Robust Regression in R

An Appendix to *An R Companion to Applied Regression, Second Edition*

John Fox & Sanford Weisberg

last revision: 15 December 2010

Abstract

Linear least-squares regression can be very sensitive to unusual data. In this appendix to Fox and Weisberg (2011), we describe how to fit several alternative robust-regression estimators, which attempt to down-weight or ignore unusual data: M-estimators; bounded-influence estimators; MM-estimators; and quantile-regression estimators, including $L1$ regression.

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 between the mid-1960s and the mid-1980s. With the exception of the L_1 methods described in Section 5, they are not widely used today.

1 Breakdown and Robustness

The finite-sample breakdown point of an estimator or procedure is the smallest fraction α of "bad" data values such that if the $[n\alpha]$ bad values $\to \infty$ then the estimator or procedure also becomes infinite. For example, the sample mean of $x_1, \ldots, x_n$ can be written as an explicit function of one of the observations in the sample as

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

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, the breakdown point cannot exceed 50%. (Reader: Why is that?)

2 *M*-Estimation

In linear regression, the breakdown of the ordinary least squares (OLS) estimator is analogous to the breakdown of the sample mean: A few extreme observations can largely determine the value of

the OLS estimator. In Fox and Weisberg (2011, Chap. 6), methods for detecting potentially influential points are presented, and these provide one approach to dealing with such points. Another approach is to replace ordinary least squares with an estimation method that is less affected by the outlying and influential points and can therefore produce useful results by accommodating the non-conforming data.

The most common general method of robust regression is *M-estimation*, introduced by Huber (1964). This class of estimators can be regarded as a generalization of maximum-likelihood estimation, hence the "*M*."

We consider only the linear model

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i \\ &= \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i \end{aligned}$$

for the ith of n independent observations. We assume that the model itself is not at issue, so $\mathrm{E}(y|\mathbf{x}) = \mathbf{x}_i'\boldsymbol{\beta}$, but the distribution of the errors may be heavy-tailed, producing occasional outliers. Given an estimator $\mathbf{b}$ for $\boldsymbol{\beta}$, the fitted model is

$$\widehat{y}_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_p x_{ip} + e_i = \mathbf{x}_i'\mathbf{b}$$

and the residuals are given by

$$e_i = y_i - \widehat{y}_i = y_i - \mathbf{x}_i'\mathbf{b}$$

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

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

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

- always-non negative, $\rho(e) \geq 0$;
- equal to 0 when its argument is 0, $\rho(0) = 0$;
- symmetric, $\rho(e) = \rho(-e)$, although in some problems one might argue that symmetry is undesirable; and
- 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.

2.1 Computing *M*-Estimates

The minimum value of Equation 1 can be found by differentiating with respect to the argument $\mathbf{b}$, and setting the resulting partial derivatives to 0:

$$\begin{aligned} \mathbf{0} &= \frac{\partial}{\partial \mathbf{b}} \sum_{i=1}^{n} \rho(y_i - \mathbf{x}_i'\mathbf{b}) \\ &= \sum_{i=1}^{n} \psi(y_i - \mathbf{x}_i'\mathbf{b})\mathbf{x}_i' \end{aligned}$$

where the *influence curve* ψ is defined to be the derivative of ρ with respect to its argument.

To facilitate computing, we would like to make this last equation similar to the estimating equations for a familiar problem like weighted least squares. To this end, define the *weight function* $w_i = w(e_i) = \psi(e_i)/e_i$. The estimating equations may then 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 $[\sum \psi'(e_i)/n]^2$ 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 ψ function of the bisquare estimator *redescends* to 0 for sufficiently large residuals.

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

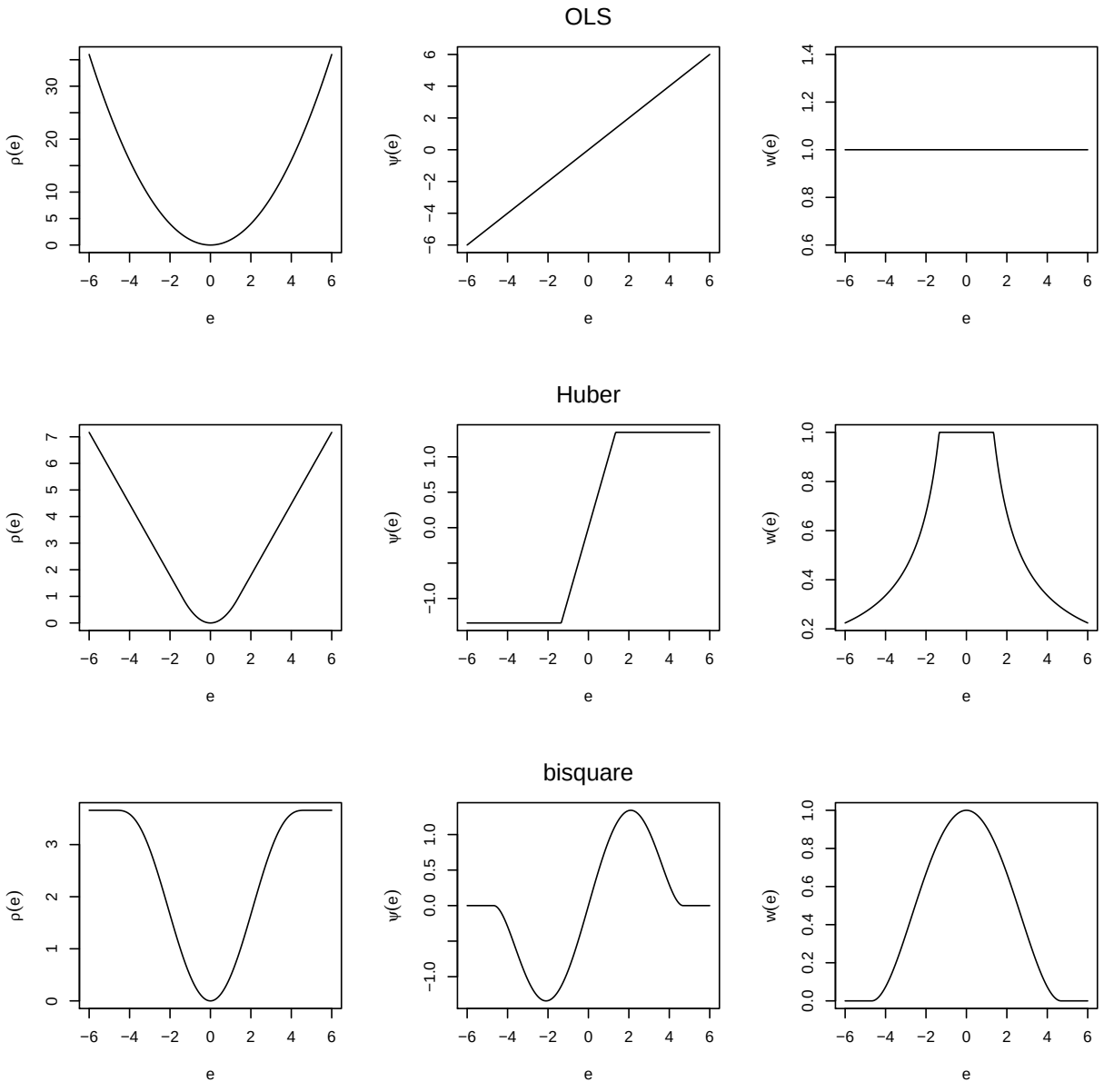

Figure 1: Objective (left), ψ (center), and weight (right) functions for the least-squares (top), Huber (middle), and bisquare (bottom) estimators. The tuning constants for these graphs are $k = 1.345$ for the Huber estimator and $k = 4.685$ for the bisquare. (One way to think about this scaling is that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

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 functions and weight functions for least-squares, Huber, and bisquare estimators.

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 employed 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.

3 Bounded-Influence Regression

Under certain circumstances, M-estimators can be vulnerable to high-leverage observations. Very-high-breakdown *bounded-influence estimators* for regression have been proposed and R functions for them are presented here. Very-high-breakdown estimates should be avoided, however, unless we have faith that the model we are fitting is correct, because these estimates do not allow for diagnosis of model misspecification (Cook et al., 1992).

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 (Rousseeuw and Leroy, 1987). Moreover, bounded-influence estimators can produce unreasonable results in certain circumstances (Stefanski, 1991), and there is no simple formula for coefficient standard errors.[1]

One application of bounded-influence estimators is to provide starting values for M-estimation. This procedure, along with using the bounded-influence estimate of the error variance, produces the so-called *MM-estimator*. The *MM*-estimator retains the high breakdown point of the bounded-influence estimator and shares the relatively high efficiency under normality of the traditional

[1]Statistical inference for the LTS estimator can be performed by bootstrapping, however. See Section 4.3.7 in the text and the Appendix on bootstrapping .

M-estimator. MM-estimators are especially attractive when paired with redescending ψ-functions such as the bisquare, which can be sensitive to starting values.

4 An Illustration: Duncan's Occupational-Prestige Regression

Duncan's occupational-prestige regression was introduced in Fox and Weisberg (2011, Chap. 1). The least-squares regression of `prestige` on `income` and `education` produces the following results:

```
> library(car) # for data
> 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
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 Fox and Weisberg (2011) that two observations, ministers and railroad conductors, serve to decrease the `income` coefficient 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** package:

```
> 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
```

The Huber regression coefficients are between those produced by the least-squares fit to the full data set and 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 employed to label all observations with weights less than 0.8:

```
> plot(mod.huber$w, ylab="Huber Weight")
> smallweights <- which(mod.huber$w < 0.8)
> showLabels(1:45, mod.huber$w, rownames(Duncan), id.method=smallweights, cex.=.6)

[1] "minister"        "reporter"        "conductor"       "contractor"
[5] "factory.owner"   "mail.carrier"    "insurance.agent" "store.clerk"
[9] "machinist"
```

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

The function `rlm` can also fit the bisquare estimator. As we explained, 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)

Call: rlm(formula = prestige ~ income + education, data = Duncan, method = "MM")
Residuals:
   Min     1Q Median     3Q    Max
```

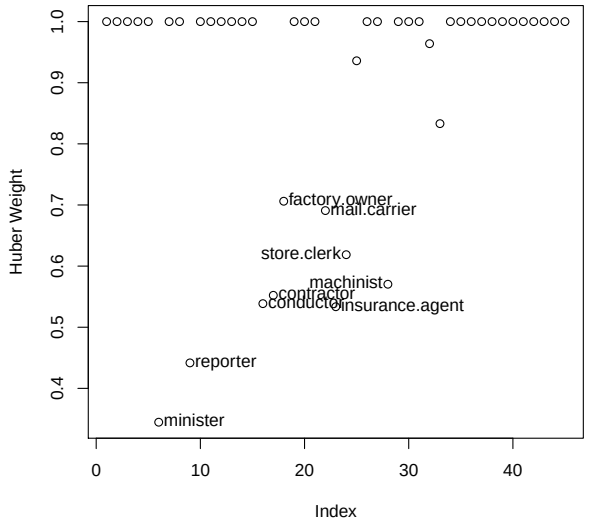

Figure 2: Weights from the robust Huber estimator for the regression of `prestige` on `income` and `education`.

```
-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, 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 < 0.8), cex.=0.6)

 [1] "minister"           "reporter"           "conductor"
 [4] "contractor"         "factory.owner"      "mail.carrier"
 [7] "insurance.agent"    "store.clerk"        "machinist"
[10] "streetcar.motorman"
```

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

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

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

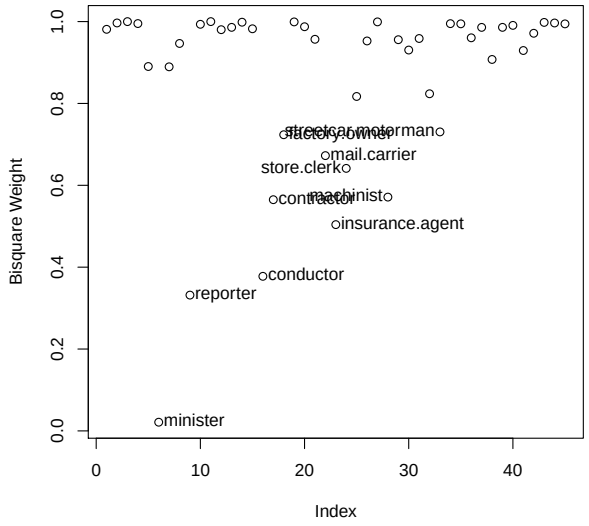

Figure 3: Weights from the robust bisquare estimator for the regression of `prestige` on `income` and `education`.

```
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 Koenker (2005) and the vignette for quantile regression in the **quantreg** package in R. We start by assuming a model like this:

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i \tag{2}$$

where the ε_i are random errors. We estimate $\boldsymbol{\beta}$ by solving the minimization problem

$$\widetilde{\boldsymbol{\beta}} = \arg\min \frac{1}{n}\sum_{i=1}^{n} \left|y_i - \mathbf{x}_i'\boldsymbol{\beta}\right| = \frac{1}{n}\sum_{i=1}^{n} \rho_{.5}(y_i - \mathbf{x}_i'\boldsymbol{\beta}) \tag{3}$$

If the ε_i are independent and identically distributed from a double exponential distribution, then $\widetilde{\boldsymbol{\beta}}$ is the maximum likelihood estimate for $\boldsymbol{\beta}$. In general L_1 regression estimates the *median* y at $\mathbf{x}_i'\boldsymbol{\beta}$, so one can think of this as *median regression*.

We begin with a simple simulated example with N_1 "good" observations and N_2 "bad" ones:

```
> set.seed(10131986) # for reproducibility
> 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=substitute(list(N[1] == n1, N[2] == n2), list(n1=n1, n2=n2)))
+   summary(r1 <-rq(Y ~ X, data=data, tau=0.5))
+   abline(r1, lwd=2)
+   abline(lm(Y ~ X, data), lty=2, lwd=2, col="red")
+   abline(lm(Y ~ X, data, subset=1:n1), lty=3, lwd=2, col="blue")
+   legend("topleft", c("L1","OLS","OLS on good"),
+          inset=0.02, lty=1:3, lwd=2, 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)
```

In Figure 4, all four panels have $N_1 = 100$ observations sampled from a bivariate normal distribution with means $(0, 0)'$, variances $(1, 1)'$, and correlation 0.9. In addition, N_2 observations are sampled from a bivariate normal distribution with means $(1.5, -1.5)$ and covariance matrix $\sqrt{2}\mathbf{I}_2$. The value of N_2 varies from panel to panel. In each panel, three regression lines are shown: OLS fit to the N_1 good data points; OLS fit to all the data; and median regression fit to all the data. If the goal is to match, more or less, the OLS regression fit to the good data, then the median regression does a respectable jobs for $N_2 \leq 30$, but it does no better than OLS on all the data for larger N_2. Of course in these latter cases the distinction between "good" and "bad" data is hard to justify.

5.1 Comparing L_1 and L_2 (OLS) Regression

L_1 regression minimizes the sum of the absolute errors while L_2, another name for ordinary least squares, minimizes squared errors. Consequently, L_1 *gives much less weight to large deviations.* The ρ-functions for L_1 and L_2 are shown in Figure 5:

```
> curve(abs(x),-2, 2, lwd=2, ylab=expression(rho(x)))
> curve(x^2, -3, 3, lty=2, lwd=2, add=TRUE, col="red")
> abline(h=0, lty=3)
> abline(v=0, lty=3)
> legend("bottomleft", inset=.05, legend=c("L1", "L2"), lty=1:2,
+     cex=0.75, col=c("black", "red"))
```

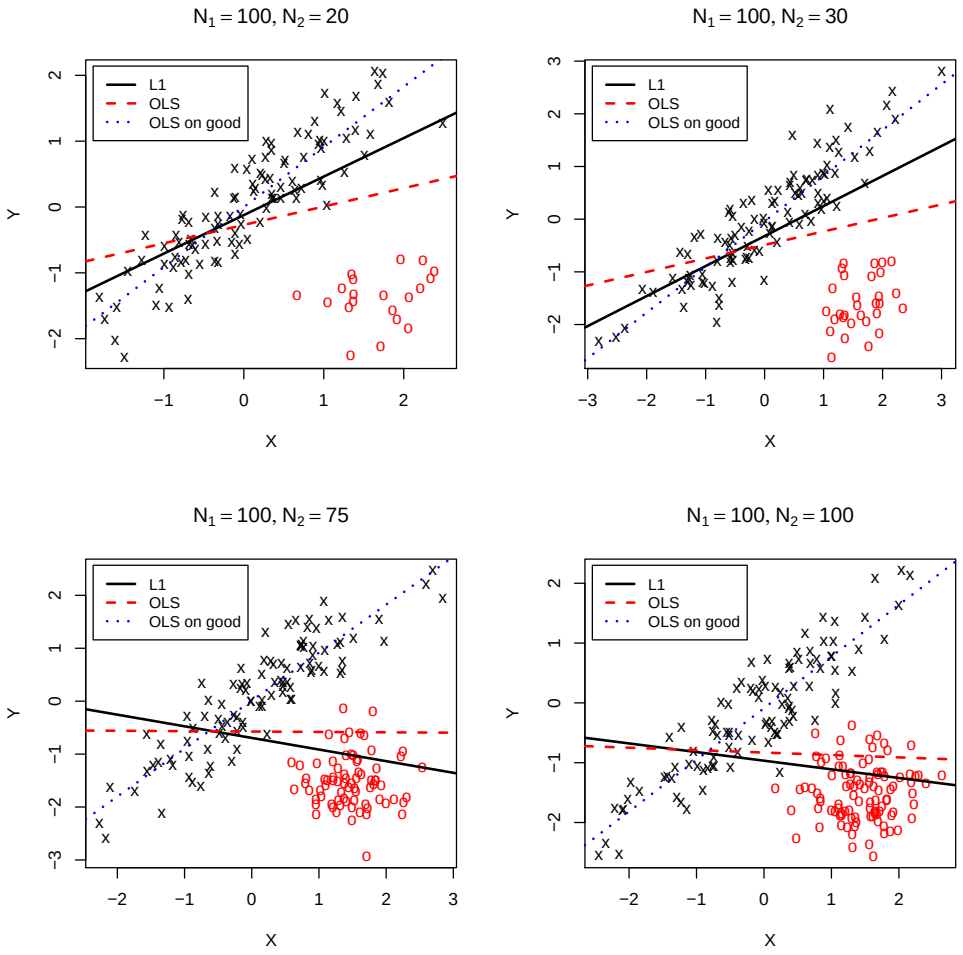

Figure 4: Simulated data with an increasing number of "bad" or "outlying" observations. Three lines are shown in each panel: the $L1$ regression (solid black line); OLS fit to all of the data (broken red line); OLS fit to the "good" data points (dotted blue line).

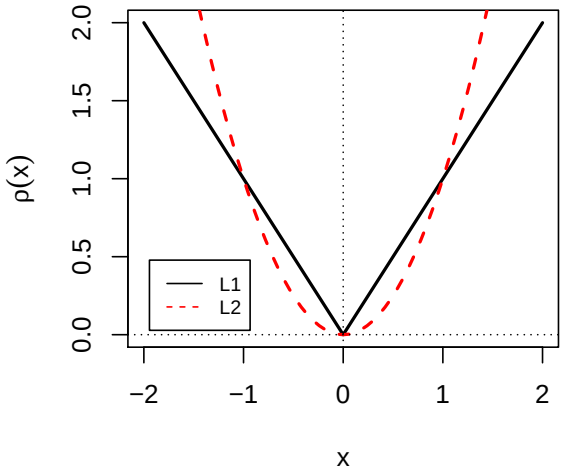

Figure 5: ρ-functions for L_1 and L_2 regressions

5.2 L_1 Facts

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

2. In Equation 2 (page 9) if $\mathbf{x}$ consists only of the constant regressor (1), then the L_1 estimator is the median.

3. Computations are not nearly as easy as for least squares, because a linear programming solution is required for $L1$ regression.

4. If the $n \times p$ model matrix $\mathbf{X} = (\mathbf{x}_1', \ldots, \mathbf{x}_n')'$ is of full column rank $p+1$, and if $\mathbf{h}$ is a set that indexes exactly $p+1$ of the rows of $\mathbf{X}$, then there is always an $\mathbf{h}$ such that the L_1 estimate $\widetilde{\boldsymbol{\beta}}$ fits these $p+1$ points exactly, so $\widetilde{\boldsymbol{\beta}} = (\mathbf{X}_\mathbf{h}'\mathbf{X}_\mathbf{h})^{-1}\mathbf{X}_\mathbf{h}'\mathbf{y}_\mathbf{h} = \mathbf{X}_\mathbf{h}^{-1}\mathbf{y}_\mathbf{h}$. 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 $\mathbf{y}$ by $a + b\mathbf{y}$ and $\mathbf{X}$ by $\mathbf{A} + \mathbf{B}^{-1}\mathbf{X}$ (where a, b, $\mathbf{A}$, and $\mathbf{B}$ are constants) will leave the solution essentially unchanged.

6. The breakdown point 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 L_1 regression estimates the median of $y|\mathbf{x}$, not the conditional mean.

8. Suppose we have Equation 2 (page 9) with the errors independent and identically distributed 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 $\widehat{\xi}_{.5} = \widehat{F}^{-1}(\tau)$. We assume a standardized version of f so $f(u) = (1/\sigma)f_0(u/\sigma)$. Write $\mathbf{Q}_n = n^{-1}\sum \mathbf{x}_i\mathbf{x}_i'$, and suppose that in large samples $\mathbf{Q}_n \to \mathbf{Q}_0$, a fixed matrix. We will then have
$$\sqrt{n}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim \mathrm{N}(\mathbf{0}, \omega\mathbf{Q}_0^{-1})$$

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, $f[F_0^{-1}(\tau)] = 1/\sqrt{2\pi} = 0.399$, and $\sqrt{\omega} = 0.5\sigma/0.399 = 1.26\sigma$, 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 tests and confidence intervals could 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

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

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

10. For non-independent and identically distributed errors, suppose that $\xi_i(\tau)$ is the τ-quantile for the distribution of the ith error. One can show that

$$\sqrt{n}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim \mathrm{N}\left[\mathbf{0}, \tau(1-\tau)\mathbf{H}^{-1}\mathbf{Q}_0\mathbf{H}^{-1}\right]$$

where the matrix $\mathbf{H}$ is given by

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

and thus a sandwich-type estimator is used for estimating the variance of $\widetilde{\boldsymbol{\beta}}$. The `rq` function in the **quantreg** package uses a sandwich formula by default for computing coefficient standard errors.

6 Quantile regression

6.1 Sample and Population Quantiles

For a sample $x_1, \ldots, x_n$, for any $0 < \tau < 1$ the τth sample quantile is the smallest value that exceeds $\tau \times 100\%$ of the data. In a population with distribution F, we define the τth population quantile to be the solution to

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

L_1 is a special case of *quantile regression* in which we minimize the $\tau = .50$-quantile, but a similar calculation can be performed for any $0 < \tau < 1$, where the objective function $\rho_\tau(u)$ is called in this instance a *check function*,

$$\rho_\tau(u) = u \times [\tau - I(u < 0)] \tag{4}$$

where I is the indicator function (more on check functions later). Figure 6 shows the check function in Equation 4 for $\tau \in \{.25, .5, .9\}$:

```
> rho <- function(u) {
+  u * (tau - ifelse(u < 0,1,0)  )
+ }
```

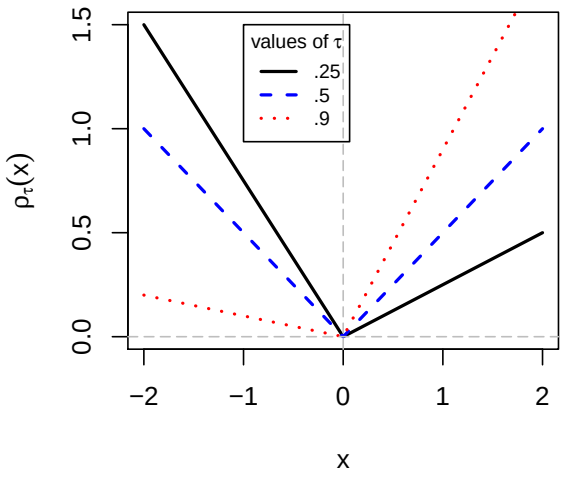

Figure 6: Check function for three values of τ for quantile regression. For $\tau = 0.5$, positive and negative errors are treated symmetrically, but for the other values of τ, positive and negative errors are treated asymmetrically.

```
> tau <- .25; curve(rho, -2, 2, lty=1, lwd=2, ylab=expression(rho[tau](x)))
> tau <- .50; curve(rho, -2, 2, lty=2, col="blue", add=TRUE, lwd=2)
> tau <- .90; curve(rho, -2, 2, lty=3, col="red", add=TRUE, lwd=2)
> abline(v=0, h=0, lty=5, col="gray")
> legend(-1, 1.5, c(".25", ".5", ".9"), lty=1:3, lwd=2, cex=0.75,
+        col=c("black" ,"blue" ,"red"), title=expression("values of"~tau))
```

Quantile regression is just like L_1 regression with ρ_τ replacing $\rho_{.5}$ in Equation 3 (page 9), and with τ replacing 0.5 in the asymptotics.

6.2 Example: Salary Data

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%. The data are shown in Figure 7. Because the dependence of the response on the predictor is apparently curved, we model the response with a 5-*df* B-spline, using the model formula `MaxSalary ~ bs(Score, 5)`. We will estimate the median regression, as well as the 0.10 and 0.90 quantile regressions:

```
> library(alr3) # for data
> library(quantreg)
> fdom <- with(salarygov, NW/NE > .8)
> taus <- c(.1, .5, .9)
> ltys <- c(2, 1, 2)
> cols <- c("blue", "red", "blue")
> x <- 100:1000
```

```
> plot(MaxSalary ~ Score, data=salarygov,
+     xlim=c(100, 1000), ylim=c(1000, 10000),
+     pch=c(2, 16)[fdom + 1], col=c("black", "green")[fdom + 1])
> mods <- rq(MaxSalary ~ bs(Score, 5), tau=c(.1, .5, .9),
+            data=salarygov[!fdom, ])
> mods

Call:
rq(formula = MaxSalary ~ bs(Score, 5), tau = c(0.1, 0.5, 0.9),
    data = salarygov[!fdom, ])

Coefficients:
              tau= 0.1 tau= 0.5 tau= 0.9
(Intercept)     1207.0   1507.3   1466.5
bs(Score, 5)1   -100.9   -151.9    437.4
bs(Score, 5)2    779.9    974.1   1300.7
bs(Score, 5)3   2010.6   2255.9   3176.0
bs(Score, 5)4   3724.4   3822.2   5010.0
bs(Score, 5)5   5122.0   6147.1   5733.8

Degrees of freedom: 357 total; 351 residual

> predictions <- predict(mods, data.frame(Score=x))
> for( j in 1:3) lines(x, predictions[, j], col=cols[j], lty=ltys[j], lwd=2)
> legend("topleft", legend=taus, title="Quantile", lty=ltys, lwd=2,
+     col=cols, inset=0.01)
> legend("bottomright", legend=c("Non-Female-Dominated","Female-Dominated"),
+     pch=c(2, 16), inset=0.01, col=c("black", "green"))
```

We begin by defining an indicator variable for the emale-dominated job classes, and a vector for the τs . We will graph the non-female-dominated classes in black and the female-dominated classes in green. The quantile regression is fit using the `rq` function in the **quantreg** package. Its arguments are similar to those for `lm` except for a new argument for setting `tau`; the default is `tau=0.5` for L_1 regression, and here we specify three values of τ. The fitted coefficients for the B-splines are then displayed, and although these are not easily interpretable, the important point is that they are different for each value of τ. The `predict` function returns a matrix with three columns, one for each τ, and we use these values to add fitted regression lines to the graph. We fit the model to the non-female-dominated occupations only, as is common is gender-discrimination studies.

The quantile regressions are of interest here to describe the variation in the relationship between salary and score in the non-female-dominated job classes. Most of the female-dominated classes fall below the median line and many below the 0.1-quantile. For extreme values of `Score` the more extreme quantiles are very poorly estimated, which accounts for the crossing of the median and the 0.9 estimated quantiles for large values of `Score`.

6.3 Duncan's Data

Quantile regression can also be used for multiple regression. For example, to compute the $L1$ regression for Duncan's occupational-prestige data:

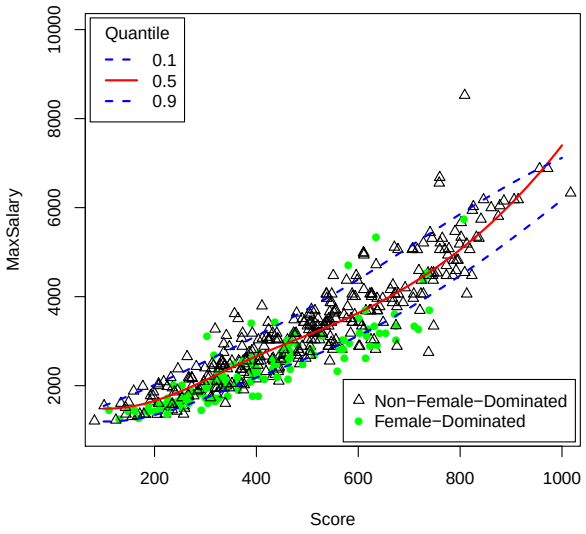

Figure 7: Quantile regressions fit to non-female-dominated job classes.

Method	b_0	b_1 (`income`)	b_2 (`education`)
OLS	−6.0647	0.5987	0.5458
OLS removing obs. 6 & 16	−6.4090	0.8674	0.3322
Huber M-estimator	−7.1107	0.7014	0.4854
bisquare MM-estimator	−7.3886	0.7825	0.4233
LTS estimator	−7.0145	0.8045	0.4318
$L1$ estimator	−6.4083	0.7477	0.4587

Table 2: Various estimators of Duncan's occupational-prestige regression.

```
> mod.quant <- rq(prestige ~ income + education, data=Duncan)
> summary(mod.quant)

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

tau: [1] 0.5

Coefficients:
            coefficients lower bd upper bd
(Intercept)  -6.4083     -12.4955  -3.6003
income        0.7477       0.4719   0.9117
education     0.4587       0.2195   0.6610
```

The `summary` method for `rq` objects reports 95-percent confidence intervals for the regression coefficients; it is also possible to obtain coefficient standard errors (see `?summary.rq`). The $L1$ estimates here are very similar to the M-estimates based on Huber's weight function. Table 2 summarizes the various estimators that we applied Duncan's regression.

7 Complementary Reading and References

Robust regression is described in Fox (2008, Chap. 19). Koenker (2005) provides an extensive treatment of quantile regression. A recent mathematical treatment of robust regression is given by Huber and Ronchetti (2009). Andersen (2007) provides an introduction to the topic.

References

Andersen, R. (2007). *Modern Methods for Robust Regression*. Sage, Thousand Oaks, CA.

Cook, R. D., Hawkins, D. M., and Weisberg, S. (1992). Comparison of model misspecification diagnostics using residuals from least mean of squares and least median of squares fits. *Journal of the American Statistical Association*, 87(418):pp. 419–424.

Fox, J. (2008). *Applied Regression Analysis and Generalized Linear Models*. Sage, Thousand Oaks, CA, second edition.

Fox, J. and Weisberg, S. (2011). *An R Companion to Applied Regression*. Sage, Thousand Oaks, CA, second edition.

Huber, P. and Ronchetti, E. M. (2009). *Robust Statistics*. Wiley, Hoboken NJ, second edition.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):pp. 73–101.

Koenker, R. (2005). *Quantile regression*. Cambridge University Press, Cambridge.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust regression and outlier detection*. Wiley, Hoboken, NJ.

Stefanski, L. (1991). A note on high-breakdown estimators. *Statistics & Probability Letters*, 11(4):353 – 358.