1}(\tau))$ can be estimated using a bootstrap procedure.

10. For non-iid errors, suppose that $\xi_i(\tau)$ is the τ-quantile for the distribution of the i-th error. One can show that

$$\sqrt{n}(\tilde{\beta} - \beta) \sim \mathrm{N}(0, \tau(1-\tau)H^{-1}Q_0H^{-1})$$

where the matrix H is given by

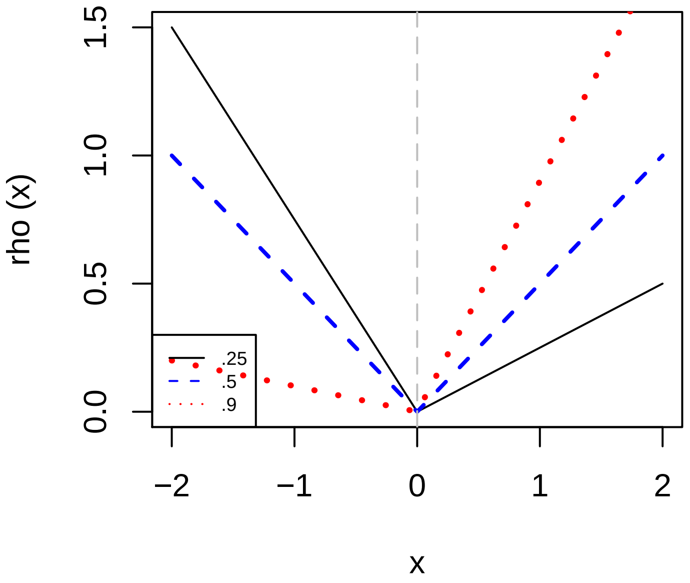
$$H = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} x_i x_i' f_i \xi_i(\tau)$$

a sandwich type estimator is used for estimating the variance of $\tilde{\beta}$. The `rq` function in `quantreg` uses a sandwich formula by default.

6 Quantile regression

L_1 is a special case of *quantile regression* in which we minimize the $\tau = .50$-quantile, but a similar calculation can be done for any $0 < \tau < 1$. Here is what the check function (2) looks like for $\tau \in \{.25, .5, .9\}$.

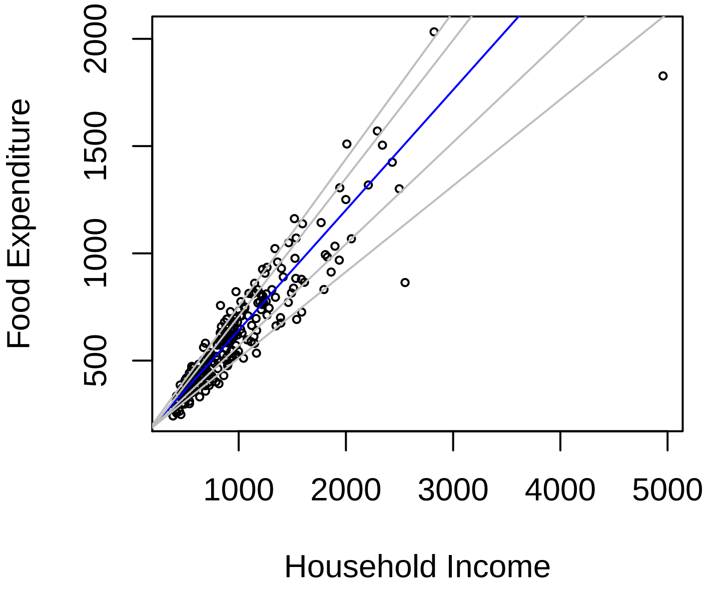
```
> rho <- function(u) {
+  u * (tau - ifelse(u < 0,1,0)  )}
> tau <- .25; curve(rho,-2,2,lty=1)
> tau <- .50; curve(rho, -2,2,lty=2,col="blue",add=T,lwd=2)
> tau <- .90; curve(rho, -2,2,lty=3,col="red",add=T, lwd=3)
> abline(v=0,lty=5,col="gray")
> legend("bottomleft",c(".25",".5",".9"),lty=1:3,col=c("black","blue","red"),cex=.6)
```



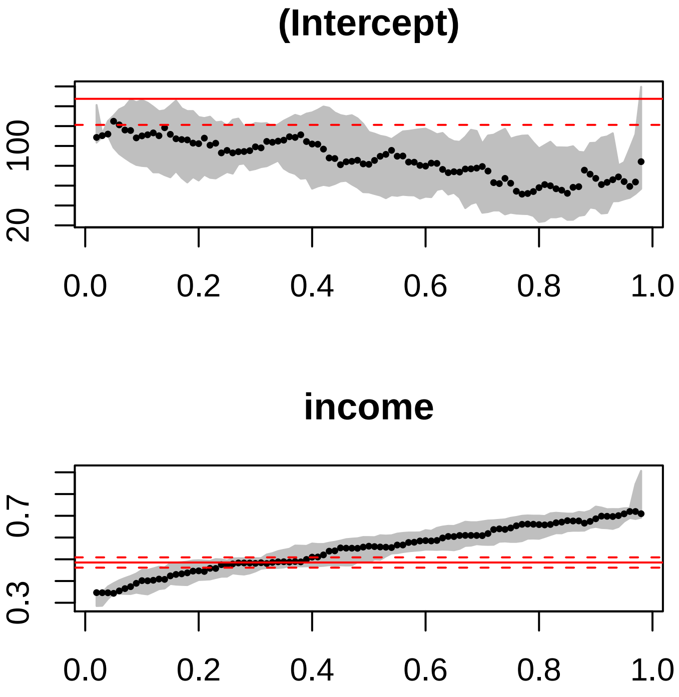
Quantile regression is just like L_1 regression with ρ_τ replacing $\rho_{.5}$ in (2), and with τ replacing 0.5 in the asymptotics.

Example. This example shows expenditures on food as a function of income for nineteenth-century Belgian households.

```
> data(engel)
> plot(foodexp~income,engel,cex=.5,xlab="Household Income", ylab="Food Expenditure")
> abline(rq(foodexp~income,data=engel,tau=.5),col="blue")
> taus <- c(.1,.25,.75,.90)
> for( i in 1:length(taus)){
+         abline(rq(foodexp~income,data=engel,tau=taus[i]),col="gray")
+         }
```



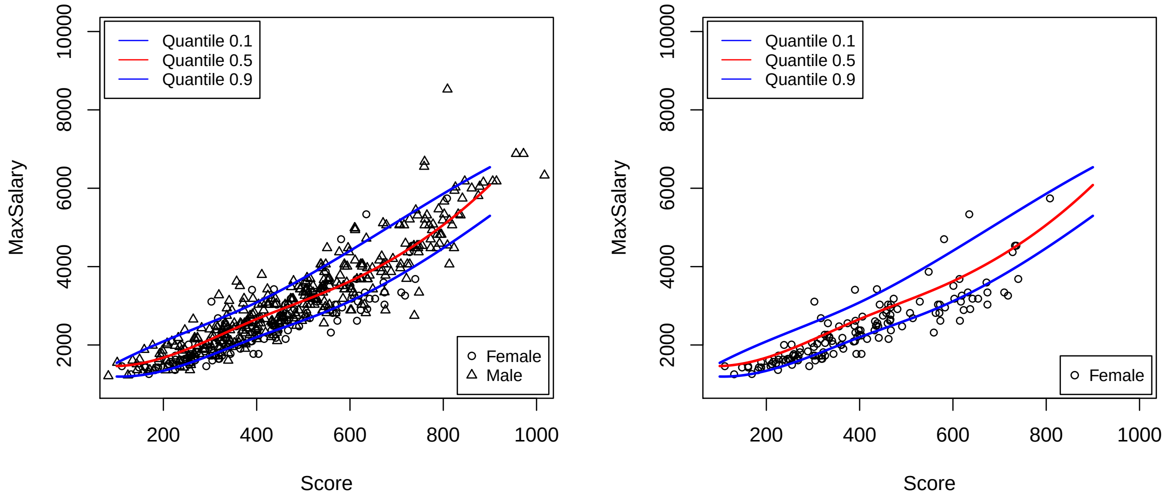
```
> plot(summary(rq(foodexp~income,data=engel,tau=2:98/100)))
```



(The horizontal line is the ols estimate, with the dashed lines for confidence interval for it.)

Second Example This example examines salary as a function of job difficulty for job classes in a large governmental unit. Points are marked according to whether or not the fraction of female employees in the class exceeds 80%.

```
> library(alr3)
> par(mfrow=c(1,2))
> mdom <- with(salarygov, NW/NE < .8)
> taus <- c(.1, .5, .9)
> cols <- c("blue", "red", "blue")
> x <- 100:900
> plot(MaxSalary ~ Score, salarygov, xlim=c(100, 1000), ylim=c(1000, 10000),
+         cex=0.75, pch=mdom + 1)
> for( i in 1:length(taus)){
+    lines(x, predict(rq(MaxSalary ~ bs(Score,5), data=salarygov[mdom, ], tau=taus[i]),
+           newdata=data.frame(Score=x)), col=cols[i],lwd=2)
+         }
> legend("topleft",paste("Quantile",taus),lty=1,col=cols,inset=.01, cex=.8)
> legend("bottomright",c("Female","Male"),pch=c(1, 2),inset=.01, cex=.8)
> plot(MaxSalary ~ Score, salarygov[!mdom, ], xlim=c(100, 1000), ylim=c(1000, 10000),
+         cex=0.75, pch=1)
> for( i in 1:length(taus)){
+    lines(x, predict(rq(MaxSalary ~ bs(Score,5), data=salarygov[mdom, ], tau=taus[i]),
+           newdata=data.frame(Score=x)), col=cols[i],lwd=2)
+         }
> legend("topleft",paste("Quantile",taus),lty=1,col=cols,inset=.01, cex=.8)
> legend("bottomright",c("Female"),pch=c(1),inset=.01, cex=.8)
```



References