The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–23

Robust Regression in Stata

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

Vincenzo Verardi is Associated Researcher of the FNRS
and gratefully acknowledges their finacial support

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

Abstract. In regression analysis, the presence of outliers in the data set 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 commands *rreg* and *qreg*. Unfortunately, these methods only resist to some specific types of outliers and turn out to be ineffective under alternative scenarios. In this paper we present more effective robust estimators that we implemented in Stata. We also present a graphical tool that allows recognizing the type of detected outliers.

KEYWORDS: S-estimators, MM-estimators, Outliers, Robustness

JEL CLASSIFICATION: C12, C21, C87

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:

$$y = X\theta + \varepsilon \tag{1}$$

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 $\hat{\theta}$, it is then possible to fit the dependent variable by $\hat{y} = X\hat{\theta}$, and compute the residuals $r_i = y_i - \hat{y}_i$ 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. In the case of the well-known ordinary least squares (LS), this aggregate prediction error is defined as the sum of squared residuals. The vector of parameters estimated by LS is then

$$\hat{\theta}_{LS} = \arg\min_{\theta} \sum_{i=1}^{n} r_i^2(\theta) \tag{2}$$

with $r_i(\theta) = y_i - \theta_0 - \theta_1 X_{i1} - ... - \theta_p X_{ip}$ for $1 \leq i \leq n$. This estimation can be performed in Stata using the `regress` command. A drawback of LS 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 existence of outliers.

The scope of this paper 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 paper is the following: we briefly present, in Section 2, the type of outliers that can be found in regression analysis and introduce the basics of robust regression. We recommend to use high breakdown point estimators, 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 dataset. In Section 5 we provide some simulation results to illustrate how the high breakdown point estimators 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 LS estimator. Rousseeuw and Leroy (1987) 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 LS-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 LS-estimation but it affects statistical inference since 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 affects significantly the LS-estimation of both the intercept and the slope.

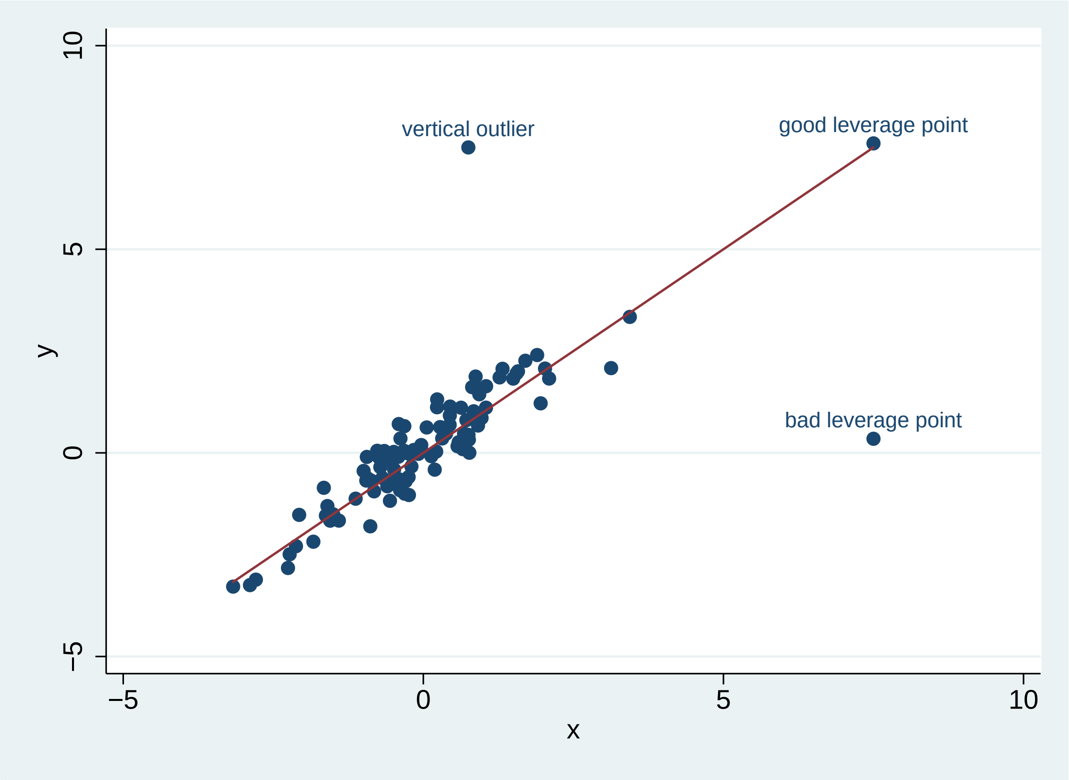

Figure 1: Outliers in regression analysis

Edgeworth (1887) realized that due to the squaring of the residuals, LS becomes extremely vulnerable to the presence of outliers. To cope with this, he proposed a method consisting in 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

$$\hat{\theta}_{L_1} = \arg\min_{\theta} \sum_{i=1}^{n} |r_i(\theta)| \tag{3}$$

The median regression estimator is available in Stata via the `qreg` command as a standard function. This estimator does protect 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 other functions than the absolute value in (3). This allows to increase Gaussian efficiency while keeping robustness with respect to vertical outliers. An M-estimator is defined as

$$\hat{\theta}_{M} = \arg\min_{\theta} \sum_{i=1}^{n} \rho(\frac{r_i(\theta)}{\sigma}) \tag{4}$$

where $\rho(\cdot)$ is a loss function which is even, non decreasing 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 least squares algorithm. To simplify, suppose that σ is known and define weights $\omega_i = \rho(r_i/\sigma)/r_i^2$, then equation (4) can be rewritten as

$$\hat{\theta}_{M} = \arg\min_{\theta} \sum_{i=1}^{n} \omega_i r_i^2(\theta), \tag{5}$$

which is a weighted least-squares estimator. The weights ω_i are however a function of θ and are thus unknown. Using an initial estimate $\tilde{\theta}$ for θ, the weights can be computed and serve as the start of an iteratively reweighted least squares algorithm. Unfortunately, the latter is guaranteed to converge to the global minimum of (4) 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

$$\rho(u) = \begin{cases} 1 - \left[1 - \left(\frac{u}{k}\right)^2\right]^3 & \text{if } |u| \leq k \\ 1 & \text{if } |u| > k \end{cases} \tag{6}$$

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

$$\rho(u) = \begin{cases} \frac{1}{2}(u)^2 & \text{if } |u| \leq c \\ c\,|u| - \frac{1}{2}c^2 & \text{if } |u| > c \end{cases} \tag{7}$$

where $c = 1.345$. Moreover, to give protection against bad leverage points, observations associated to Cook distances larger than 1, receive a weight 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 has not the expected robustness properties for two main reasons. First, Cook distances only manage to identify isolated outliers and are inappropriate in case of existence of clusters of outliers, 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 least squares algorithm are monotone M-estimators that are not robust to bad leverage points and may lead the algorithm to converge to a local instead of a global minimum.

3 High Breakdown point estimators

Full robustness can be achieved by tackling the regression problem from a different perspective. Recall that the LS estimator is based on the minimization of the variance of the residuals. Hence, since the variance is highly sensitive to outliers, LS is largely influenced as well. For this reason, Rousseeuw and Yohai (1987) propose to minimize a measure of dispersion of the residuals that is less sensitive to extreme values than the variance[1]. They call this class of estimators the S-estimators. The intuition behind the method is simple. For LS, the objective is to minimize the variance $\hat{\sigma}^2$ of the residuals. The latter can be rewritten as $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{r_i}{\hat{\sigma}}\right)^2 = 1$. As stated previously, the square value can be damaging as it gives a huge importance to large residuals. Thus, to increase robustness, the square function could be replaced by another loss function ρ which awards less importance to large residuals[2]. The estimation problem would now consist in finding the smallest robust scale of the residuals. This robust dispersion, denoted $\hat{\sigma}^S$ satisfies

$$\frac{1}{n}\sum_{i=1}^{n}\rho(\frac{r_i(\theta)}{\hat{\sigma}^S}) = b \tag{8}$$

[1] Note that 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 available through the command `ltsregress` and `lmsregress`.

[2] As before, $\rho(\cdot)$ is a function which is even, non decreasing for positive values, less increasing than the square with a unique minimum at zero

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

$$\hat{\theta}^S = \arg\min_{\theta} \ \hat{\sigma}^S(r_1(\theta), ..., r_n(\theta)) \tag{9}$$

where $\hat{\sigma}^S$ is the robust estimator of scale as defined in (8).

The choice of $\rho(\cdot)$ is crucial to have good robustness properties and a high Gaussian efficiency. The Tukey Biweight function defined in (6), with $k = 1.547$, is a common choice. This S-estimator resists to a contamination of up-to 50% of outliers. In other words, 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 raises to 96.6% but the breakdown point drops to 10%. To cope with this, Yohai et al. (1987) introduced MM-estimators that combine high breakdown point and a high efficiency. These estimators are redescending M-estimators as defined in (4), but where the scale is fixed at $\hat{\sigma}^S$. So an MM-estimator is defined as

$$\hat{\theta}^{MM} = \arg\min_{\theta} \sum_{i=1}^{n} \rho(\frac{r_i(\theta)}{\hat{\sigma}^S}) \tag{10}$$

The preliminary S-estimator guarantees a high breakdown point, and the the final MM-estimate 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 (10) to guarantee a 95% efficiency of the final estimator.

For computing the MM-estimator, the iteratively reweighted least squares algorithm can be used, taking $\hat{\theta}^S$ as initial value. Once the initial S-estimate is computed, $\hat{\theta}^{MM}$ comes at almost no additionnal 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 respectively `sregress` and `mmregress`) 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 observation (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 (9). 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 sub-samples 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

$$N = \left\lceil \frac{\log(1 - P_{clean})}{\log[1 - (1 - \alpha)^p]} \right\rceil \tag{11}$$

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[3].

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

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 $\hat{\sigma}^S$ is computed from them as in (8). In this way, scale estimates are obtained for each *p-subset*, and an approximation for the final scale estimate $\hat{\sigma}^S$ is then given by the trial value that leads to the smallest scale over all *p-subset.* This approximation can be improved further by carrying some refinement steps, that bring the approximation even closer to the solution of (9).

This algorithm is implemented in Stata and can be called either directly using the `sregress` function or indirectly using the `mmregress` function and invoking the `initial` option. Once the S-estimator is obtained, the MM-estimator directly follows by applying the iteratively reweighted least squares 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 Horelbeke, 2008).

The need of calling on subsampling algorithms becomes Achille's 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 sub-samples 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 somehow out of the scope of the paper and we thus do not elaborate on it here. We nevertheless briefly describe the Stata command implemented (`msregress`) to compute it in practice. Note that 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 analyzes separately from the bulk of 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 $r_i/\hat{\sigma}^S$, with $r_i \equiv r_i(\hat{\theta}^S)$, 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 the i^{th} row-vector of matrix X, for $i \leq i \leq n$. Obviously both μ and Σ should be estimated robustly if we want these distances to resist to the presence of outliers. Several methods have been proposed to estimate robustly the Mahalanobis distances. In Stata, the command `hadimvo` 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 distributed as a

χ^2_p distribution 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 if the car is not built in the US. 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 done by calling the followin Stata command lines (that are described more precisely in Section 6)

```
   . webuse auto, clear

   . xi:  mmregress price mpg headroom trunk length weight turn
displacement gear_ratio foreign i.rep78, outlier graph label(make)
```

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 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 since they are outlying in the horizontal dimension but not on the vertical one. This means that their characteristics are rather different from the other cars but their price is in accordance with what the model predicts. Finally the Cadillac

Eldorado, the Lincoln Versaille, 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.

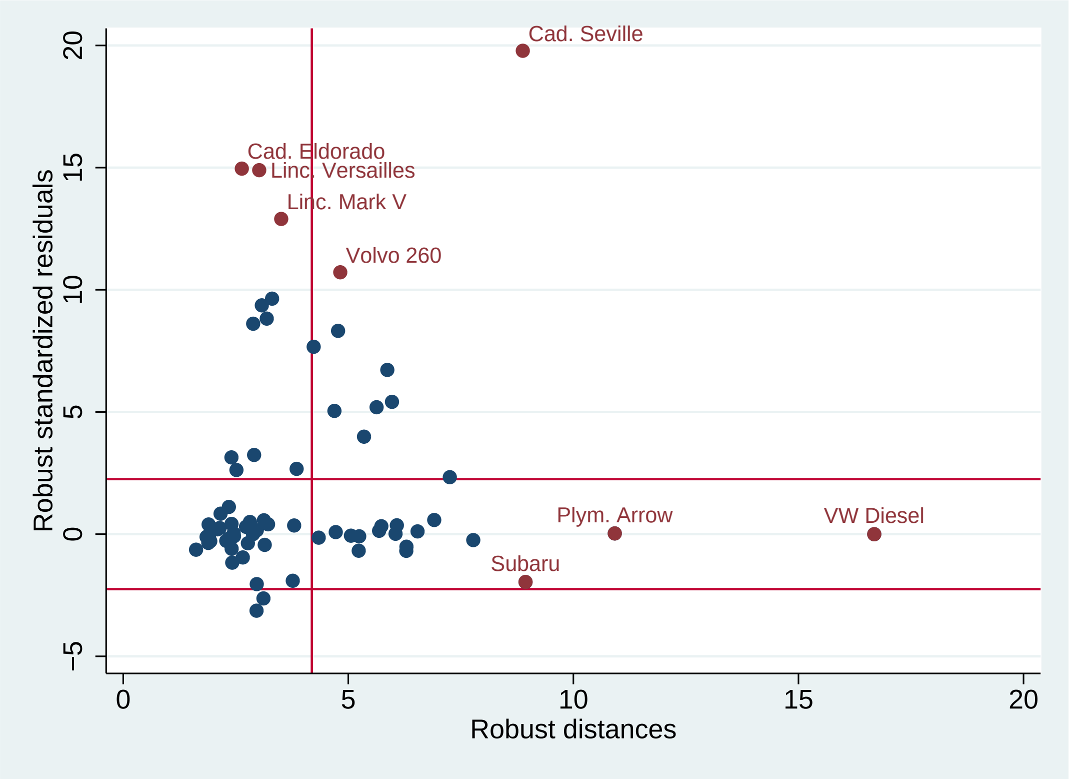

Figure 2: Diagnostic plot of standardized robust residuals versus roust Mahalanobis distances for the auto.dta dataset

Are these outlying observations sufficient to distort classical estimations? Since several vertical outliers are present as well as a severe bad leverage point, there is a serious risk that the least squares estimator becomes strongly attracted by the outliers. To illustrate this, we compare the results obtained using the recommended high breakdown point estimator `mmregress` with those obtained using least squares (`regress`), Huber's monotonic M-estimator (`rreg`) and median regression (`qreg`). Both an MM-estimator with 70% and 95% efficiency (for normal errors) are considered. The command lines (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.

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 in parentheses

Significant at ***1%, ** 5%, * 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 becomes again 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 only using the most robust estimators.

5 Simulations

Several recent articles have proven the theoretical properties of the estimators described in the previous sections. In this paper we will compare the performances of the Stata codes we implemented with the previously available robust commands and LS. To do so we run some simulations according to the following setup. We start by creating a dataset (of size $n = 1000$) by randomly generating 5 independent explanatory continuous variables (labelled $X_1, \dots, X_5$) 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 $j = 1, \dots, 5$. This dataset is called the clean dataset. We then contaminate the data by replacing randomly 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 1000 times and each time we estimate the parameters using LS, L_1, M, S 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 (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 to the clean sample since all estimation methods lead to comparable and very low biases.

Table 2: Simulated Bias and MSE (sample size n=1000, 10% of outliers)

Estimation method		β_1	β_2	β_3	β_4	β_5	β_0
LS	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 respectively of 0.087 and 0.024 for β_1 and of -0.016 and 0.002 for β_0. As a comparison, the bias and MSE of LS are 0.715 and 0.512 for β_1 and -1.144 and 0.02 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 LS.

6 The implemented commands

The `mmregress` command computes the high breakdown point regression **MM-estimators**, described in Section 3, and their standard errors. The general syntax for the command is:

`mmregress` *varlist* [`if` *exp*] [`in` *range*] [, `eff(#)` `dummies(`*varlist*`)` `noconstant` `outlier` `graph` `label(`*varname*`)` `replic(#)` `init`]

The first optional parameter is `eff`, which allows to fix 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 as a default value `eff`=0.7. The `dummies` option allows to declare which variables are dichotomous. In case `dummies` is declared, the initial estimator will be the MS 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 third option, `noconstant`, states that no constant term has to be considered in the regression. The fourth option, `outlier`, provides robust standardized residuals, and robust Mahalanobis distances. They can be used to construct a diagnostic plot, as discussed in Section 3.2, and the option `graph` calls on this graphical tool for outliers identification. The option `label` allows to chose the variable that will label the largest outliers. This option only works jointly with the `graph` option. If this option is not declared, the label will be the observation number. The option `replic` allows to fix the number of p-subsets to consider in the initial steps of the algorithm. The user can use equation (11) to change the value of N in accordance to his desired level of P_{clean} and/or α. The default value for N corresponds to $P_{clean} = 0.99$ and $\alpha = 0.2$. Finally, the option `init` will return as output the initial S-estimator, or the MS-estimator if the option `dummy` is invoked, instead ot the final MM-estimator.

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

`sregress` *varlist* [`if` *exp*] [`in` *range*] [, `noconstant` `outlier` `graph` `replic(#)`]

The optional parameters available are a subset of those available in `mmregress`. Their use is therefore the same. Note that if `sregress` is called defining exclusively 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 of the **MS-estimator** is

`msregress` *varlist* [`if` *exp*] [`in` *range*], `dummies(`*varlist*`)` [`noconstant` `outlier` `graph` `replic(#)`]

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

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

`mregress` *varlist* [`if` *exp*] [`in` *range*] [, `noconstant` `tune(#)`]

The `noconstant` option allows to remove the constant while the `tune` option allows to change the tuning parameter as in the `rreg` command. Note that `mregress` is only a slight modification of the `rreg` code.

The general syntax for the **Minimum Covariance Determinant** command is

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

The options `e` and `proba` are used to modify respetively α and P_{clean} in (11), `trim` sets the percentage of trimming desired, `outlier` calls for for robust Mahalanobis distances and flags outliers, `best` identifies the observations that have been used for calculating the robust covariance matrix, `raw` returns the raw robust covariance matrix instead of one 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 least square regression estimator is known for a long time. Consequently, a large 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, since they combine a high resistance to outliers and high efficiency at regression models with normal errors.

On the other hand, robust methods were not so 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 becomes much more widespread in the applied sciences, like engeneering and chemistry. By providing the Stata code, we make robust regression methods also available for the econometrics research community.

In this paper we summarize the properties of the best known robust estimation procedures and provide Stata code to implement them. We create the `mmregress` command (based on a set functions that can be run separately if needed). We furthermore show how this estimator outperforms all "robust" estimators available in Stata by mean of a modest simulation study. We hope that this paper 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*-observation subsets will not be appropriate, since they break the temporal structure of the data

8 References

[1] Bramati, M. C. and C. Croux. 2007. Robust Estimators for the Fixed Effects Panel Data Model. *Econometrics Journal* 10(3): 521-540.

[2] Croux, C., G. Dhaene, and D. Hoorelbeke. 2008. Robust Standard Errors for Robust Estimators, manuscript. www.econ.kuleuven.be.christophe.croux/public.

[3] Edgeworth, F. Y. 1887. On Observations Relating to Several Quantities. *Her-*

mathena, 6: 279-285.

[4] Huber, P. 1964. Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics* 35(1): 73-101.

[5] Huber, P. 1981. *Robust Statistics.* New York: John Wiley and Sons.

[6] Maronna, R., and V. Yohai. 2006. Robust regression with both continuous and categorical predictors. *Journal of Statistical Planning and Inference* 89: 197-214.

[7] Maronna, R., D. Martin and V. Yohai. 2006. *Robust Statistics.* New York: John Wiley and Sons.

[8] Rousseeuw, P. J.. Least Median of Squares Regression. *Journal of the American Statistical Association* 79: 871-880.

[9] Rousseeuw, P. J. and A. Leroy. 1987. *Robust Regression and Outlier Detection.* New York: John Wiley and Sons.

[10] Rousseeuw, P. J. and K. Van Driessen. 1999. A Fast algorithm for the Minimum Covariance Determinant Estimator. *Technometrics* 41: 212-223.

[11] Rousseeuw, P. J. and B. Van Zomeren. 1990. Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association* 85: 633-639.

[12] Rousseeuw, P. J. and V. Yohai. 1987. Robust Regression by Means of S-estimators in *Robust and Nonlinear Time Series Analysis*: 256-272, edited by J. Franke, W. Härdle and D. Martin. Berlin: Springer Verlag.

[13] Salibian-Barrera, M. and V. Yohai. 2006. A Fast Algorithm for S-regression Estimates. *Journal of Computational and Graphical Statistics* 15: 414-427.

[14] Yohai, V. 1987. High Breakdown-point and High Efficiency Estimates for Regression. *The Annals of Statistics* 15: 642-665.