Benchmark Testing of Algorithms for Very Robust Regression

Francesca Torti
Dipartimento di Economia, Università di Parma, Italy,[*]
Domenico Perrotta
European Commission, Joint Research Centre, Ispra, Italy,[†]
Anthony C. Atkinson
The London School of Economics[‡]
and Marco Riani
Dipartimento di Economia, Università di Parma, Italy[§]

August 23, 2011

Abstract

The methods of very robust regression resist up to 50% of outliers. The algorithms for very robust regression rely on selecting numerous subsamples of the data. We describe new algorithms for LMS and LTS estimators that have increased efficiency due to improved combinatorial sampling. These and other publicly available algorithms are compared for outlier detection. An algorithm using the forward search has the best properties for both size and power of the outlier tests.

Keywords: combinatorial search; concentration step; forward search (FS); least median of squares (LMS); least trimmed squares (LTS); logistic plots of power; masking; outlier detection

1 Introduction

Multiple regression is one of the main tools of applied statistics. It has however long been appreciated that ordinary least squares as a method of fitting regression models is exceptionally susceptible to the presence of outliers. Instead, very robust methods, that asymptotically resist 50% of outliers, are to be preferred.

Several algorithms have been proposed for very robust regression. All algorithms estimate the parameters by least squares applied to subsets of observations; they differ in the number and sizes of the subsets and how those subsets are found. We introduce new

[*]e-mail: francesca.torti@nemo.unipr.it
[†]e-mail: domenico.perrotta@ec.europa.eu
[‡]e-mail: a.c.atkinson@lse.ac.uk
[§]e-mail: mriani@unipr.it

versions of standard algorithms with improved combinatorial searches for these subsets. The purposes of our paper are to describe these improvements and to compare publicly available versions of seven algorithms.

Very robust regression was introduced by Rousseeuw (1984) who developed suggestions of Hampel (1975) that led to the Least Median of Squares (LMS) and Least Trimmed Squares (LTS) algorithms. The algorithms for finding these estimates compare properties of the fits to many subsets of p observations, p being the number of parameters in the linear model. Of course, in these cases, the least squares fit is an exact fit to the observations. In the reweighted versions of these algorithms described in §2.2 all k observations that are identified as outliers are rejected and least squares estimation is used on the remaining $n-k$ observations. Parameter estimation thus involves two subsets of the data.

More recently developed methods fit to subsets of several sizes. In the Forward Search (FS: Atkinson and Riani 2000 with a recent discussion in Atkinson, Riani, and Cerioli 2010) subsets of increasing size m are used, starting from $m_0 = p$ and increasing until all observations not in the subset are identified as outliers. Least squares is used for parameter estimation for each subset. In this way the subset size flexibly responds to the properties of the data. The FS for regression is described by Riani and Atkinson (2007).

In order to speed up the LTS algorithm for very large data sets, Rousseeuw and Van Driessen (2006) introduced a “concentration” step into the LTS algorithm in which subsets of both size p and of size $h = (n+p+1)/2$ are used. Our numerical experience, outlined in the appendix, is that this can result in a slower algorithm when using the same number of subsets as the algorithm without concentration. However, if we reduce the number of subsets, to allow for the effect of the concentration step, we lose accuracy in parameter estimation (see §3). We provide a different, faster, version of LTS that also uses concentration steps whilst overcoming these problems. We consider only reweighted versions of these algorithms.

To compare these seven methods of robust regression we look at their ability to detect outliers. The motivation comes from the study by Riani, Atkinson, and Cerioli (2009), who reported excellent size and power for the FS for multivariate data. We likewise find that the FS dominates the other six regression methods.

Our paper is structured as follows. The algorithms for very robust regression are described in §2: the six for comparison in §2.2 and the FS in §2.3. Comparison of the size of the tests is in §3, with power comparisons in §4. Brief conclusions are in §5. Details of the improved combinatorial searches over subsets are in the appendix.

2 Algorithms for Very Robust Regression

We compare and contrast the properties of seven methods for very robust regression. The algorithms that we use are the Forward Search Regression routine (FSR), the LMS and LTS routines and their reweighted versions contained in the MATLAB toolbox called FSDA (Forward Search Data Analysis) at `http://www.riani.it/MATLAB`, two versions of Fast LTS, the first based on the implementation contained in the LIBRA toolbox at `http://wis.kuleuven.be/stat/robust` (Verboven and Hubert 2005, Verboven and Hubert 2010) as originally proposed by Rousseeuw and Van Driessen (2006) and the second based on our implementation in FSDA.

2.1 Least Squares for Subsets of Observations

In the regression model $y = X\beta + \epsilon$, y is the $n \times 1$ vector of responses, X is an $n \times p$ full-rank matrix of known constants, with ith row x_i^T, and β is a vector of p unknown parameters. The normal theory assumptions are that the errors ϵ_i are i.i.d. $N(0, \sigma^2)$.

Let $S(m)$ be a subset of m observations for which the matrix of regressors is $X(m)$. Least squares on this subset of observations yields parameter estimates $\hat{\beta}(m)$ and $s^2(m)$, the mean square estimate of σ^2 on $m - p$ degrees of freedom. Residuals can be calculated for all observations including those not in $S(m)$. The n resulting least squares residuals are

$$e_i(m) = y_i - x_i^T \hat{\beta}(m), \qquad i = 1, 2, \ldots, n. \tag{1}$$

The algorithms use different functions of the $e_i(m)$ to declare observations as outlying. The forward search algorithm is designed to have size α of declaring an outlier free sample to contain at least one outlier. Therefore, when evaluating the properties of the other algorithms, we use Bonferroni corrections for simultaneity, with level $\alpha^* = \alpha/n$, so taking the $1 - \alpha^*$ cutoff value of the reference distribution. In our calculations $\alpha = 0.01$.

2.2 Some Very Robust Procedures

• LMS - Least Median of Squares

The LMS estimator minimizes the hth ordered squared residual $e^2_{[h]}(\beta)$ with respect to β, where $h = \lfloor (n + p + 1)/2 \rfloor$ and $\lfloor . \rfloor$ denotes integer part. Algorithms for LMS find an approximation to the estimator, selecting the best fit, that is the one giving the smallest hth ordered squared residual out of all n residuals, to randomly chosen subsets of p observations. Since there are p parameters and p observations in the subset, the fit to the observations in the subset is exact. In the PROGRESS algorithm of Rousseeuw and Leroy (1987, §4.4) sampling of observations is at random. Samples containing duplicate rows are rejected, although the same subset may be selected more than once.

In our versions of LMS and LTS we instead sample without replacement from the lexicographic list of all possible subsets. If ${}^nC_p < 5 \times 10^7$ we store a list of all subsets in Integer8 format. This very large number was chosen as being, at the time of writing, reasonably below the the current limit of storage of a typical 32-bit PC. Subsets are then sampled without replacement from this list. For larger values of ${}^nC_p$ we use an algorithm which avoids explicit storage of the subsets. The strategy requires binomial coefficients which are either computed with an algorithm which can be dated back to Lehmer (1964) or retrieved from a look-up table of Pascal's triangle values, previously built using the property that each cell is given by the sum of the number above it and that above to the left. The strategy is built to trade off time and space requirements on the basis of the machine resources available at the time of execution. Details of this procedure can be found in the appendix. In our comparisons we sampled 10,000 subsets.

Let $\tilde{\beta}_{\text{LMS}}$ be the LMS estimate of β. Rousseeuw (1984) bases the estimate of σ on the value of the median squared residual $e^2_{\text{med}}(\tilde{\beta}_{\text{LMS}})$. As in Rousseeuw and Leroy (1987, p. 202) we define

$$\tilde{\sigma}_{\text{LMS}} = 1.4826\{1 + 5/(n - p)\} \left\{ e^2_{\text{med}}(\tilde{\beta}_{\text{LMS}}) \right\}^{0.5}. \tag{2}$$

We declare as an outlier any observation i for which the absolute scaled residual

$$|e^{\mathrm{S}}_{\mathrm{LMS},i}| = |e_i(\tilde{\beta}_{\mathrm{LMS}})|/\tilde{\sigma}_{\mathrm{LMS}} > \Phi^{-1}(1-\alpha^*) \qquad (i = 1, \ldots, n). \tag{3}$$

• LTS - Least Trimmed Squares

The convergence rate of $\tilde{\beta}_{\mathrm{LMS}}$ is $n^{-1/3}$. Rousseeuw (1984, p. 876) also suggested Least Trimmed Squares (LTS) which has a convergence rate of $n^{-1/2}$ and so better properties than LMS for large samples. As opposed to minimising the median squared residual, we now find $\tilde{\beta}_{\mathrm{LTS}}$ to

$$\text{minimize } SS_{\mathrm{T}}\{\hat{\beta}(h)\} = \sum_{i=1}^{h} e_i^2\{\hat{\beta}(h)\}, \tag{4}$$

where, for any subset $\mathcal{H}$ of size h the parameter estimates $\hat{\beta}(h)$ are straightforwardly estimated by least squares.

Let the minimum value of (4) be $SS_{\mathrm{T}}(\tilde{\beta}_{\mathrm{LTS}})$. We base the estimator of σ^2 on this residual sum of squares. However, since the sum of squares contains only the central h observations from a normal sample, the estimate needs scaling. The variance of the truncated normal distribution containing the central h/n portion of the full distribution is

$$\sigma^2_T(h) = 1 - \frac{2n}{h}\Phi^{-1}\left(\frac{n+h}{2n}\right)\phi\left\{\Phi^{-1}\left(\frac{n+h}{2n}\right)\right\},$$

where $\phi(.)$ and $\Phi(.)$ are respectively the standard normal density and c.d.f. This result follows from the more general results of Tallis (1963) on elliptical truncation. To estimate σ^2 we accordingly take

$$\tilde{\sigma}^2_{\mathrm{LTS}} = SS_{\mathrm{T}}(\tilde{\beta}_{\mathrm{LTS}})/\{h \times \sigma^2_T(h)\}. \tag{5}$$

Outliers are found as in (3) but with LTS parameter estimates.

• LMSR and LTSR - Reweighted Least Median of Squares and Reweighted Least Trimmed Squares

To increase efficiency, reweighted versions of the LMS and LTS estimators can be computed. These reweighted estimators are computed by giving weight 0 to observations which (3), or its LTS analogue, suggests are outliers. We then obtain a sample of reduced size $n-k$, possibly outlier free, to which OLS is applied. (Rousseeuw and Leroy 1987) pp. 44-45, suggest to estimate σ^2 by dividing the residual sum of squares from OLS by $n-k-p$. On the other hand, for comparability with the implementation in the LIBRA library, we divided by $n-k-1$.

Let the parameter estimates be $\tilde{\beta}_{\mathrm{LXSR}}$ and $\tilde{\sigma}_{\mathrm{LXSR}}$, where by LXSR we mean either LMSR or LTSR. The outliers are the k^* observations rejected at this second stage, that is those for which

$$|e_i(\tilde{\beta}_{\mathrm{RLXS}})|/\tilde{\sigma}_{\mathrm{RLXS}} > \Phi^{-1}(1-\alpha^*) \qquad (i = 1, \ldots, n). \tag{6}$$

We may find that $k^* \gtreqless k$. Both in (3) and (6) we perform a test of Bonferronised size α^*.

• LTSRL - "Fast" Least Trimmed Squares

The simple LTS algorithm outlined above can be slow as the number of observations increases. Rousseeuw and Van Driessen (2006) introduced an improved algorithm with increased speed and several refinements. The most important of these is the concentration step, which is similar to the step used in the Forward Search in moving from a subset of size m to one of $m + 1$.

Given a set of n residuals from a first parameter estimate $\tilde{\beta}(h_1)$ based on a subset of h observations, the absolute values of the residuals are ordered and a new subset h_2 formed as those observations giving the h smallest values of $|e_i\{\tilde{\beta}(h_1)\}|$. This process can be repeated on some or all of the initial subsets, or iterated to convergence on a few specially selected initial subsets. A discussion of the properties of such options in the more general context of S-estimation is in Maronna, Martin, and Yohai (2006, §5.7).

Rousseeuw and Van Driessen (2006, p. 38) give the pseudo-code for their algorithm. For small values of n, perhaps less than 600, they suggest taking 500 random subsets of size $h_1 = p$. Each is subject to two concentration steps. The ten subsets so obtained with the smallest value of $SS_{\mathrm{T}}\{\hat{\beta}(h_3)\}$ in (5) then have the concentration steps iterated to convergence, with the minimum value of $SS_{\mathrm{T}}\{\hat{\beta}(h_\infty)\}$ providing the LTS estimate. For larger sample sizes more complicated, but related strategies are suggested. In practice, a divide and conquer strategy based on blocks of 300 observations is used to reduce the number of concentration steps.

Once the estimate $\tilde{\beta}_{\mathrm{LTS}}$ has been obtained, the estimation of σ and the identification of outliers are as in the LTS algorithm above.

• LTSRF - Fully Iterated Reweighted Least Trimmed Squares

Our experience is that the LTSRL algorithm was slower, for problems of our size, than our implementation of LTS that uses a very efficient random sample generation method (see the appendix). We think this was, at least in part, also due to the housekeeping (storage and sorting) involved in the concentration step strategy in the algorithm of Rousseeuw and Van Driessen (2006). Therefore, in our implementation of the Fast LTS we made some simplifications in the use of the concentration steps. Instead of our usual 10,000 subsamples, we also drew only 500. However, all of these were subject to concentration steps to convergence. We observed that the number of iterations was essentially independent of the number of variables and was increasing, from about 5 iterations for $n = 50$ to 20 iterations for $n = 1000$, with a rate which decreased as the sample size increased.

2.3 The Forward Search

2.3.1 Background

The forward search fits subsets of observations of size m to the data, with $m_0 \leq m \leq n$. Let $S^*(m)$ be the subset of size m found by the forward search. Least squares on this subset of observations yields parameter estimates $\hat{\beta}(m^*)$ and $s^2(m^*)$. From (1) the n resulting least squares residuals can be written $e_i(m^*)$. The search moves forward with the augmented subset $S^*(m+1)$ consisting of the observations with the $m+1$ smallest absolute values of $e_i(m)$. To start we take $m_0 = p$ and search over subsets of p observations to find the subset that yields one of the very robust estimates of β described in §2.2.

2.3.2 Testing for Outliers

In the search we want m to increase until all $n-m$ observations not in $S^*(m)$ are outliers. To test for outliers the deletion residuals are calculated for these $n-m$ observations not in $S^*(m)$. These residuals, which form the maximum likelihood tests for the outlyingness of individual observations, are

$$r_i(m^*) = e_i(m^*)/\sqrt{s^2(m^*)\{1+h_i(m^*)\}}, \tag{7}$$

where the leverage $h_i(.) = x_i^T\{X(.)^T X(.)\}^{-1}x_i$. The observation nearest to those forming $S^*(m)$ is that with the minimum value of $|r_i(m^*)|, i \notin S^*(m)$. Call this i_{min}. To test whether observation i_{min} is an outlier we use the absolute value of the minimum deletion residual $r_{\text{min}}(m^*)$. If the absolute value is too large, the observation i_{min} is considered to be an outlier, as well as all other observations not in $S^*(m)$. See Riani and Atkinson (2007) for further details.

2.3.3 The FSR Algorithm

The automatic procedure in the FSR algorithm is based on that of Riani, Atkinson, and Cerioli (2009) who used scaled Mahalanobis distances to detect outliers in multivariate normal data. For regression these distances are replaced by the absolute value of the minimum deletion residual. Some further details of the regression algorithm are given by Torti (2011).

The implementation in FSDA allows appreciable diagnostic and graphical output. Riani, Atkinson, and Perrotta (2012) also present an illustrative examples of a FS analysis of the kind of trade data motivating Riani et al. (2008).

3 The Size of the Tests

In our simulation studies we considered regression models with an intercept and v explanatory variables over a range of values of v from 1 to 10 (so p ranged from 2 to 11). The values of the x_{ij} $(i = 1,\dots,n; j = 1,\dots,v)$ were sampled once for each pair of values of v and n from independent standard normal distributions $\mathcal{N}(0,1)$. Since the tests for outliers are functions of residuals, which do not depend on the values of the parameters β, these values were set to zero. We added standard normal random variables to these null models, estimating the parameters and repeating the process $n_{\text{sim}} = 10,000$ times. We count the proportion of samples declared to contain at least one outlier.

We start with results for $v = 5$. Figure 1 gives the size of the seven tests for sample sizes n in the range 100 to 1,000. For the largest sample sizes ($n = 500$ and 1000) all tests, except the traditional LTS and LMS without re-weighting, have sizes near 1%. Unlike the other tests, the size of the FS is stable around 1% even for the smallest sample sizes, whereas the other methods perform comparatively poorly. In particular note that the fast re-weighted LTS (both the LTSRF and the LTSRL versions) has larger sizes than the standard re-weighted LTS (LTSR). We believe this is because the concentration step of the fast LTS is based on a number of subsets, 500, which is not sufficient to visit all the local minima that are explored with the standard LTS estimate using 10,000 subsets. Further simulations with increased numbers of subsets did indeed show that the average squared norm of the parameter estimates decreased as the number of subsets increased.

The results for $v = 10$ in Figure 2 are similar to, but more extreme than, those for $v = 5$. For the larger sample sizes all sizes, except LTS without re-weighting, are between 1% and 2%. For smaller sample sizes all reweighted versions of LTS perform with the two fast versions (LTSRF and LTSR) having larger sizes than LTSR.

As a last exploration of size we look at results for small sample sizes when $v = 1$. As Figure 3 shows, the sizes for LMSR and FS are closest to 1% for all values of n. The size for LMS increases with sample size up to a final 2% at $n = 95$. Standard LTS is constantly between 4% and 5% with the three re-weighted LTS methods (LTSR, LTSRF and LTSRL) now virtually equivalent, the size decreasing with the sample size, from an initial 4% to a final 1.5%.

4 The Power of the Tests

The comparisons of size suggest that, overall, FS and LMSR have the best performance. However, power comparisons reveal that LMSR has the lowest power of the seven procedures.

In our power calculations we shifted the mean of 5% or 30% of the observations by up to 7 units (the errors in our observations were standard normal) and calculated the average power, that is the proportion of samples declared to contain at least one outlier. We start in Figure 4 with 5% contamination when $v = 5$ and $n = 500$. FS clearly has the best power. The other methods are almost equal to each other. For the majority of shift values, the ordering of the rest from best to worst is standard LTS and LMS, the two fast re-weighted LTS (LTSRF and LTSRL), which are indistinguishable, and finally the LTS and LMS with re-weighting (LTSR and LMSR). The labels in the figure are positioned from the top to the bottom to reflect this ranking.

The conclusions from Figure 4 appear clear. However, in general there are two problems of interpretation for such straightforward plots of power. One is that they are bounded below and above by zero and one; thus the eye focuses on comparisons in the centre of the plot, that is on powers around 50%. The other is that it is impossible to adjust by eye for the size of the different procedures. Accordingly we accompany our plots of power by logit plots.

Figure 5 repeats Figure 4 with the power r replaced by logit$\{(r-3/8)/(n+1/4)\}$ (Cox and Hinkley 1974, p. 470). Under this transformation procedures with the same power plot as parallel curves regardless of size. For an example see Atkinson (1985, Figure 8.12). In our case, Figure 5 again shows the superior performance of FS. However it is now apparent that the seemingly superior performance of LTS over LMS in Figure 4 is a reflection of its larger size in Figure 1. In Figure 5 the plots for LTS and LMS are parallel. However, we know from the results in Figure 1 for $n = 500$ that FS has the correct size, that is power when the shift is zero.

The difference between the procedures becomes more marked as the level of contamination increases. In our last two examples we have 30% contamination. In Figures 6 and 7 we consider small samples with $n = 50$ when $v = 1$. The plots show that the re-weighted methods LTS (LTSR, LTSRF and LTSRL) are indistinguishable over most of the range. The highest power is for the FS, although its size was the lowest in Figure 3.

An interesting feature of the logit transformed power in Figure 7 is apparent around a shift of three where the three reweighted LTS methods are distinguishable. They and

LMSR have slightly reduced, rather than increased, power compared to smaller shifts. This phenomenon comes from the masking of outliers which leads to an overestimate of the error variance and a reduction in the number of outliers detected. As the shift increases further the outliers become identifiable. An example of such inflation of variance due to unidentified outliers in multivariate data is in §7.3 of Atkinson, Riani, and Cerioli (2004).

The good performance of the FS becomes even more apparent when v increases from one to five. The results are in Figures 8 and 9. Both figures show the outstanding performance of the FS. The logit transformation of Figure 9 not only confirms the performance of the FS but shows the strong effect of masking on the four reweighted rules.

5 Conclusions

The idea of reweighing very robust estimates to obtain information from all apparently non-outlying observations is intuitively appealing. However our power comparisons show that all four reweighted rules have a behaviour which is not as good as expected (at least for the sample sizes and the number of variables we considered). The best behaviour for both size and power is provided by the FS. Of the other two rules LMS behaves better for size than LTS. The logit plots show that, once the tests have been adjusted for size, there is little to choose between them for power. This is perhaps surprising in view of the superior asymptotic properties of LTS; the convergence rate of the LMS estimates is $n^{-1/3}$ rather than $n^{-1/2}$ for LTS. However, the simulation results of Rousseeuw (1984, p. 183) fail to reveal any differences in the behaviour of the two estimates for values of n up to 2,000.

Of course, the primary interest in fitting regression models in applied statistics is to use the fitted model rather than solely to detect outliers. Our paper concertizes exclusively on a through investigation of outlier detection. A less extensive evaluation of the properties of the parameter estimates, and the relationship with outlier detection, for fewer rules is given by Riani, Atkinson, and Perrotta (2012). Their results show a strong relationship between the variance and bias of parameter estimates in very robust regression and the ability of the fitted model to detect outliers.

Acknowledgment This paper was completed at the Isaac Newton Institute for Mathematical Sciences in Cambridge, England, during the 2011 programme on the Design and Analysis of Experiments.

References

Atkinson, A. C. (1985). *Plots, Transformations, and Regression*. Oxford: Oxford University Press.

Atkinson, A. C. and M. Riani (2000). *Robust Diagnostic Regression Analysis*. New York: Springer–Verlag.

Atkinson, A. C., M. Riani, and A. Cerioli (2004). *Exploring Multivariate Data with the Forward Search*. New York: Springer–Verlag.

Atkinson, A. C., M. Riani, and A. Cerioli (2010). The forward search: theory and data analysis (with discussion). *Journal of the Korean Statistical Society 39*, 117–134. doi:10.1016/j.jkss.2010.02.007.

Cochran, W. G. (1977). *Sampling Techniques (3rd edition)*. New York: Wiley.

Cox, D. R. and D. V. Hinkley (1974). *Theoretical Statistics*. London: Chapman and Hall.

Hampel, F. R. (1975). Beyond location parameters: robust concepts and methods. *Bulletin of the International Statistical Institute 46*, 375–382.

Knuth, D. (2005). *Generating All Combinations and Partitions*. The Art of Computer Programming, Vol. 4, Fascicle 3. Reading, Mass.: Addison-Wesley.

Knuth, D. E. (1997). *The Art of Computer Programmig, Volume 2: Seminumerical Algorithms, Third Edition*. Reading, Mass.: Addison-Wesley.

Lehmer, D. H. (1964). The machine tools of combinatorics. In E. F. Beckenbach (Ed.), *Applied Combinatorial Mathematics*, pp. 5–31. New York: Wiley.

Maronna, R. A., D. R. Martin, and V. J. Yohai (2006). *Robust Statistics: Theory and Methods*. New York: Wiley.

Matsumoto, M. and T. Nishimura (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul. 8*, 3–30.

Riani, M. and A. C. Atkinson (2007). Fast calibrations of the forward search for testing multiple outliers in regression. *Advances in Data Analysis and Classification 1*, 123–141. doi:10.1007/s11634-007-0007-y.

Riani, M., A. C. Atkinson, and A. Cerioli (2009). Finding an unknown number of multivariate outliers. *Journal of the Royal Statistical Society, Series B 71*, 447–466.

Riani, M., A. C. Atkinson, and D. Perrotta (2012). Calibrated very robust regression. (Submitted).

Riani, M., A. Cerioli, A. Atkinson, D. Perrotta, and F. Torti (2008). Fitting mixtures of regression lines with the forward search. In F. Fogelman-Soulié, D. Perrotta, J. Piskorski, and R. Steinberger (Eds.), *Mining Massive Data Sets for Security*, pp. 271–286. Amsterdam: IOS Press.

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 (1987). *Robust Regression and Outlier Detection*. New York: Wiley.

Rousseeuw, P. J. and K. Van Driessen (2006). Computing LTS regression for large data sets. *Data Mining and Knowledge Discovery 12*, 29–45.

Tallis, G. M. (1963). Elliptical and radial truncation in normal samples. *Annals of Mathematical Statistics 34*, 940–944.

Torti, F. (2011). *Advances in the Forward Search: methodological and applied contributions*. Padova: Italian Statistical Society.

Verboven, S. and M. Hubert (2005). LIBRA: a MATLAB library for robust analysis. *Chemometrics and Intelligent Laboratory Systems 75*, 127–136. doi:10.1016/j.chemolab.2004.06.003.

Verboven, S. and M. Hubert (2010). Matlab library LIBRA. *WIREs Computational Statistics 2*, 509–515.

A Appendix: Efficient Random Sample Generation

LMS and LTS estimation (and, in general, all algorithms of robust statistics) spends a large part of the computational time in sampling subsets of observations and then computing parameter estimates from the subsets. In addition, each new subset has to be checked as to whether it is in general position (that is, it has a positive determinant). For these reasons, when the total number of possible subsets ${}^nC_p$ is much larger than the number of distinct subsets used for estimation (e.g. $k = 500$ or $k = 10000$ as in this paper), we need an efficient method to generate a new random p-element subset without checking explicitly if it contains repeated elements. We also need to ensure that the current subset has not been previously extracted. The lexicographic approach that we present in this appendix fulfils these requirements. In combinatorial terms, the problem can be reformulated as follows:

> Given a totally ordered set $S = \{1, 2, ..., n\}$, generate a set of k different p-combinations $\{s_1, ..., s_p\}$ of ordered elements of S.

Following Lehmer (1964), the p-combinations of elements in S can be generated in lexicographic order without repetitions. The lexicographic ordering is a biunivocal correspondence between the p-combinations and the set of integers $\{N \mid 0 \le N < {}^nC_p\}$. This correspondence, called by Knuth (2005, pp. 5-6) a "combinatorial number system", has an explicit formulation.

In one direction, the generation order N has the following unique p binomial coefficient terms representation (called *RANK* in the computer science literature)

$$N = \sum_{i=1}^{p} \binom{n - s_i}{p - i + 1}, \tag{A1}$$

with $0 \le x_p < \ldots < x_2 < x_1 < n$, where $x_i = n - s_i$. For example, when $n = 7$, the generation order of the 3-combination $\{2, 5, 6\}$ is: $N = \binom{5}{3} + \binom{2}{2} + \binom{1}{1} = 12$. In the inverse direction (*UNRANK*) the function that, given the generation order N, produces the p-combination at position N in the lexicographic ordering is defined by the following greedy algorithm, again due to Lehmer:

$$\begin{array}{ll}
x_1 \text{ is the greatest integer such that} & \binom{x_1}{p} \le N \\
x_2 \text{ is the greatest integer such that} & \binom{x_2}{p-1} \le N - \binom{x_1}{p} \\
x_3 \text{ is the greatest integer such that} & \binom{x_3}{p-2} \le N - \binom{x_1}{p} - \binom{x_2}{p-1} \\
\vdots & \\
x_p \text{ is the greatest integer such that} & x_p \le N - \sum_{i=1}^{p-1} \binom{x_i}{p-i+1}.
\end{array} \tag{A2}$$

Now, if we want to extract k different subsamples, we simply need to extract k random integers $N_1, \ldots, N_k$ between 0 and ${}^nC_p - 1$ and find the corresponding p-combinations. We employ the following strategy:

A. For small values of ${}^nC_p$, use a look-up table with all p-combinations built beforehand in lexicographic order, and extract rows $N_1, \ldots, N_k$.

B. For small to moderate values of ${}^nC_p$, use UNRANK (A2) to build the p-combinations associated with the random integers $N_1, \ldots, N_k$.

C. For big values of ${}^{n}C_p$, abandon the lexicographic approach and:

   1. If $n \geq 4p$, repeatedly sample from $S = \{1, 2, ..., n\}$ until there are p unique values. Repeat this k times.
   2. If $n < 4p$, randomize S (i.e. make a number of switches between two elements chosen at random positions in S) and take the first p as a p-combination. Repeat this k times.

Strategy B (using UNRANK) requires a number of binomial coefficients. Depending on this number and the memory available, we adopt one of these two options:

a. Explicitly compute the binomial coefficients; each requires $\min(p-1, n-p-1)$ multiplications.

b. Build beforehand a Pascal triangle for given n and p, which requires $O(np)$ addition operations (each cell is given by the sum of the cell above it and that above to the left). Then, use the Pascal triangle as a look-up table.

If the random number generator is unbiased, as in the case of the Mersenne twister algorithm (Matsumoto and Nishimura 1998) or the general linear congruential method (Knuth 1997, pp. 10-26), our strategies ensure that every combination is chosen with equal probability $1/\binom{n}{p}$. To avoid duplicates in the random integers generated by strategies A and B we adopt a very simple and efficient systematic sampling technique, consisting in selecting every $\binom{n}{p}/k$th integer from an ordered list. See Cochran (1977) for systematic sampling.

We have observed that with this strategy the execution time of the traditional LTS and LMS algorithms are drastically reduced and become even better than that of the Fast LTS, at least for the combinations of n and p discussed in the present paper. In addition, our fully iterated LTS in the FSDA toolbox turns out to be at least twice as fast as the Fast LTS in LIBRA, which uses just two concentration steps for all combinations of p and $n < 600$. For example, on a computer with a 1.6 GHz Intel T-5450 when $p = 6$ and $n = 500$, Libra takes about 2 seconds to obtain the FAST LTS solution based on two refining C-steps for each subsample and 500 subsets. In contrast, our FAST fully iterated LTS (or LMS) takes about 1.1 seconds, while the traditional LTS (or LMS) solution based on 10000 subsets needs just 0.5 seconds. This difference becomes more dramatic if we increase the number of subsets in FAST LTS to 10000. The Fast LTS in LIBRA slightly outperforms our fully iterated reweighted LTS only for $p = 11$ and $n \geq 600$, i.e. when the divide and conquer strategy of Rousseeuw and Van Driessen (2006) is applied to reduce the application of the concentration steps.

Legend

- traditional LTS is represented with a dashed line.
- LTS with the final reweighing step is represented with dashed and dotted line.
  - LTSR stands for the standard re-weighted LTS.
  - LTSRL stands for the fast re-weighted LTS as in Libra.
  - LTSRF stands for the fast re-weighted LTS as in our FSDA.
- LMS and its re-weighted version are represented with a hatched line
- FS (Forward Search) is represented with a solid line.

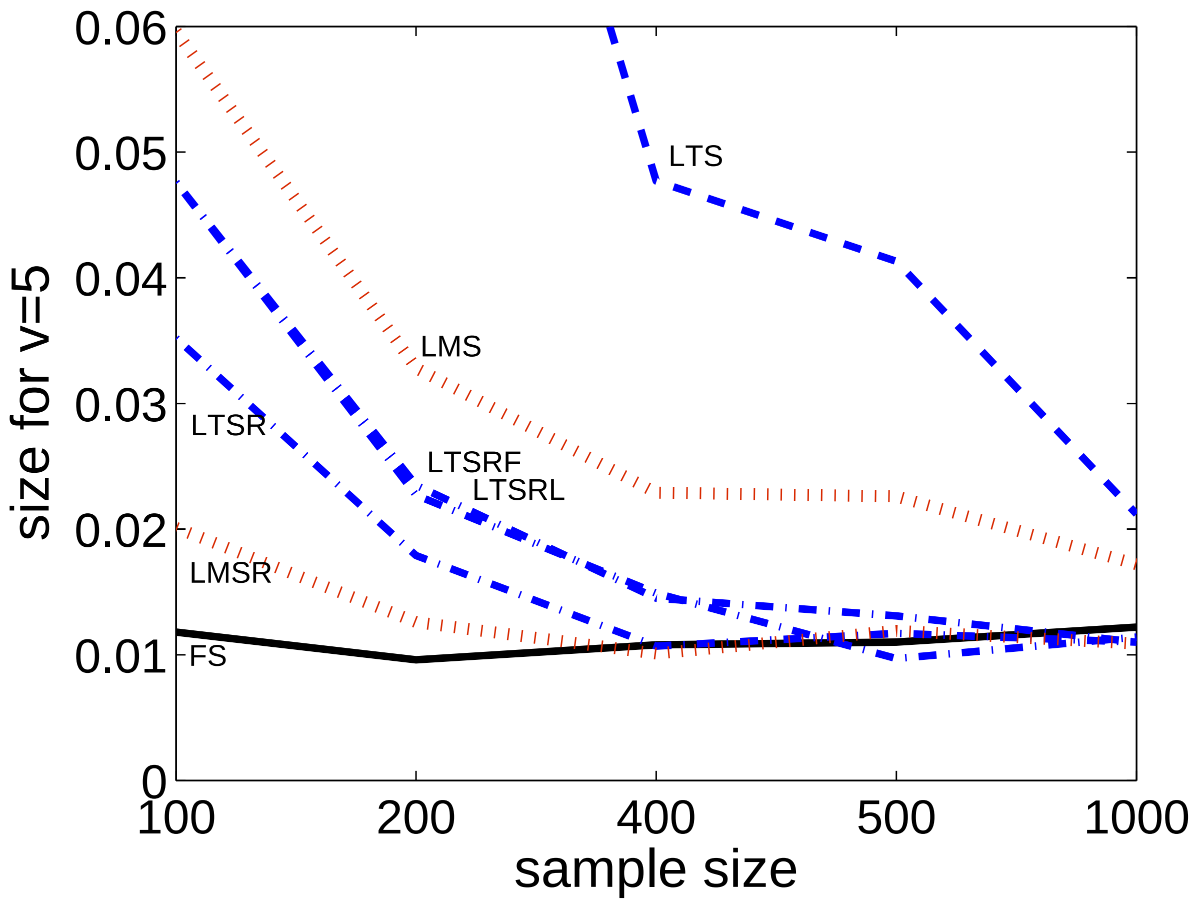

Figure 1: Size for $v = 5$. The FS is stable around 1% even for smaller sample sizes, where the other methods perform badly. The size of LTS is particularly high. Note the uneven spacing of values of n.

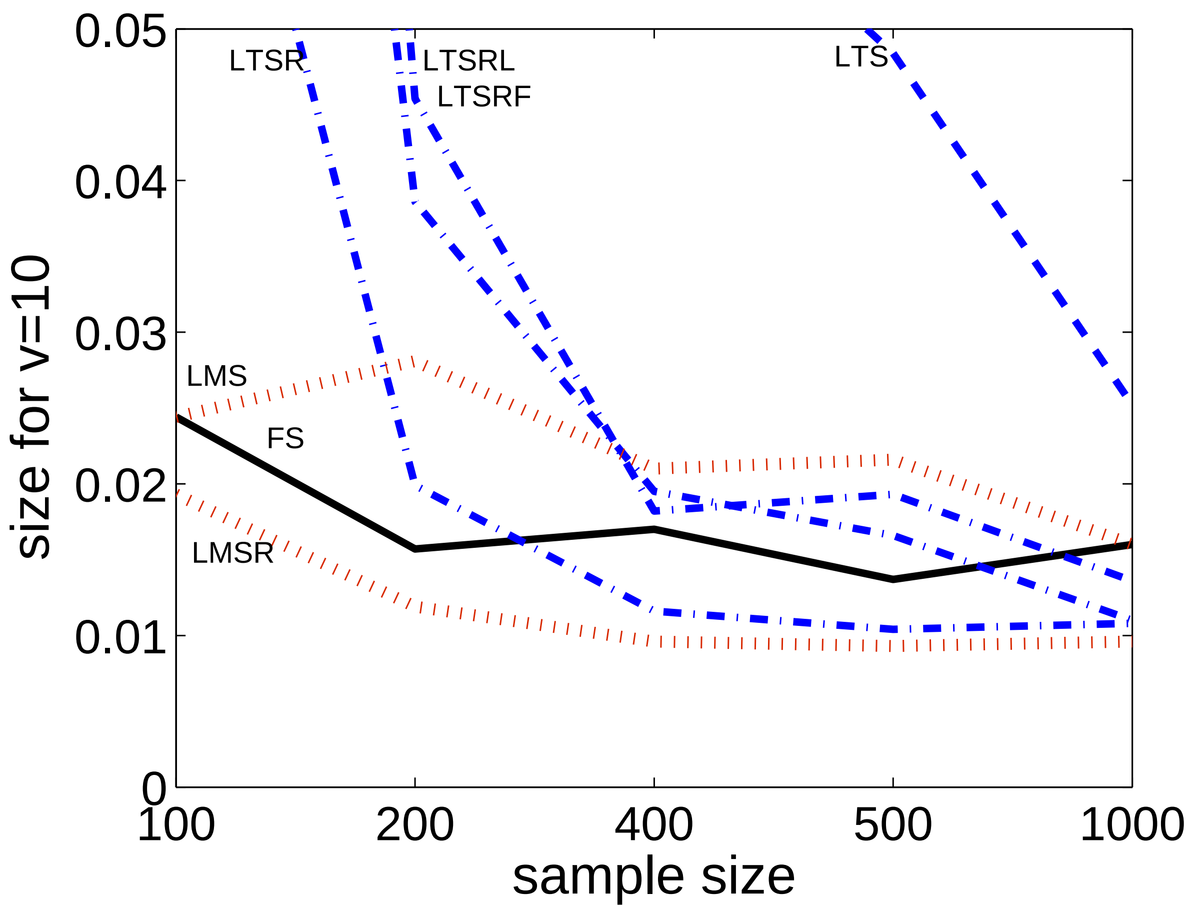

Figure 2: Size for $v = 10$. For large sample sizes, all the curves except the LTS without re-weighting are between 1% and 2%. For smaller sample sizes LTS performs badly in all cases.

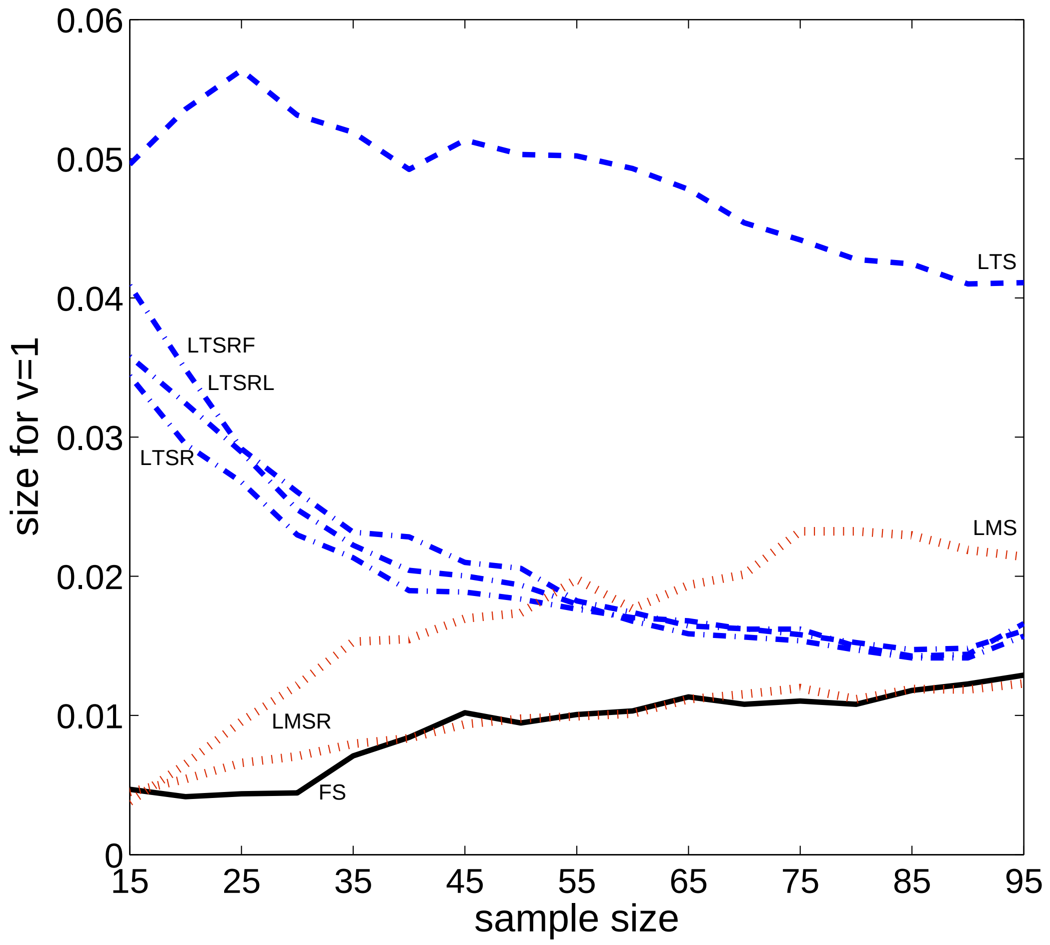

Figure 3: Smoothed values of size for small n and $v = 1$. The FS and the LMS re-weighted (LMSR) are near 1% for all sample sizes. The standard LMS tends to increase with the sample size up to a final 2%. The standard LTS is constantly between 4% and 5%. The three LTS re-weighted methods (LTSR, LTSRF and LTSRL) are almost equivalent and their size decreases with the sample size, from an initial 4% to a final 1.5%. In this case the fact of using only 500 subsets to estimate the fast LTS re-weighted (LTSRF and LTSRL) does not produce a big difference from the standard LTS re-weighted (LTSR), which uses 10, 000 subsets.

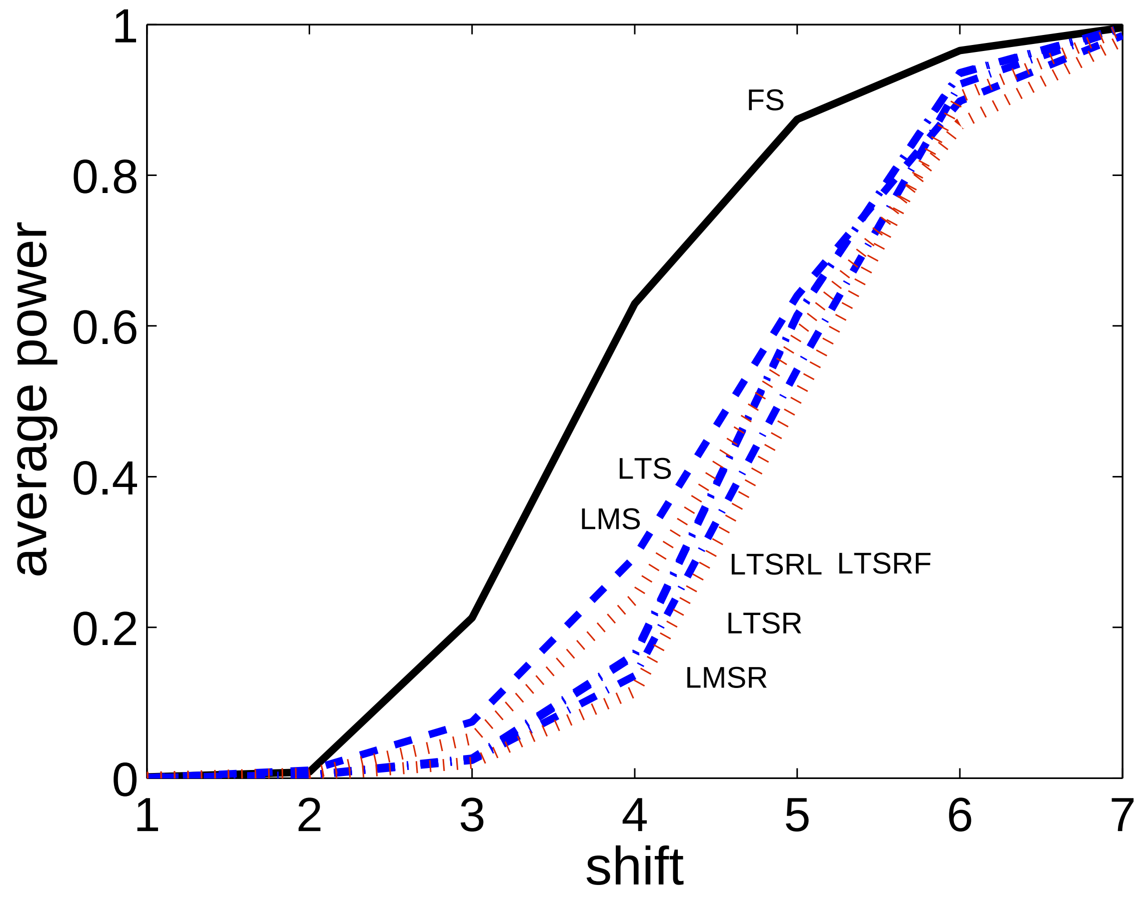

Figure 4: Average power for $n = 500, v = 5$ and 5% contamination. The FS clearly has the best power. The other methods are almost equivalent. For most of the shift values, from the best to the worst we find the standard LTS and LMS, the two fast LTS re-weighted (LTSRF and LTSRL), indistinguishable, and finally the LTS and LMS with re-weighting step (LTSR and LMSR). The labels in the figure are positioned from the top to the bottom to reflect this ranking.

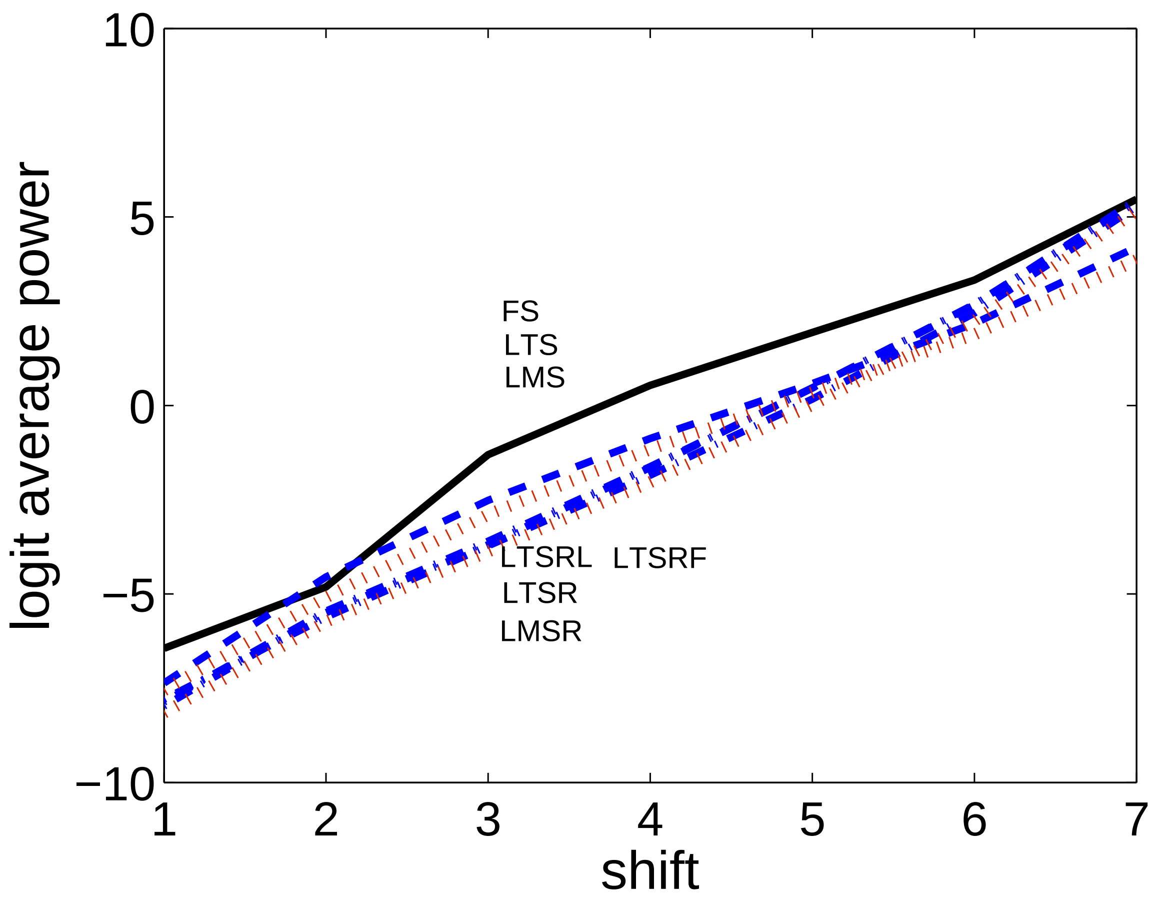

Figure 5: Logit average power for $n = 500, v = 5$ and 5% contamination. For intermediate shift values, the FS has the best performance. The other methods have indistinguishable performance until the shift becomes large.

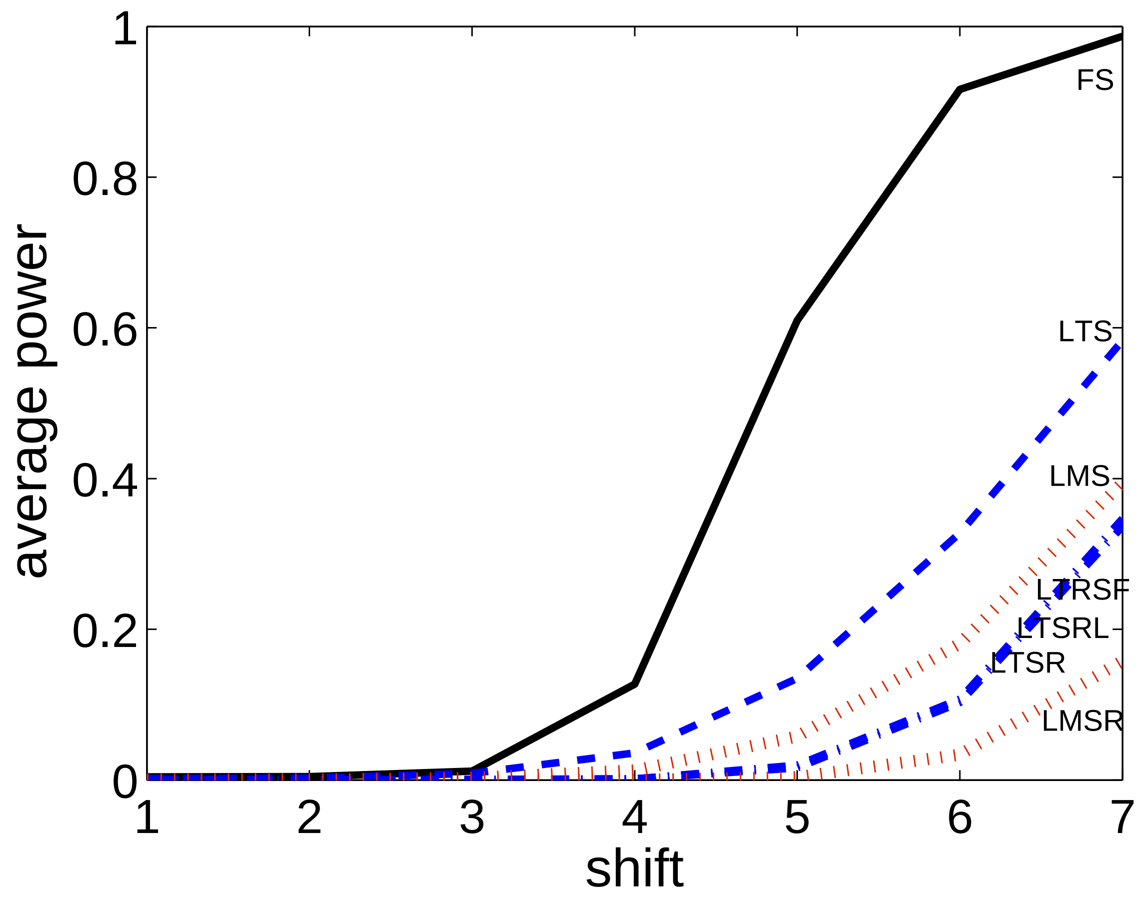

Figure 6: Average power for $n = 50, v = 1$ and 30% contamination. The LTS re-weighted methods (LTSR, LTSRF and LTSRL) are indistinguishable. Power is highest for the FS, although the size was the lowest in Figure 3.

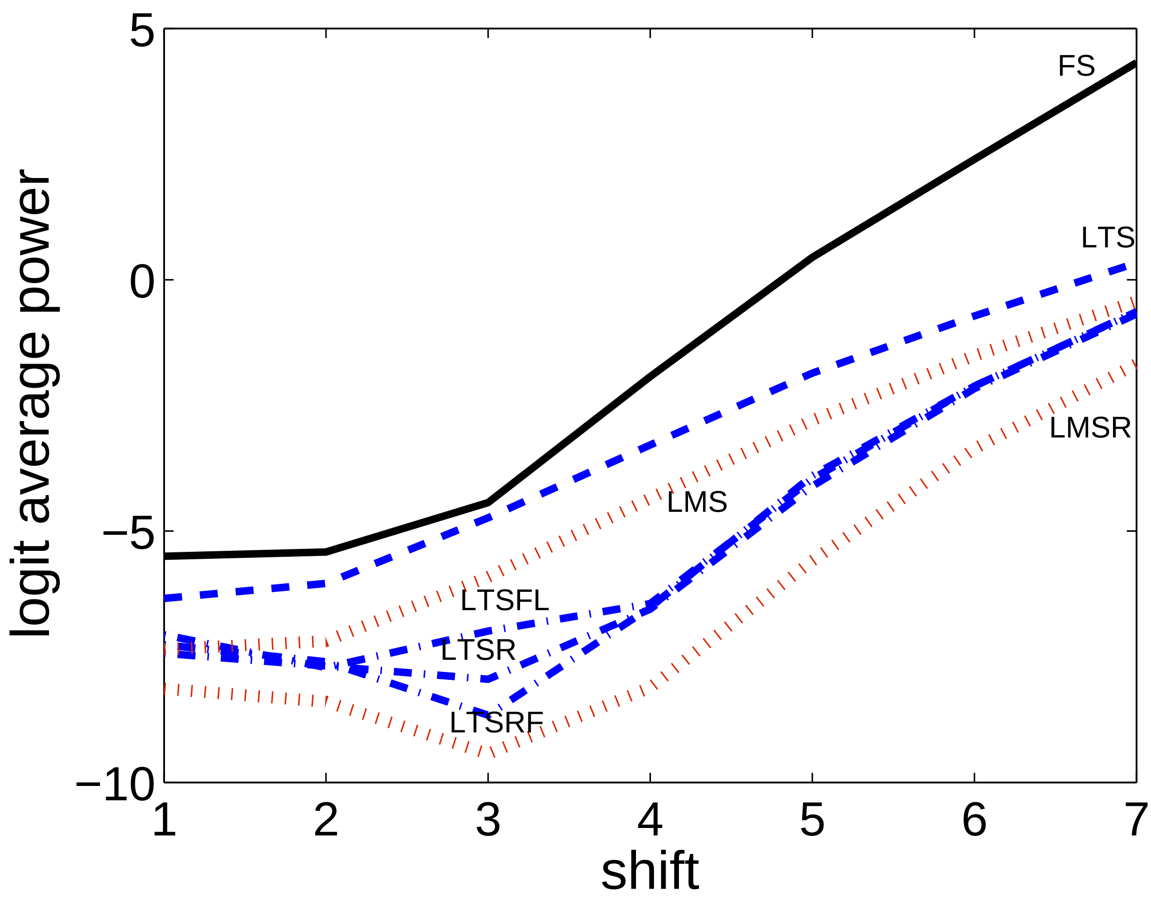

Figure 7: Logit of average power for $n = 50, v = 1$ and 30% contamination. The re-weighted fast LTS (LTSRF and LTSRL) and the standard LTS re-weighted (LTSR) almost coincide for all shift values except three. These three methods suffer some masking for this intermediate contamination level, which disappears for larger contaminations.

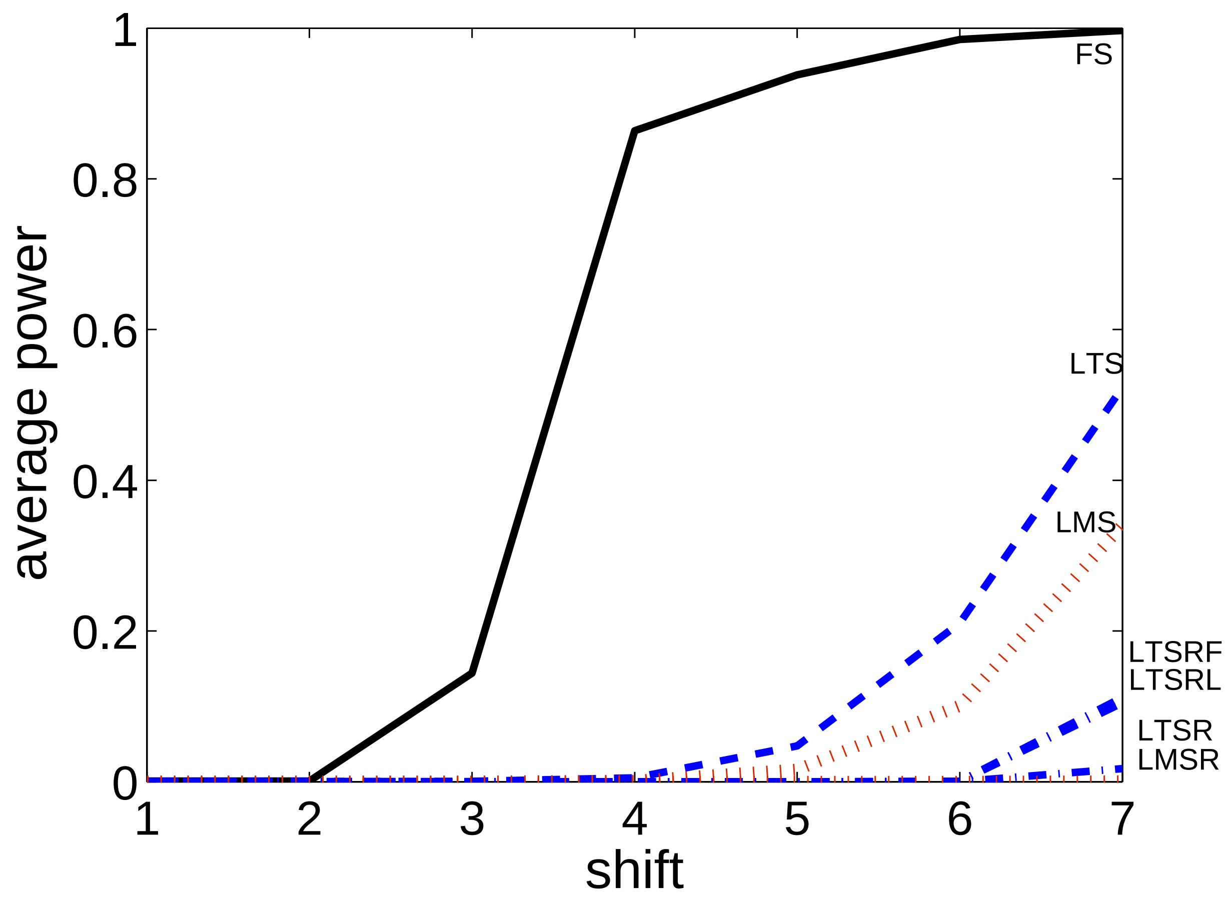

Figure 8: Average power for $n = 500, v = 5$ and 30% contamination. The FS has the best power. The labels are positioned from the top to the bottom to reflect their ranking.

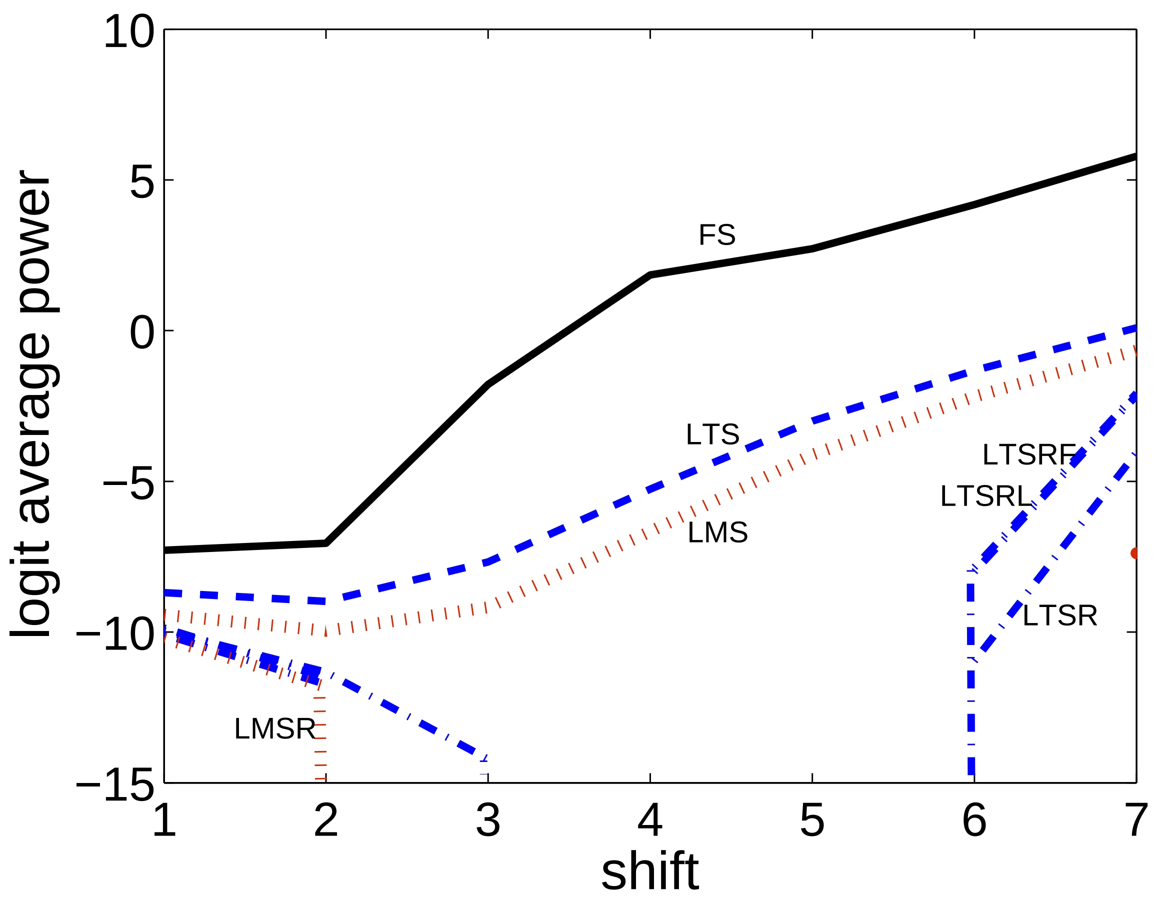

Figure 9: Logit average power for $n = 500, v = 5$ and 30% contamination. For shift values between 2 and 6 the average power for LMS and LTS with re-weighting (LMSR, LTSR, LTSRF and LTSRL) is zero and the logit is not defined. These methods suffer from masking for these intermediate contamination levels. This phenomenon was seen less strongly in Figure 7. FS has by far the highest power.