The Stata Journal (2009)
9, Number 3, pp. 439–453

Robust regression in Stata

Vincenzo Verardi[1]
University of Namur (CRED)
and Université Libre de Bruxelles (ECARES and CKE)
Rempart de la Vierge 8, B-5000
Namur, Belgium
vverardi@fundp.ac.be

Christophe Croux
K. U. Leuven, Faculty of Business and Economics
Naamsestraat 69, B-3000
Leuven, Belgium
christophe.croux@econ.kuleuven.be

Abstract. In regression analysis, the presence of outliers in the dataset can strongly distort the classical least-squares estimator and lead to unreliable results. To deal with this, several robust-to-outliers methods have been proposed in the statistical literature. In Stata, some of these methods are available through the `rreg` and `qreg` commands. Unfortunately, these methods resist only some specific types of outliers and turn out to be ineffective under alternative scenarios. In this article, we present more effective robust estimators that we implemented in Stata. We also present a graphical tool that recognizes the type of detected outliers.

Keywords: st0173, mmregress, sregress, msregress, mregress, mcd, S-estimators, MM-estimators, outliers, robustness

1 Introduction

The objective of linear regression analysis is to study how a dependent variable is linearly related to a set of regressors. In matrix notation, the linear regression model is given by

$$y = X\theta + \varepsilon$$

where, for a sample of size n, y is the $n \times 1$ vector containing the values for the dependent variable, X is the $n \times p$ matrix containing the values for the p regressors, and ε is the $n \times 1$ vector containing the error terms. The $p \times 1$ vector θ contains the unknown regression parameters and needs to be estimated. On the basis of the estimated parameter $\widehat{\theta}$, it is then possible to fit the dependent variable by $\widehat{y} = X\widehat{\theta}$ and compute the residuals $r_i = y_i - \widehat{y}_i$ for $i = 1 \leq i \leq n$. Although θ can be estimated in several ways, the underlying idea is always to try to get as close as possible to the true value by reducing the magnitude of the residuals, as measured by an aggregate prediction error. For the

1. Vincenzo Verardi is an associated researcher of the FNRS and gratefully acknowledges their financial support.

well-known ordinary least squares (OLS), this aggregate prediction error is defined as the sum of squared residuals. The vector of parameters estimated by OLS is then

$$\widehat{\theta}_{\text{OLS}} = \arg\min_{\theta} \sum_{i=1}^{n} r_i^2(\theta)$$

with $r_i(\theta) = y_i - \theta_0 - \theta_1 X_{i1} - \cdots - \theta_p X_{ip}$ for $1 \leq i \leq n$. This estimation can be performed in Stata by using the `regress` command. A drawback of OLS is that by considering squared residuals, it tends to award an excessive importance to observations with very large residuals and, consequently, distort parameters' estimation in case of the existence of outliers.

The scope of this article is, first, to describe regression estimators that are robust with respect to outliers and, second, to propose Stata commands to implement them in practice. The structure of the article is the following: in the next section, we briefly present the types of outliers that can be found in regression analysis and introduce the basics of robust regression. We recommend using estimators with a high breakdown point, which are known to be resistant to outliers of different types. In section 3, we describe them and provide a sketch of the Stata code we implemented to estimate them in practice. In section 4, we give an example using the well-known Stata `auto.dta` dataset. In section 5, we provide some simulation results to illustrate how the estimators with a high breakdown point outperform the robust estimators available in Stata. Finally, in section 6, we conclude.

2 Outliers and robust regression estimators

In regression analysis, three types of outliers influence the OLS estimator. Rousseeuw and Leroy (2003) define them as *vertical outliers*, *bad leverage points*, and *good leverage points*. To illustrate this terminology, consider a simple linear regression as shown in figure 1 (the generalization to higher dimensions is straightforward). Vertical outliers are those observations that have outlying values for the corresponding error term (the y dimension) but are not outlying in the space of explanatory variables (the x dimension). Their presence affects the OLS estimation and, in particular, the estimated intercept. Good leverage points are observations that are outlying in the space of explanatory variables but that are located close to the regression line. Their presence does not affect the OLS estimation, but it affects statistical inference because they do deflate the estimated standard errors. Finally, bad leverage points are observations that are both outlying in the space of explanatory variables and located far from the true regression line. Their presence significantly affects the OLS estimation of both the intercept and the slope.

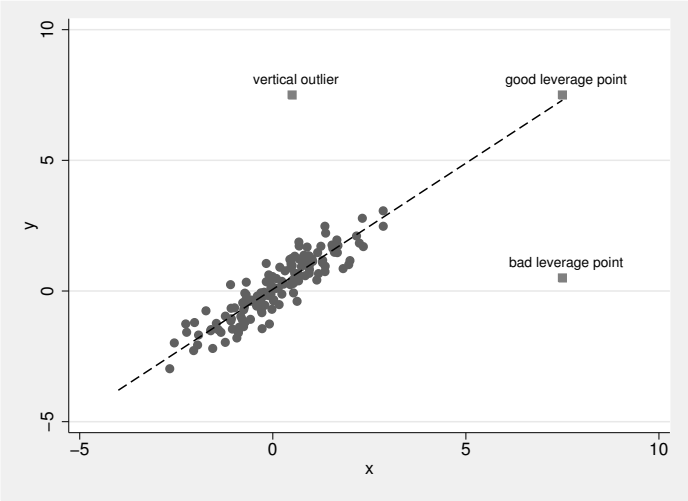

Figure 1. Outliers in regression analysis

Edgeworth (1887) realized that because of the squaring of the residuals, OLS becomes extremely vulnerable to the presence of outliers. To cope with this, he proposed a method consisting of minimizing the sum of the absolute values of the residuals rather than the sum of their squares. More precisely, his method defines the L_1, or *median regression*, estimator as

$$\widehat{\theta}_{L_1} = \arg\min_{\theta} \sum_{i=1}^{n} \mid r_i(\theta) \mid \tag{1}$$

The median regression estimator is available with Stata's offical `qreg` command. This estimator protects against vertical outliers but not against bad leverage points. It has an efficiency of only 64% at a Gaussian error distribution (see Huber [1981]).

Huber (1964) generalized median regression to a wider class of estimators, called M-estimators, by considering functions other than the absolute value in (1). This allows an increase in Gaussian efficiency while keeping robustness with respect to vertical outliers. An M-estimator is defined as

$$\widehat{\theta}_{M} = \arg\min_{\theta} \sum_{i=1}^{n} \rho\left\{\frac{r_i(\theta)}{\sigma}\right\} \tag{2}$$

where $\rho(\cdot)$ is a loss function, which is even, nondecreasing for positive values and less increasing than the square function. To guarantee scale equivariance (i.e., independence with respect to the measurement units of the dependent variable), residuals are standardized by a measure of dispersion σ. M-estimators are called monotone if $\rho(\cdot)$ is convex over the entire domain and redescending if $\rho(\cdot)$ is bounded.

The practical implementation of M-estimators uses an iteratively reweighted OLS algorithm. To simplify, suppose that σ is known, and define weights as $\omega_i = \rho(r_i/\sigma)/r_i^2$. Then (2) can be rewritten as

$$\widehat{\theta}_M = \arg\min_{\theta} \sum_{i=1}^{n} \omega_i r_i^2(\theta)$$

which is a weighted OLS estimator. The weights ω_i are, however, a function of θ and are thus unknown. Using an initial estimate $\widetilde{\theta}$ for θ, the weights can be computed and serve as the start of an iteratively reweighted OLS algorithm. Unfortunately, the latter is guaranteed to converge to the global minimum of (2) only for monotone M-estimators, which are not robust with respect to bad leverage points.

In Stata, the `rreg` command computes a highly efficient M-estimator. The loss function used is the Tukey biweight function defined as

$$\rho(u) = \begin{cases} 1 - \left\{1 - \left(\frac{u}{k}\right)^2\right\}^3 & \text{if } |\,u\,| \le k \\ 1 & \text{if } |\,u\,| > k \end{cases} \tag{3}$$

where $k = 4.685$. The starting value of the iterative algorithm $\widetilde{\theta}$ is taken to be a monotone M-estimator with a Huber $\rho(\cdot)$ function:

$$\rho(u) = \begin{cases} \frac{1}{2}(u)^2 & \text{if } |\,u\,| \le c \\ c\,|\,u\,| - \frac{1}{2}c^2 & \text{if } |\,u\,| > c \end{cases}$$

where $c = 1.345$. Moreover, to give protection against bad leverage points, observations associated with Cook distances larger than 1 receive a weight of zero. A command (`mregress`) to compute a standard monotone M-estimator with a Huber $\rho(\cdot)$ function is described in section 6.

Unfortunately, the `rreg` command does not have the expected robustness properties for two main reasons. First, Cook distances only manage to identify isolated outliers and are inappropriate when clusters of outliers exist, where one outlier can mask the presence of another (see Rousseeuw and van Zomeren [1990]). It can therefore not be guaranteed to have identification of all leverage points. Second, the initial values for the iteratively reweighted OLS algorithm are monotone M-estimators that are not robust to bad leverage points and that may lead the algorithm to converge to a local instead of a global minimum.

3 Estimators with a high breakdown point

Full robustness can be achieved by tackling the regression problem from a different perspective. The OLS estimator is based on the minimization of the variance of the residuals. Hence, because the variance is highly sensitive to outliers, OLS is largely influenced as well. For this reason, Rousseeuw and Yohai (1984) propose to minimize

a measure of dispersion of the residuals that is less sensitive to extreme values than the variance.[2] They call this class of estimators the S-estimators. The intuition behind the method is simple. For OLS, the objective is to minimize the variance, $\widehat{\sigma}^2$, of the residuals. The latter can be rewritten as $1/n \sum_{i=1}^{n} \left(r_i/\widehat{\sigma}\right)^2 = 1$. As stated previously, the square value can be damaging because it gives a huge importance to large residuals. Thus, to increase robustness, the square function could be replaced by another loss function, ρ, that awards less importance to large residuals.[3] The estimation problem would now consist of finding the smallest robust scale of the residuals. This robust dispersion, denoted by $\widehat{\sigma}^S$, satisfies

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left\{\frac{r_i(\theta)}{\widehat{\sigma}^S}\right\} = b \tag{4}$$

where $b = E\{\rho(Z)\}$ with $Z \sim N(0,1)$. The value of θ that minimizes $\widehat{\sigma}^S$ is then called an S-estimator. More formally, an S-estimator is defined as

$$\widehat{\theta}^S = \arg\min_{\theta} \ \widehat{\sigma}^S \left\{r_1(\theta), \ldots, r_n(\theta)\right\} \tag{5}$$

where $\widehat{\sigma}^S$ is the robust estimator of scale as defined in (4).

The choice of $\rho(\cdot)$ is crucial to have good robustness properties and a high Gaussian efficiency. The Tukey biweight function defined in (3), with $k = 1.547$, is a common choice. This S-estimator resists contamination of up to 50% of outliers; it is said to have a breakdown point of 50%. Unfortunately, this S-estimator has a Gaussian efficiency of only 28.7%. If $k = 5.182$, the Gaussian efficiency rises to 96.6%, but the breakdown point drops to 10%. To cope with this, Yohai (1987) introduced MM-estimators that combine a high breakdown point and a high efficiency. These estimators are redescending M-estimators as defined in (2), but with the scale fixed at $\widehat{\sigma}^S$. So an MM-estimator is defined as

$$\widehat{\theta}^{MM} = \arg\min_{\theta} \sum_{i=1}^{n} \rho\left\{\frac{r_i(\theta)}{\widehat{\sigma}^S}\right\} \tag{6}$$

The preliminary S-estimator guarantees a high breakdown point, and the final MM-estimate guarantees a high Gaussian efficiency. It is common to use a Tukey biweight $\rho(\cdot)$ function for both the preliminary S-estimator and the final MM-estimator. The tuning constant k can be set to 1.547 for the S-estimator to guarantee a 50% breakdown point, and it can be set to 4.685 for the second-step MM-estimator in (6) to guarantee a 95% efficiency of the final estimator.

2. The least trimmed squares estimator and the least median squares estimator, introduced by Rousseeuw (1984) rely on the same logic. We programmed these two estimators in Stata and made them available through the `ltsregress` and `lmsregress` commands. `ltsregress` and `lmsregress` are available from the authors upon request.
3. As before, $\rho(\cdot)$ is a function that is even, nondecreasing for positive values, less increasing than the square with a unique minimum at zero.

For computing the MM-estimator, the iteratively reweighted OLS algorithm can be used, taking $\widehat{\theta}^S$ as its initial value. Once the initial S-estimate is computed, $\widehat{\theta}^{MM}$ comes at almost no additional computational cost. We programmed an S- and an MM-estimator in Stata (with Tukey biweight loss function) using the algorithm of Salibian-Barrera and Yohai (2006). Explicit formulas for the estimators are not available, and it is necessary to call on numerical optimization to compute them. We present, in the next section, a sketch of the algorithm we implemented in Stata. The commands to compute S- and MM-estimators (called `sregress` and `mmregress`, respectively) are described in section 6.

3.1 S-estimator and MM-estimator algorithms

The algorithm implemented in Stata for computing the S-estimator starts by randomly picking N subsets of p observations (defined as p-subset), where p is the number of regression parameters to estimate. For each p-subset, the equation of the hyperplane that fits all points perfectly is obtained, yielding a trial solution of (5). This trial value is more reliable if all p points are regular observations, such that the p-subset does not contain outliers. The number N of subsamples to generate is chosen to guarantee that at least one p-subset without outliers is selected with high probability. As shown in Salibian-Barrera and Yohai (2006), this can be achieved by taking

$$N = \left\lceil \frac{\log(1 - P_{\text{clean}})}{\log\{1 - (1 - \alpha)^p\}} \right\rceil \tag{7}$$

where α is the (maximal) expected proportion of outliers, p is the number of parameters to estimate, and P_{clean} is the desired probability to have at least one p-subset without outliers among the N subsamples.[4]

For each of the p-subsets, a hyperplane that perfectly fits the p-subset is computed. Then, for all n observations in the sample, residuals with respect to this hyperplane are computed, and a scale estimate, $\widehat{\sigma}^S$, is computed from them as in (4). In this way, scale estimates are obtained for each p-subset, and an approximation for the final scale estimate, $\widehat{\sigma}^S$, is then given by the trial value that leads to the smallest scale over all p-subsets. This approximation can be improved further by carrying some refinement steps that bring the approximation even closer to the solution of (5).

This algorithm is implemented in Stata and can be called either directly using the `sregress` command or indirectly using the `mmregress` command and invoking the `initial` option. Once the S-estimator is obtained, the MM-estimator directly follows by applying the iteratively reweighted OLS algorithm up to convergence. We provide a Stata command for MM-estimators through the `mmregress` command. As far as inference is concerned, standard errors robust to heteroskedasticity (and asymmetric errors) are computed according to the formulas available in the literature (see, e.g., Croux, Dhaene, and Hoorelbeke [2008]).

4. The default values we use in the implementation of the algorithm are $\alpha = 0.2$ and $P_{\text{clean}} = 0.99$.

The need of calling on subsampling algorithms becomes the Achilles' heel of the algorithm when several dummy variables are present. Indeed, as stated by Maronna and Yohai (2000), subsampling algorithms can easily lead to collinear subsamples if various dummies are among the regressors. To cope with this, Maronna and Yohai (2000) introduce the MS-estimator that alternates an S-estimator (for continuous variables) and an M-estimator (for dummy ones) till convergence. This estimator is out of the scope of this article, and we thus do not elaborate on it here. We nevertheless briefly describe the Stata command implemented to compute it in practice (`msregress`). This estimator can be particularly helpful in the fixed-effects panel-data models, as suggested by Bramati and Croux (2007).

3.2 Outlier detection

In addition to reducing the importance of outliers on the estimator, robust statistics are also intended to identify atypical individuals. Once identified, they could be analyzed separately from the bulk of the data. To do so, it is important to recognize their type. This can be easily achieved by calling on the graphical tool proposed by Rousseeuw and van Zomeren (1990). This graphical tool is constructed by plotting, on the vertical axis, the robust standardized residuals, defined as $r_i/\widehat{\sigma}^S$, with $r_i \equiv r_i(\widehat{\theta}^S)$, to give an idea of outlyingness with respect to the fitted regression plane. On the horizontal axis, a measure of the (multivariate) outlyingness of the explanatory variables is plotted. The latter is measured by Mahalanobis distance defined as $d_i = \sqrt{(X_i - \mu)\Sigma^{-1}(X_i - \mu)'}$, where μ is the multivariate location vector, Σ is the covariance matrix of the explanatory variables, and X_i is the ith row vector of matrix X, for $1 \leq i \leq n$. Obviously, both μ and Σ should be estimated robustly if we want these distances to resist the presence of outliers. Several methods have been proposed to robustly estimate the Mahalanobis distances. In Stata, the `hadimvo` command is available, but more robust estimates for the covariance matrix (such as the minimum covariance determinant estimator) are also available. We briefly describe the command (`mcd`) to compute the minimum covariance determinant in section 6.

It is possible to set the limits outside which individuals can be considered as outliers. For the y dimension, we set them to -2.25 and $+2.25$. These represent the values of the standard normal that separate the 2.5% remotest area of the distribution from the central mass. For the x dimension, we set the limit to $\sqrt{\chi^2_{p,0.975}}$, motivated by the fact that the squared Mahalanobis distance is χ^2_p distributed under normality.

4 Example

To illustrate the usefulness of the robust methods, we present an example based on the well-known Stata `auto.dta` dataset. More specifically, we regress the price of cars on the following set of characteristics: the mileage (mpg), the headroom (in.), the trunk space (cu. ft.), the length (in.), the weight (lbs.), the turn circle (ft.), the displacement (cu. in.), the gear ratio, four dummies identifying the categorical variable repair record

in 1978, and a foreign dummy identifying whether the car was built in the United States. We first identify outliers. For this purpose, we call on the graphical tool described in section 3.2. The resulting plot is pictured in figure 2. This can be easily replicated by typing the following Stata commands (which are described more precisely in section 6).

```
. use http://www.stata-press.com/data/r11/auto
(1978 Automobile Data)
. xi: mmregress price mpg headroom trunk length weight turn displacement
> gear_ratio foreign i.rep78, outlier graph label(make)
```

(*output omitted*)

Several features emerge. First, the Cadillac Seville is a bad leverage point. Indeed, it is an outlier in the horizontal as well as in the vertical dimension. This means that its characteristics are pretty different from those of the bulk of the data and its price is much higher than it should be according to the fitted model. The Volkswagen Diesel and the Plymouth Arrow are large good leverage points because they are outlying in the horizontal dimension but not in the vertical one. This means that their characteristics are rather different from the other cars but their prices are in accordance with what the model predicts. Finally, the Cadillac Eldorado, the Lincoln Versailles, the Lincoln Mark V, the Volvo 260, and some others are standard in their characteristics but are more expensive than the model would suggest. They correspond to vertical outliers.

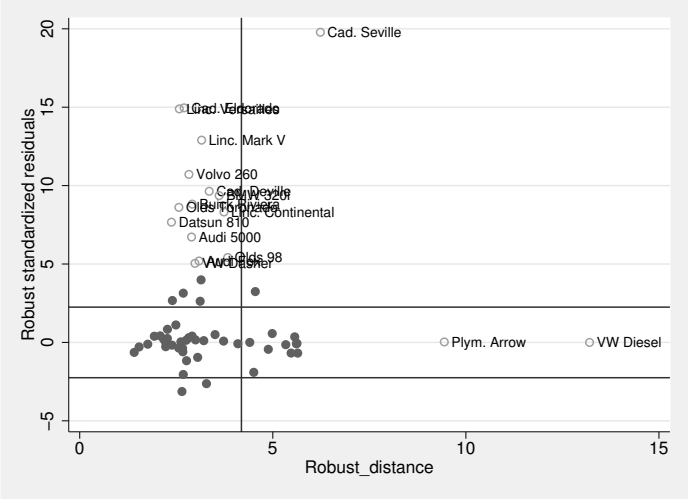

Figure 2. Diagnostic plot of standardized robust residuals versus robust Mahalanobis distances for the `auto.dta` dataset

Are these outlying observations sufficient to distort classical estimations? Because several vertical outliers are present as well as a severe bad leverage point, there is a serious risk that the OLS estimator becomes strongly attracted by the outliers. To illustrate this, we compare the results obtained by using the recommended estimator

with a high breakdown point, `mmregress`, with those obtained by using OLS (`regress`), Huber's monotonic M-estimator (`rreg`), and median regression (`qreg`). MM-estimators with 70% and with 95% efficiency (for normal errors) are considered. The commands (used in a do-file) to estimate these models are

```
. webuse auto, clear
. local exogenous="mpg headroom trunk length weight turn displacement
> gear_ratio foreign i.rep78"
. xi: regress price `exogenous'
. xi: qreg price `exogenous'
. xi: rreg price `exogenous'
. xi: mmregress `exogenous', eff(0.7)
. xi: mmregress `exogenous', eff(0.95)
```

The differences are, as expected, important. We present the regression output in table 1.

(*Continued on next page*)

Table 1: Pricing of autos

Auto dataset. Dependent variable: Price in US$

	`regress`	`qreg`	`rreg`	MM(0.70)	MM(0.95)
Mileage	−43.95	−44.45	−68.91	−44.88	−46.74
	(0.52)	(0.55)	(0.92)	(−1.67)	(1.56)
Headroom	−689.40*	−624.19*	−739.30**	−311.96**	−440.06***
	(1.72)	(1.71)	(2.09)	(2.52)	(4.10)
Trunk space	74.29	37.50	114.53	186.60***	128.98***
	(0.74)	(0.40)	(1.29)	(7.10)	(3.53)
Length	−80.66*	−48.78	−27.50	−33.74**	0.03
	(1.86)	(1.17)	(0.72)	(2.57)	(0.00)
Weight	4.67***	2.89**	2.59*	1.03***	0.37
	(3.19)	(2.10)	(1.99)	(5.29)	(0.62)
Turn circle	−143.71	30.22	−104.26	10.51	−23.79
	(1.11)	(0.30)	(0.91)	(0.48)	(0.69)
Displacement	12.71	9.79	11.34	2.31	2.51
	(1.45)	(1.27)	(1.46)	(0.98)	(0.58)
Gear ratio	115.08	92.28	917.19	492.467	370.20
	(0.09)	(0.08)	(0.82)	(0.89)	(0.99)
Foreign	3064.52***	2496.04**	2326.91**	−91.66	763.91*
	(2.89)	(2.38)	(2.48)	(0.19)	(1.89)
`rep78==2`	1353.80	−355.92	465.98	5.99	31.45
	(0.79)	(0.27)	(0.31)	(0.02)	(0.11)
`rep78==3`	955.44	19.24	488.23	−720.50***	−286.70
	(0.59)	(0.02)	(0.34)	(2.76)	(1.17)
`rep78==4`	976.63	241.79	813.11	−275.89	390.71
	(0.59)	(0.18)	(0.55)	(1.04)	(1.49)
`rep78==5`	1758.00	1325.18	1514.13	606.77*	359.01
	(0.97)	(0.91)	(0.95)	(1.70)	(0.86)
Constant	9969.75	4083.51	2960.68	5352.18***	3495.97
	(1.40)	(0.60)	(0.47)	(3.10)	(1.43)

Absolute value of t statistics is in parentheses.
Significant at ***1%, **5%, and *10%.

Let's compare the results. First, headroom, trunk space, and length seem to be unimportant in explaining prices (at a 5% level) when looking at the OLS, median, and M-estimators (i.e., `regress`, `qreg`, and `rreg`). However, when the influence of outliers (and especially of the bad leverage point) is taken into account (i.e., `MM(0.7)` column), they turn out to be significantly different to zero. If we consider a more efficient estimator (i.e., `MM(0.95)` column), length again becomes insignificant. The weight variable is flagged as significant by most specifications (though the size of the effect is very different). The turn, displacement, and gear ratio variables turn out to be insignificant in all specifications. The foreign dummy is insignificant when using only the most robust estimators.

5 Simulations

Several recent articles have proven the theoretical properties of the estimators described in the previous sections. In this article, we will compare the performances of the Stata codes we implemented with the previously available robust commands and OLS. To do so, we run some simulations according to the following setup. We start by creating a dataset (of size $n = 1{,}000$) by randomly generating five independent explanatory continuous variables (labeled $X_1, \ldots, X_5$) and an error term (e) from six independent univariate normal distributions with mean zero and unit variance. A y variable is then generated according to the formula $y_i = \beta_0 + \sum_{j=1}^{5} \beta_j X_{ij} + e_i$, where $\beta_0 = 0$ and $\beta_j = 1$ for $j = 1, \ldots, 5$. This dataset is called the clean dataset. We then contaminate the data by randomly replacing 10% of the X_1 observations without modifying y. These contaminated points are generated from a normal distribution with mean 5 and standard deviation 0.1 and are bad leverage points. We call this the contaminated dataset. We then repeat this procedure 1,000 times, and each time we estimate the parameters using OLS, L_1, M-estimators, S-estimators, and MM-estimators (with a 95% and a 70% efficiency). On the basis of all the estimated parameters, we measure the bias (i.e., the average of the estimated parameters minus the true value) and the mean squared error (MSE) (i.e., the variance of the estimated parameters plus the square of the bias). The results are presented in table 2. We do not present the results associated with the clean sample because all estimation methods lead to comparable and very low biases.

Table 2: Simulated bias and MSE (sample size $n = 1{,}000$, 10% of outliers)

Estimation method		β_1	β_2	β_3	β_4	β_5	β_0
OLS	Bias	0.7149	0.0015	0.0010	0.0002	0.0016	−0.1440
`reg`	MSE	0.5118	0.0017	0.0018	0.0019	0.0018	0.0223
L_1	Bias	0.6369	0.0006	0.0013	0.0004	0.0011	−0.1281
`qreg`	MSE	0.4071	0.0026	0.0024	0.0027	0.0027	0.0188
M	Bias	0.6725	0.0012	0.0010	0.0005	0.00167	−0.1353
`rreg`	MSE	0.4532	0.0018	0.0018	0.0019	0.0019	0.0200
MM(0.95)	Bias	0.6547	0.0011	0.0009	0.0010	0.00167	−0.1318
`mmregress`	MSE	0.4298	0.0018	0.0018	0.0020	0.0020	0.0190
MM(0.7)	Bias	0.0867	0.0012	0.0028	−0.0008	−0.0010	−0.0164
`mmregress`	MSE	0.0236	0.0015	0.0015	0.0015	0.0014	0.0024

The results of the simulations clearly show that for this contamination setup, the least biased estimator among those we considered is the MM-estimator with an efficiency of 70%. Its bias and MSE are 0.087 and 0.024, respectively, for β_1 and −0.016 and 0.002 for β_0. As a comparison, the bias and MSE of OLS are 0.715 and 0.512 for β_1 and −0.144 and 0.022 for β_0. For the other coefficients, the performances of all estimators are comparable. It is important to stress that if we set the efficiency of MM to 95%, its performance in terms of bias worsens too much and would thus not be desirable. The L_1 and M-estimators (computed respectively with the `qreg` and `rreg` commands) behave rather poorly and have a bias and an MSE comparable to that of OLS.

6 The implemented commands

The `mmregress` command computes the MM-estimators with a high breakdown point, described in section 3, and their standard errors. The general syntax for the command is

`mmregress` *depvar indepvars* [*if*] [*in*] [`, noconstant eff(#)`
`dummies(`*dummies*`) outlier graph label(`*varname*`) replic(#) initial`]

The optional parameter `eff()` fixes the efficiency of the MM-estimator. It can take any value between 0.287 and 1; the higher its value, the more efficient the MM-estimator. While the breakdown point of the MM-estimator is always 50%, its bias increases with its efficiency. Therefore, to have a good compromise between robustness and efficiency of the MM-estimator, we take `eff(0.7)` as a default. The `dummies()` option specifies which variables are dichotomous. If `dummies()` is declared, the initial estimator will be the MS-estimator rather than the S-estimator. Not declaring this option when dummy variables are present may cause the algorithm for computing the S-estimator to fail (see section 3.1).

The `noconstant` option specifies that no constant term has to be considered in the regression. The `outlier` option provides robust standardized residuals and robust Mahalanobis distances. These can be used to construct a diagnostic plot, as discussed in section 3.2. The `graph` option calls on this graphical tool for outlier identification. The `label()` option specifies the variable that will label the outlier. This option only works jointly with the `graph` option. If `label()` is not declared, the label will be the observation number.

The `replic()` option fixes the number of p-subsets to consider in the initial steps of the algorithm. The user can use (7) to change the value of N in accordance to the desired level of P_{clean} or α. The default value for N corresponds to $P_{\text{clean}} = 0.99$ and $\alpha = 0.2$. Finally, the `initial` option will return as output the initial S-estimator, or the MS-estimator if the `dummies()` option is invoked, instead of the final MM-estimator.

The general syntax for the command to compute the S-estimator is

`sregress` *depvar indepvars* [*if*] [*in*] [`, noconstant outlier graph`
`replic(#)`]

The optional parameters available are a subset of those available in `mmregress`; their use is therefore the same as described above. If `sregress` is called exclusively defining a dependent variable, the code will return an M-estimator of scale (sometimes called an S-estimator of scale) and an S-estimator of location of that variable.

The general syntax for the command to compute the MS-estimator is

`msregress` *depvar indepvars* [*if*] [*in*], `dummies(`*dummies*`)` [`noconstant outlier graph replic(`#`)`]

Here again the use of options is comparable to `mmregress`. The `dummies()` option is compulsory and is used to declare which variables among the explanatory are dichotomous.

The general syntax for the command to compute the Huber M-estimator is

`mregress` *depvar indepvars* [*if*] [*in*] [, `noconstant tune(`#`) level(`#`)`]

The `noconstant` option removes the constant, while the `tune()` option changes the tuning parameter as in Stata's `rreg` command. `mregress` is only a slight modification of the `rreg` command.

The general syntax for the minimum covariance determinant command is

`mcd` *varlist* [*if*] [*in*] [, `e(`#`) proba(`#`) trim(`#`) outlier bestsample raw setseed(`#`)`]

The `e()` and `proba()` options are used to modify α and P_{clean}, respectively, in (7); `trim()` sets the percentage of trimming desired; `outlier` calls for robust Mahalanobis distances and flags outliers; `bestsample` identifies the observations that have been used for calculating the robust covariance matrix; `raw` returns the raw robust covariance matrix estimated classically, but on the sample cleaned of identified outliers; and `setseed()` sets the seed. The algorithm for computing the minimum covariance determinant is described in Rousseeuw and van Driessen (1999).

7 Conclusion

The strong impact of outliers on the OLS regression estimator has been known for a long time. Consequently, much literature has been developed to find robust estimators that cope with the "atypical" observations and have a high breakdown point. At the same time, the statistical efficiency of the robust estimators needs to remain sufficiently high. In recent years, it seems that a consensus has emerged to recommend the MM-estimators as the best-suited estimation method, because they combine a high resistance to outliers and high efficiency for regression models with normal errors.

On the other hand, robust methods were not often used by applied researchers, mainly because their practical implementation remained quite cumbersome. Over the last decade, efficient and relatively fast algorithms for computing robust estimators, including MM-estimators, were developed. Nowadays, the use of robust statistical methods has become much more widespread in the applied sciences, like engineering

and chemistry. By providing the Stata code, we also make robust regression methods available for the econometrics research community.

In this article, we summarized the properties of the best-known robust estimation procedures and provided Stata commands to implement them. We created the `mmregress` command (based on a set of commands that can be run separately if needed); furthermore, we showed how this estimator outperforms all "robust" estimators available in Stata by means of a modest simulation study. We hope that this article will contribute to the development of further robust methods in Stata. In particular, development of robust procedures for panel-data and time-series models would be of major interest for applied economic research. The time-series setting will give rise to new problems; for example, selecting random p-subsets will not be appropriate because they break the temporal structure of the data.

8 References

Bramati, M. C., and C. Croux. 2007. Robust estimators for the fixed effects panel data model. *Econometrics Journal* 10: 521–540.

Croux, C., G. Dhaene, and D. Hoorelbeke. 2008. Robust standard errors for robust estimators. Unpublished manuscript. http://www.econ.kuleuven.be/ew/academic/econmetr/members/Dhaene/papers/rsejan2004.pdf.

Edgeworth, F. Y. 1887. On observations relating to several quantities. *Hermathena* 6: 279–285.

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

———. 1981. *Robust Statistics*. New York: Wiley.

Maronna, R. A., and V. J. Yohai. 2000. Robust regression with both continuous and categorical predictors. *Journal of Statistical Planning and Inference* 89: 197–214.

Rousseeuw, P. J. 1984. Least median of squares regression. *Journal of the American Statistical Association* 79: 871–880.

Rousseeuw, P. J., and A. M. Leroy. 2003. *Robust Regression and Outlier Detection*. New York: Wiley.

Rousseeuw, P. J., and K. van Driessen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41: 212–223.

Rousseeuw, P. J., and B. C. van Zomeren. 1990. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association* 85: 633–639.

Rousseeuw, P. J., and V. J. Yohai. 1984. Robust regression by means of S-estimators. In *Robust and Nonlinear Time Series Analysis*, ed. J. Franke, W. Härdle, and R. D. Martin, 256–276. New York: Springer.

Salibian-Barrera, M., and V. J. Yohai. 2006. A fast algorithm for S-regression estimates. *Journal of Computational and Graphical Statistics* 15: 414–427.

Yohai, V. J. 1987. High breakdown-point and high efficiency robust estimates for regression. *Annals of Statistics* 15: 642–656.

About the authors

Vincenzo Verardi is an associate researcher of the FNRS (Belgian National Science Foundation) and a professor of economics and econometrics at the University of Namur and at the Université Libre de Bruxelles (Belgium). His research interests are applied econometrics, development economics, political economics, and public finance.

Christophe Croux is a professor of statistics and econometrics at the Katholieke Universiteit Leuven (Belgium). His research interests are robust statistics, multivariate data analysis, computational statistics, applied time-series analysis, and predictive marketing modeling. His work has been published in *Biometrika*, *Journal of the American Statistical Association*, *Journal of Multivariate Analysis*, *Journal of Marketing Research*, and *The Review of Economics and Statistics*, among other publications.