Journal of Statistical Software

July 2005, Volume 14, Issue 7. http://www.jstatsoft.org/

Rank-Based Analyses of Linear Models Using R

Jeff T. Terpstra
North Dakota State University

Joseph W. McKean
Western Michigan University

Abstract

It is well-known that Wilcoxon procedures out perform least squares procedures when the data deviate from normality and/or contain outliers. These procedures can be generalized by introducing weights; yielding so-called weighted Wilcoxon (WW) techniques. In this paper we demonstrate how WW-estimates can be calculated using an L_1 regression routine. More importantly, we present a collection of functions that can be used to implement a robust analysis of a linear model based on WW-estimates. For instance, estimation, tests of linear hypotheses, residual analyses, and diagnostics to detect differences in fits for various weighting schemes are discussed. We analyze a regression model, designed experiment, and autoregressive time series model for the sake of illustration. We have chosen to implement the suite of functions using the R statistical software package. Because R is freely available and runs on multiple platforms, WW-estimation and associated inference is now universally accessible.

Keywords: estimation, inference, linear models, R functions, rank-based procedures, robust, weighted Wilcoxon estimates.

1. Introduction

One of the most widely used models in statistics is the linear model which is typically written as

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_p X_{pi} + \varepsilon_i = \beta_0 + \boldsymbol{X}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, 2, \ldots, n \quad (1)$$

where Y_i is an observed univariate response variable, $\boldsymbol{X}_i = (X_{1i}, X_{2i}, \ldots, X_{pi})^\top$ is a $p \times 1$ vector of (known) explanatory variables, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^\top$ is a $p \times 1$ regression parameter vector, and β_0 is the intercept parameter. When needed, we will write the combined parameter vector as $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta}^\top)^\top$. Throughout this paper we assume that the ε_i are iid according to a continuous distribution function F that satisfies $F(0) = 1/2$ and $f(0) > 0$ where f denotes the corresponding probability density function.

In regards to (1), it is well-known that Wilcoxon procedures out perform least squares (LS)

procedures when F deviates from the Gaussian distribution. Furthermore, in designed experiments, Wilcoxon procedures provide protection against outlying responses (i.e. Y_i). See, for example, the books Hettmansperger (1984) and Hettmansperger and McKean (1998). Here and after we refer to the Hettmansperger and McKean reference as HM.

Briefly, a Wilcoxon estimate of $\boldsymbol{\beta}$ is defined to be a minimum of the following dispersion function

$$D_R(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(R\left(\varepsilon_i(\boldsymbol{\beta})\right) - \frac{n+1}{2} \right) \varepsilon_i(\boldsymbol{\beta}) \tag{2}$$

where $\varepsilon_i(\boldsymbol{\beta}) = Y_i - \boldsymbol{X}_i^{\intercal}\boldsymbol{\beta}$ and $R\left(\varepsilon_i(\boldsymbol{\beta})\right)$ denotes the rank of $\varepsilon_i(\boldsymbol{\beta})$ among $\{\varepsilon_j(\boldsymbol{\beta})\}$. This corresponds to the dispersion function of Jaeckel (1972) with Wilcoxon scores. Because (2) is invariant to location, β_0 can not be simultaneously estimated with $\boldsymbol{\beta}$. Instead, $\widehat{\beta}_0 = \text{med}\{\varepsilon_i(\widehat{\boldsymbol{\beta}})\}$, where $\widehat{\boldsymbol{\beta}}$ is a minimum of (2), is typically used as an estimate of β_0. See, for example, Section 3.5.2 of HM (1998).

Now, since Wilcoxon estimates are only robust in regards to the response, they may not be appropriate in observational studies if the independent variables (i.e. $\boldsymbol{X}_i$) are contaminated. As an alternative, one can consider an analysis based on weighted Wilcoxon (WW) estimates. In short, a WW-estimate corresponds to a minimum of the following objective function

$$D_{WR}(\boldsymbol{\beta}) = \sum_{1 \le i < j \le n} b_{ij} \left| \varepsilon_j(\boldsymbol{\beta}) - \varepsilon_i(\boldsymbol{\beta}) \right| \tag{3}$$

where b_{ij} denotes a weight to be used in the (i,j)th comparison. Note that $D_{WR}(\boldsymbol{\beta})$ is essentially a weighted version of Gini's mean difference measure of scale. When $b_{ij} = 1$ for $i \neq j$ and 0 otherwise, it can be shown (e.g. Hettmansperger 1984, p.277) that $D_{WR}(\boldsymbol{\beta}) = 2D_R(\boldsymbol{\beta})$; hence the name WW-estimate.

WW-estimates and corresponding inference have been studied extensively in the context of linear regression models. See, for example, Sievers (1983); Naranjo and Hettmansperger (1994); Chang, McKean, Naranjo, and Sheather (1999) and Chapter 5 of HM (1998). Depending on the weighting scheme (see Section 2) used, WW-estimates can attain a continuous totally bounded influence function and a positive breakdown point, which for one class is the maximum of 50%. Thus, this class of estimates can be simultaneously robust and highly efficient. This makes WW-estimates particularly appealing when it comes to autoregressive time series analysis where an observation plays a dual role as both a response and an explanatory variable. The paper by Terpstra, McKean, and Naranjo (2001) provides a good overview of WW-estimates and their application to autoregressive time series modeling.

In spite of the abundance of literature, the desirable robustness and efficiency properties, and the wide applicability to various linear models, this class of estimators has yet to be implemented in any mainstream statistical software packages. It is here where we hope to make a contribution. That is, we present an R (see e.g. Ihaka and Gentleman 1996) implementation of WW-estimates and corresponding inference.

More specifically, we review some popular weighting schemes from the literature and illustrate how these weights and corresponding estimates can be calculated with an L_1 regression routine in Section 2. Furthermore, the asymptotic theory for WW-estimation and inference is well established. For example, the asymptotic distribution of the estimates, tests of linear

hypotheses, studentized residuals, and diagnostics for comparing different WW-fits are summarized in Section 3. In Section 4 we present some of the critical functions that are used to compute these estimates, test statistics, and diagnostics. We also discuss some of the R packages that are required for our implementation. Some data set examples, which include a regression model, a designed experiment, and an autoregressive time series are used for illustration in Section 5. The purpose of these examples is to illustrate the wide applicability of our functions to various kinds of data sets (e.g. observational and experimental). Lastly, Section 6 provides some examples relating to simulation studies and bootstrapping. We conclude with a brief discussion of possible implementations in other statistical software packages.

2. Computational details

To compute the WW-estimate one can use an L_1 regression routine with

$$b_{ij}(Y_j - Y_i) \quad \text{and} \quad b_{ij}\,(\boldsymbol{X}_j - \boldsymbol{X}_i)$$

playing the role of the response variables and design points, respectively. Note that this is essentially how weighted least squares regression estimators are calculated. However, to our knowledge, R does not provide an explicit L_1 regression function. Nevertheless, since L_1 regression estimates are equivalent to (median) quantile regression estimates, the **quantreg** package written by Roger Koenker can be used to calculate the WW-estimate. We refer the interested reader to Koenker and Bassett (1978) for more information on regression quantiles. To obtain the estimates then, we can call the `rq.fit.br` and `rq.fit.fn` functions using the aforementioned weighted pairwise differences to obtain the estimates. We note that `rq.fit.br` and `rq.fit.fn` are based on exterior and interior point methods, respectively. According to the R documentation for `rq`, `rq.fit.br` is recommended for smaller scaled problems. For instance, when the sample size is smaller than 5,000 and the number of regressors is less than 20.

As shown, WW-estimates are readily obtained once the weights (i.e. b_{ij}) have been determined. In this paper, we essentially consider three classes of weights; namely constant weights, Mallows (1975) weights, and Schweppe (e.g. Handschin, Kohlas, Fiechter, and Schweppe 1975) and Chang *et al.* (1999) weights. Briefly, weight functions that only depend on the design points are typically referred to as Mallows weights. Examples are given in Sections 2.2 and 2.3. On the other hand, a weighting scheme that depends on both the design point and the response is a member of the Schweppe class. See, for example, Section 2.4. We now give a brief discussion of some of the more popular members of these classes that appear in the literature.

2.1. Wilcoxon weights

The simplest weighting scheme corresponds to $b_{ij} = 1$ for $i \neq j$ and 0 otherwise. Note that $\varepsilon_j(\boldsymbol{\beta}) - \varepsilon_i(\boldsymbol{\beta}) = 0$ when $i = j$ so that the value of b_{ii} is essentially arbitrary. Practically speaking then, these weights are constant weights. Furthermore, as mentioned in Section 1, these weights yield the well-known rank-based Wilcoxon (WIL) estimate. Wilcoxon procedures are typically more efficient than least squares procedures when the data are non-normal and feature 95.5% efficiency when the data are normally distributed (e.g. HM (1998, p.163)). However, the influence function of the Wilcoxon estimate is only bounded for the response

and not the design point (e.g. HM (1998, p.164)). Thus, Wilcoxon estimates are not robust against outlying points in the design. This, of course, is irrelevant when the data are obtained from a designed experiment.

2.2. Theil weights

When it comes to discussing outlier resistant estimates for the simple linear regression model many nonparametric textbooks present the median of the pairwise slopes (e.g. Theil 1950) as an estimate of β. For example, the books by Conover (1999), Daniel (1990), and Hollander and Wolfe (1999) discuss this estimator. Now, suppose the weights in (3) are defined by $b_{ij} = |X_j - X_i|^{-1}$; ignoring the possibility of ties for the sake of simplicity. Then, as shown in Terpstra *et al.* (2001), the minimum of (3) corresponds to Theil's estimator. Thus, we see that Theil's estimator is a member of the class of WW-estimates. From this perspective then, a generalization of Theil's estimator to the case where $p > 1$ can be obtained by letting $b_{ij} = \|\boldsymbol{X}_j - \boldsymbol{X}_i\|^{-1}$ where $\|\cdot\|$ represents the Euclidean norm. Note that these weights are of the form $b_{ij} = b(\boldsymbol{X}_i, \boldsymbol{X}_j)$ for some real-valued function $b(\cdot)$. That is, this weighting scheme is a member of the Mallows class.

Naranjo and Hettmansperger (1994) derived both the influence function and breakdown point of the Mallows-based WW-estimate. These results are also stated as Theorems 5.7.1 and 5.7.3 respectively in HM (1998). The theorems imply that Theil's estimator is a bounded influence estimator that attains a breakdown point of 1/3. Hence, we see that the Theil estimator is robust.

2.3. GR weights

Another Mallows-based weighting scheme is defined by $b_{ij} = h_i h_j$ where $h_i = h(\boldsymbol{X}_i)$ and

$$h(\boldsymbol{X}_i) \quad = \quad \min\left\{1, \left[\frac{c}{d_i^2(\boldsymbol{X}_i)}\right]^{k/2}\right\}. \tag{4}$$

Here, c and k correspond to tuning constants and $d_i^2(\boldsymbol{X}_i)$ denotes the squared Mahalanobis distance of $\boldsymbol{X}_i$ based on some (robust) measure of location and dispersion for the design set $\{\boldsymbol{X}_i\}$. For example, our default implementation calculates $d_i^2(\boldsymbol{X}_i)$ using the fast minimum covariance determinant estimates proposed by Rousseeuw and Van Driessen (1999). These estimates are available in R through the **lqs** (R 1.8.1 and earlier) and **MASS** (R 1.9.0 and later) packages written by Brian Ripley. For the tuning constants, we use $c = \chi^2_{0.95}(p)$, the 95th percentile of a $\chi^2(p)$ distribution, and $k = 2$.

These weights have been studied extensively in the context of linear regression. The interested reader is referred to Naranjo and Hettmansperger (1994); Naranjo, McKean, Sheather, and Hettmansperger (1994); McKean, Naranjo, and Sheather (1996b) and Chapter 5 of HM (1998). Once more, it follows from the results of Naranjo and Hettmansperger (1994) that these weights admit a bounded influence function and a positive breakdown point. See also McKean, Naranjo, and Sheather (1996a). We follow the convention in the literature and refer to this particular WW-estimate as a generalized rank (GR) estimate. Lastly, note that there is a fundamental difference between the GR-estimate and the Theil estimate. That is, the weights for the GR-estimate are factored, and the weights for the Theil estimate are not factored.

2.4. HBR weights

These weights also yield robust estimates but typically have higher efficiency than the Theil and GR estimates. More specifically, let

$$b_{ij} = \psi\left(\left|\frac{b}{a_i a_j}\right|\right), \quad \text{where} \quad a_i = \frac{\varepsilon_i(\widehat{\boldsymbol{\beta}}_0)}{\widehat{\sigma}\psi(\chi^2_{0.95}(p)/d_i^2(\boldsymbol{X}_i))}, \tag{5}$$

$d_i^2(\cdot)$ is defined in (4), and $\psi(t) = (t - \text{sgn}\,(t))I(-1 < t < 1) + \text{sgn}\,(t)$. The tuning constant, b, is set at $[\text{med}\{a_i\} + 3\text{MAD}\{a_i\}]^2$ and

$$\widehat{\sigma} = \text{MAD}\{\varepsilon_i(\widehat{\boldsymbol{\beta}}_0)\} = 1.483\text{med}|\varepsilon_i(\widehat{\boldsymbol{\beta}}_0) - \text{med}\{\varepsilon_j(\widehat{\boldsymbol{\beta}}_0)\}|.$$

Lastly, $\varepsilon_i(\widehat{\boldsymbol{\beta}}_0)$ denotes the ith residual based on an initial estimate. Note that these weights incorporate information from the design points and the responses via the initial residuals. Our default implementation uses the fast least trimmed squares estimator proposed by Rousseeuw and Van Driessen (2002) for $\widehat{\boldsymbol{\beta}}_0$. Again, this estimate is available in R via the **lqs** or **MASS** package. The weights in (5) were suggested by Chang *et al.* (1999) and are used in Section 5.8.1 of HM (1998). This particular WW-estimate is referred to as the HBR-estimate since it acquires a 50% breakdown point provided the initial estimates (i.e. $\widehat{\boldsymbol{\mu}}$, $\widehat{\boldsymbol{\Sigma}}$, and $\widehat{\boldsymbol{\beta}}_0$) are high breakdown (50%) estimates.

3. Theoretical results

This section summarizes some of the main theoretical results pertaining to the estimation, inference, and diagnostic procedures that we have chosen to implement. In general, WW-estimates do not exist in closed form. Thus, it is not universally possible to determine exact distributions of estimates and/or test statistics. Expectedly then, all of the results presented in this section are asymptotic in nature. For the sake of brevity, we refer the reader to appropriate references for the technical details.

3.1. Asymptotic distributions

Recall that $D_{WR}(\boldsymbol{\beta})$ is invariant to location and therefore, β_0 can not be directly estimated via the minimization. As an estimate then, we use $\widehat{\beta}_0 = \text{med}\{Y_i - \boldsymbol{X}_i^\intercal \widehat{\boldsymbol{\beta}}_{WR}\}$ where $\widehat{\boldsymbol{\beta}}_{WR}$ denotes a minimum of (3). Note that this is essentially a robust analog of the least squares estimate where the mean of the residuals is used as an estimate of the intercept. See, for example, Section 3.5 of HM (1998). In what follows we let $\widehat{\boldsymbol{\theta}} = (\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}_{WR}^\intercal)^\intercal$ denote the joint estimated parameter vector.

The main result is that $\widehat{\boldsymbol{\theta}}$ is asymptotically normal. That is, $\sqrt{n}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{d} \text{N}(\boldsymbol{0}, \boldsymbol{\Omega})$ where $\boldsymbol{\Omega}$ has the general form

$$\boldsymbol{\Omega} = \begin{bmatrix} \tau_s^2 + \tau^2 \overline{\boldsymbol{x}}^\intercal \boldsymbol{C}^{-1}\boldsymbol{V}\boldsymbol{C}^{-1}\overline{\boldsymbol{x}} & -\tau^2 \overline{\boldsymbol{x}}^\intercal \boldsymbol{C}^{-1}\boldsymbol{V}\boldsymbol{C}^{-1} \\ -\tau^2 \boldsymbol{C}^{-1}\boldsymbol{V}\boldsymbol{C}^{-1}\overline{\boldsymbol{x}} & \tau^2 \boldsymbol{C}^{-1}\boldsymbol{V}\boldsymbol{C}^{-1} \end{bmatrix}. \tag{6}$$

Here, $\overline{\boldsymbol{x}}$ denotes the $p \times 1$ vector of column means corresponding to the $n \times p$ design matrix $\boldsymbol{X}$ and $\tau_s = (2f(0))^{-1}$. The remaining components (i.e. τ, $\boldsymbol{C}$, and $\boldsymbol{V}$) depend on the type of weights that are used.

For Mallows weighting schemes define $\boldsymbol{W}$ to be the $n \times n$ matrix whose elements are

$$w_{ij} \;=\; \begin{cases} -\frac{1}{n} b_{ij} & ; \;\; i \neq j \\ \frac{1}{n} \sum_{k=1}^{n} b_{ik} & ; \;\; i = j \end{cases} \tag{7}$$

where b_{ii} is defined to be zero. Note that both $\boldsymbol{W}$ and $\boldsymbol{X}$ depend on n. Then, the quantities of $\boldsymbol{\Omega}$ are defined as follows:

$$\boldsymbol{C} \;=\; \lim_{n\to\infty} \frac{1}{n} \boldsymbol{X}^{\intercal} \boldsymbol{W} \boldsymbol{X}, \qquad \boldsymbol{V} \;=\; \lim_{n\to\infty} \frac{1}{n} \boldsymbol{X}^{\intercal} \boldsymbol{W}^2 \boldsymbol{X}, \tag{8}$$

and $\tau = (\sqrt{12}\mathsf{E}\left[f(\varepsilon_1)\right])^{-1}$. The details regarding this result can be found in Sievers (1983) and/or Section 5.2 of HM (1998). We note that when Wilcoxon weights are used,

$$\boldsymbol{C} \;=\; \boldsymbol{V} \;=\; \lim_{n\to\infty} \frac{1}{n} \boldsymbol{X}_c^{\intercal} \boldsymbol{X}_c$$

where $\boldsymbol{X}_c$ denotes the centered design matrix. Now, for practical applications an estimate of $\boldsymbol{\Omega}$ is needed. Estimates of $\boldsymbol{C}$ and $\boldsymbol{V}$ correspond to (8) without the limits. For the scale parameters, we have implemented the confidence interval estimate discussed on pages 7–8 and 25–26 of HM (1998) for τ_s and the density estimate presented in Section 3.7.1 of HM (1998) for τ.

Next, let us consider Schweppe weights; and recall that these weights are random since they depend on the response variable. Essentially, this is what is responsible for changing the definitions of τ, $\boldsymbol{C}$, and $\boldsymbol{V}$ from those given in (8). Here, we have $\tau = 1/2$. However, for $\boldsymbol{C}$ and $\boldsymbol{V}$ we need to define the following quantities:

$$\begin{aligned} B_{ij}(t) \;&=\; \mathsf{E}\left[b_{ij} I(0 < Y_i - Y_j < t)\right], \\ \gamma_{ij} \;&=\; B_{ij}'(0) / \mathsf{E}\left[b_{ij}\right], \\ \boldsymbol{C}_n \;&=\; \frac{1}{n^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \gamma_{ij} b_{ij} (\boldsymbol{X}_j - \boldsymbol{X}_i)(\boldsymbol{X}_j - \boldsymbol{X}_i)^{\intercal}, \quad \text{and} \end{aligned} \tag{9}$$

$$\boldsymbol{U}_i \;=\; \frac{1}{n} \sum_{j=1}^{n} (\boldsymbol{X}_j - \boldsymbol{X}_i) \mathsf{E}\left[b_{ij} \mathrm{sgn}\left(Y_j - Y_i\right) \mid Y_i\right]. \tag{10}$$

Then, $\boldsymbol{C} = \mathrm{plim}_{n\to\infty} \boldsymbol{C}_n$ and $\boldsymbol{V} = \lim_{n\to\infty} (1/n) \sum_{i=1}^{n} \mathsf{VAR}\left[\boldsymbol{U}_i\right]$. The details regarding this result can be found in Chang *et al.* (1999) and/or Section 5.8 of HM (1998). Estimates of these quantities are also discussed in these references.

3.2. Tests of linear hypotheses

Consider testing a general linear hypothesis of the form

$$H_0 : \boldsymbol{A}\boldsymbol{\beta} = \boldsymbol{0} \quad \text{versus} \quad H_1 : \boldsymbol{A}\boldsymbol{\beta} \neq \boldsymbol{0} \tag{11}$$

where $\boldsymbol{A}$ is a $q \times p$ hypothesis matrix with rank q. For example, when $\boldsymbol{A}$ is the $p \times p$ identity matrix, H_1 corresponds to *regression significance*. We only consider tests of the regression parameters as these are typically the focus in regression analyses.

The first test is based on a standardization of the full model estimate and is typically referred to as a Wald test. More specifically, let $\widehat{\boldsymbol{\beta}}_{WR}$ denote a WW-estimate obtained from minimizing

(3) and let $\widehat{\boldsymbol{\Omega}}_{2,2}$ denote a consistent estimate of $\tau^2 \boldsymbol{C}^{-1}\boldsymbol{V}\boldsymbol{C}^{-1}$, which is defined in (6). Then, an approximate α level test of (11) is given by: reject H_0 if

$$\widehat{W}^2 \quad = \quad n\left(\boldsymbol{A}\widehat{\boldsymbol{\beta}}_{WR}\right)^{\intercal}\left(\boldsymbol{A}\widehat{\boldsymbol{\Omega}}_{2,2}\boldsymbol{A}^{\intercal}\right)^{-1}\left(\boldsymbol{A}\widehat{\boldsymbol{\beta}}_{WR}\right) \tag{12}$$

is larger than $\chi^2_{1-\alpha}(q)$. However, finite sample simulation studies suggest that a better test is given by: reject H_0 if $\widehat{W}^2/q > F_{1-\alpha}(q, n-p-1)$ where $F_{1-\alpha}(q, n-p-1)$ corresponds to the $(1-\alpha)100\%$ percentile of a F distribution with q and $n-p-1$ degrees of freedom. See, for example, McKean and Sheather (1991) and/or Section 3.6 of HM (1998). Note that this test can be modified to include the intercept term should one choose to do so.

The second test, typically referred to as a *drop in dispersion* test, is based on both the full model estimate (say $\widehat{\boldsymbol{\beta}}_f$) and a reduced model estimate (say $\widehat{\boldsymbol{\beta}}_r$). That is, the reduced model estimate minimizes (3) subject to the linear constraints in (11). This can be accomplished using a QR decomposition of the matrix $\boldsymbol{A}^{\intercal}$. Our discussion here is limited to Mallows schemes since this is the extent of the current (at least to our knowledge) theoretical development; see, for example, Theorem 5.2.12 of HM (1998). Nevertheless, in principle, the same idea is also applicable to Schweppe weights.

Now, once the reduced and full model estimates are obtained, the drop in dispersion test statistic is given by

$$SRD \quad = \quad \frac{\sqrt{12}}{n\widehat{\tau}}\left(D_{WR}(\widehat{\boldsymbol{\beta}}_r) - D_{WR}(\widehat{\boldsymbol{\beta}}_f)\right) \tag{13}$$

where $D_{WR}(\cdot)$ denotes the dispersion function in (3) and $\widehat{\tau}$ is a consistent estimate of τ. Next, let $\boldsymbol{C}$ and $\boldsymbol{V}$ be the matrices defined in (8), let $\boldsymbol{C}_r$ denote the reduced model analog of $\boldsymbol{C}$, and define

$$\boldsymbol{C}^{+} \quad = \quad \begin{bmatrix} \boldsymbol{C}_r^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Then, $SRD \xrightarrow{d} \sum_{i=1}^{q} \lambda_i \chi_i^2(1)$ where $\lambda_1, \lambda_2, \ldots, \lambda_q$ are the q positive eigenvalues of $\boldsymbol{V}(\boldsymbol{C}^{-1} - \boldsymbol{C}^{+})$ and $\chi_1^2(1), \chi_2^2(1), \ldots, \chi_q^2(1)$ are iid χ^2 random variables, each with one degree of freedom. To obtain a p-value, Hettmansperger and McKean (1998, p.288) suggest either bootstrapping (see e.g. Section 6.1) the test statistic or simulating the sum of weighted χ^2 distributions.

Our current implementation only considers Wilcoxon weights (i.e. $b_{ij} = 1$). For these weights it is readily shown that the q positive eigenvalues are all equal to one, so the limiting distribution is χ^2 with q degrees of freedom. However, like the Wald test, finite sample simulation studies suggest that a better test is given by: reject H_0 if $F_R = SRD/q > F_{1-\alpha}(q, n-p-1)$. See, for example, McKean and Sheather (1991) and/or Section 3.6 of HM (1998). Lastly, it should be pointed out that Hettmansperger and McKean (1983) compared the performances of F_R and $\widehat{W}^2/q$ via small sample simulation studies. In short, their findings indicate that F_R exhibits a more stable Type I error rate and slightly dominates $\widehat{W}^2/q$ in terms of power. This is consistent with the simulation study presented in Section 6.2 of this paper.

3.3. Studentized residuals

Let $\widehat{\varepsilon}_i = Y_i - \widehat{\beta}_0 - \boldsymbol{X}_i^{\intercal}\widehat{\boldsymbol{\beta}}_{WR}$ denote the ith residual where $\widehat{\beta}_0 = \text{med}\{Y_i - \boldsymbol{X}_i^{\intercal}\widehat{\boldsymbol{\beta}}_{WR}\}$. For general residual analyses it is desirable to know the variance of $\widehat{\varepsilon}_i$, say σ_i^2. Then, the studentized

residual, which is often used for outlier identification, is defined as $\widehat{\varepsilon}_i/\widehat{\sigma}_i$. A general rule of thumb is to declare the ith observation a potential outlier if the absolute value of the studentized residual is larger than two.

In what follows we let $\widehat{\boldsymbol{\varepsilon}}_M$ and $\widehat{\boldsymbol{\varepsilon}}_S$ denote $n \times 1$ vectors which contain the residuals for Mallows and Schweppe weights, respectively. Then, a first order approximation of $\mathsf{VAR}\,[\widehat{\boldsymbol{\varepsilon}}_M]$ is given by

$$\mathsf{VAR}\,[\widehat{\boldsymbol{\varepsilon}}_M] \doteq \sigma^2 \boldsymbol{I} - K_3 \boldsymbol{J} - (K_4 \boldsymbol{I} - K_5 \boldsymbol{J}) \boldsymbol{K}_w^\intercal + \tau^2 \boldsymbol{K}_w \boldsymbol{K}_w^\intercal \tag{14}$$

where $\sigma^2 = \mathsf{VAR}\,[\varepsilon_1]$, $\boldsymbol{I}$ is the $n \times n$ identity matrix, $\boldsymbol{J} = (1/n)\mathbf{1}\mathbf{1}^\intercal$, $\mathbf{1}$ is an $n \times 1$ vector of ones, $\boldsymbol{K}_w = \boldsymbol{X}_c(\boldsymbol{X}_c^\intercal \boldsymbol{W} \boldsymbol{X}_c)^{-1} \boldsymbol{X}_c^\intercal \boldsymbol{W}$,

$$\begin{array}{lll} K_3 = 2\tau_s\delta_3 - \tau_s^2, & K_4 = \sqrt{12}\tau\xi, & K_5 = \sqrt{12}\tau\tau_s\delta_5, \\ \delta_3 = \mathsf{E}\,[\varepsilon_1 \mathrm{sgn}\,(\varepsilon_1)]\,, & \delta_5 = \mathsf{E}\,[\mathrm{sgn}\,(\varepsilon_1)\,\mathrm{sgn}\,(\varepsilon_1 - \varepsilon_2)]\,, & \text{and} \quad \xi = \mathsf{E}\,[\varepsilon_1 \mathrm{sgn}\,(\varepsilon_1 - \varepsilon_2)]\,. \end{array} \tag{15}$$

See, for example, Naranjo *et al.* (1994) and Section 5.4 of HM (1998). We note that (14) is valid for any Mallows weighting scheme. In particular, (14) holds for the three weighting schemes discussed in Sections 2.1, 2.2, and 2.3; the only quantity that changes is the $\boldsymbol{W}$ matrix, which appears in $\boldsymbol{K}_w$. Now, in practical applications we need estimates of the quantities defined in (15). For $\widehat{\sigma}$ we use $\mathrm{MAD}\{\widehat{\boldsymbol{\varepsilon}}_{M,i}\}$, where MAD represents the median absolute difference estimator of scale. Estimates of τ_s and τ were discussed in conjunction with (8). For the remaining quantities we use the residual-based moment estimators. See, for example, Section 5.4 of HM (1998).

The Schweppe weights version of (14) is similar and is given by

$$\begin{aligned} \mathsf{VAR}\,[\widehat{\boldsymbol{\varepsilon}}_S] &\doteq \sigma^2 \boldsymbol{I} + \tau_s^2 \boldsymbol{J} + \frac{1}{4} \boldsymbol{X}_c \boldsymbol{C}^{-1} \boldsymbol{V} \boldsymbol{C}^{-1} \boldsymbol{X}_c^\intercal - 2\tau_s\kappa_1 \boldsymbol{J} \\ &- \sqrt{12}\tau\kappa_2 \left\{ \boldsymbol{A} \boldsymbol{X}_c \boldsymbol{C}^{-1} \boldsymbol{X}_c^\intercal + \boldsymbol{X}_c \boldsymbol{C}^{-1} \boldsymbol{X}_c^\intercal \boldsymbol{A} \right\}. \end{aligned} \tag{16}$$

See, for example, Chang *et al.* (1999) and Section 5.9 of HM (1998). Here, $\boldsymbol{C}$ and $\boldsymbol{V}$ correspond to the matrices defined in conjunction with (9) and (10) respectively, $\kappa_1 = \mathsf{E}\,[|\varepsilon_1|]$, and $\kappa_2 = \mathsf{E}\,[\varepsilon_1(2F(\varepsilon_1) - 1)]$. Residual-based moment estimates of κ_1 and κ_2 can be obtained by replacing F with the empirical cumulative distribution function of the residuals. The other quantities, namely σ^2, τ_s, τ, $\boldsymbol{I}$, and $\boldsymbol{J}$ are identical to those defined for Mallows weights. Finally, $\boldsymbol{A} = n(\sqrt{12}\tau)^{-1}\boldsymbol{W}$ where $\boldsymbol{W}$ corresponds to the Schweppe analog of (7). Once again, we note that (16) holds for any Schweppe weighting scheme since the only quantities that depend on the weights are $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{V}$.

3.4. Diagnostics for comparing fits

It is clear from Section 2 that different weighting schemes yield different estimates. In fact, it follows from Lemma 5.2.11 of HM (1998) that the most efficient WW-estimate corresponds to the Wilcoxon estimate. Recall that all points are equally weighted for this particular estimate. However, equal weights do not always make sense for those data sets which contain outliers. Thus, it is desirable to have a diagnostic that compares the Wilcoxon estimate ($b_{ij} = 1$ for all (i, j)) to a WW-estimate ($b_{ij} \neq 1$ for some (i, j)). Actually, in principle, the diagnostics in this section can be used to compare any two WW-estimates (e.g. GR and HBR). However, to our knowledge, no studies directly address this problem. Nevertheless, in view of the theory presented in Section 5.5 of HM (1998), such a practice can be justified.

In what follows we let $\widehat{\boldsymbol{\theta}}_R = (\widehat{\beta}_{R0}, \widehat{\boldsymbol{\beta}}_R^\intercal)^\intercal$ denote the parameter estimates based on Wilcoxon weights. An analogous estimate, say $\widehat{\boldsymbol{\theta}}_{WR}$, will be defined for any WW-estimate whose weights are not all equal to one. Then, the two estimates can be compared using the following diagnostic

$$TDBETAS_R = (\widehat{\boldsymbol{\theta}}_R - \widehat{\boldsymbol{\theta}}_{WR})^\intercal \widehat{\boldsymbol{\Omega}}_R^{-1} (\widehat{\boldsymbol{\theta}}_R - \widehat{\boldsymbol{\theta}}_{WR}) \tag{17}$$

where $\widehat{\boldsymbol{\Omega}}_R$ is an estimate of the Wilcoxon version of (6). A more thorough discussion of this diagnostic can be found in McKean *et al.* (1996a), McKean *et al.* (1996b) and Section 5.5 of HM (1998). These references suggest using $4(p+1)^2/n$ as a benchmark for declaring the fits to be different.

Now, when two fits are declared to be different it is sometimes desirable to investigate the nature of the difference. This can be accomplished by comparing the individual fits for the two estimates. For example, let $\widehat{Y}_{R,i} = \widehat{\beta}_{R0} + \boldsymbol{X}_i^\intercal \widehat{\boldsymbol{\beta}}_R$ denote the Wilcoxon fit for the ith case and let $\widehat{Y}_{WR,i}$ denote the ith fit for the other WW-estimate. A diagnostic which compares these fits is given by

$$CFITS_{R,i} = \frac{\widehat{Y}_{R,i} - \widehat{Y}_{WR,i}}{\sqrt{\frac{1}{n}\widehat{\tau}_s^2 + \boldsymbol{X}_{ci}^\intercal (\boldsymbol{X}_c^\intercal \boldsymbol{X}_c)^{-1} \boldsymbol{X}_{ci}}} \tag{18}$$

where the denominator of (18) corresponds to the estimated standard error of $\widehat{Y}_{R,i}$. See, for example, Section 5.5 of HM (1998). Hettmansperger and McKean (1998, p.303) suggest using $2\sqrt{(p+1)/n}$ as a benchmark for declaring two individual fits different. We note that these diagnostics are designed to distinguish differences between fits and therefore, do not necessarily provide any information regarding which fit is best. Instead, we recommend a standard residual analysis, based on the studentized residuals of Section 3.3, for this endeavor.

4. R code

4.1. Web resources

In this section we give a brief description of some of the R functions that can be used to perform a weighted Wilcoxon analysis. However, we begin by listing some important web sites related to our software.

http://CRAN.R-project.org/. This is the main web site for the R statistical software package (R Development Core Team 2005). Specifically, this is where one can find information on downloading, installing, and updating R. An abundance of other related material can also be found here. For instance, information on the **quantreg**, **lqs**, and **MASS** packages, which are required to run our functions, is available under the `Packages` link. See also the R help files corresponding to `install.packages` and `library`.

http://www.stat.wmich.edu/mckean/HMC/Rcode. Our entire suite of functions is contained in one file (`ww.r`) and can be downloaded from this site. Alternatively, one can source the code directly by supplying the appropriate URL as an argument to the R `source` function.

See, for example, the R help file corresponding to `source`. There is also a help file (`wwhlp.r`) that can be downloaded. This file contains a detailed description of each function along with some examples which illustrate usage. Lastly, two of the subdirectories correspond to code that is related to Sections 6.1 and 6.2, respectively.

`http://www.stat.wmich.edu/mckean/book/data/datasets.html`. The data sets pertaining to the examples in Sections 5.1 and 5.2 can be downloaded from here. In fact, many of the data sets in HM (1998) are available from this web site.

4.2. Descriptions of R functions

`wwest`. This function performs a weighted Wilcoxon analysis using the weighting schemes discussed in Section 2. As input, this function requires a design matrix, a response vector, and a name for the weighting scheme. Arbitrary Mallows weights are also allowed. Specifically, output is produced which summarizes a test of regression significance along with tests on individual parameters. A graphical residual analysis can also be obtained.

`wwfit`. As input, this function requires a design matrix, a response vector, and an $n(n-1)/2 \times 1$ vector of weights. Note that this function is not limited to the weighting schemes discussed in Section 2. It then minimizes the dispersion function in (3) via the L_1 procedure discussed in Section 2. The return value is a list which contains the parameter estimates, residuals, and a weight matrix.

`wilwts`. This function returns an $n(n-1)/2 \times 1$ vector of ones. These weights correspond to the Wilcoxon estimate.

`theilwts`. This function returns the $n(n-1)/2 \times 1$ vector of Theil weights discussed in Section 2.2. When $p = 1$ the slope estimate corresponds to the median of the pairwise slopes.

`grwts`. This function returns the $n(n-1)/2 \times 1$ vector of GR weights defined in (4). The function is flexible enough to accommodate arbitrary distances and tuning constants.

`hbrwts`. This function returns the $n(n-1)/2 \times 1$ vector of HBR weights defined in (5). The function is flexible enough to accommodate arbitrary distances, initial estimates, and tuning constants.

`wts`. This is a wrapper function that essentially calls one of the above weight functions. Typically, it is only used in conjunction with `wwfit`. For example, in simulation studies where only the estimates are being studied, it is not necessary to evaluate the extra output produced by `wwest`.

`wilcoxontau`. This function calculates the density estimate of $\tau = (\sqrt{12}\mathsf{E}[f(\varepsilon_1)])^{-1}$ discussed in Section 3.7.1 of HM (1998). It is a scale parameter estimate which appears in the Mallows-based variance-covariance matrix defined in (6).

`taustar.` This function calculates the confidence interval estimate of $\tau_s = (2f(0))^{-1}$ discussed on pages 7-8 and 25-26 of HM (1998). It corresponds to the asymptotic standard deviation of the median-based estimate of the intercept parameter.

`varcov.gr.` This function calculates an estimate of the variance-covariance matrix for the regression coefficients (including the intercept) when Mallows weights are used. In particular, it can be used to determine the variance-covariance matrix of the Wilcoxon estimate. It returns several components associated with the matrix. See, for example, (6), (7), and (8).

`varcov.hbr.` This function calculates an estimate of the variance-covariance matrix for the regression coefficients (including the intercept) when Schweppe weights are used. It returns several components associated with the matrix. See, for example, (6), (9), and (10).

`wald.` This function calculates the Wald statistic defined in (12) and the corresponding p-value for the hypotheses given in (11). It requires both the intercept parameter estimate and the regression parameter estimates. The use of the F distribution for calculating p-values is documented in McKean and Sheather (1991) and Section 3.6 of HM (1998).

`droptest.` This function performs a drop in dispersion test for the general linear hypotheses defined in (11). The test statistic is defined after (13). Our current implementation requires that Wilcoxon weights (i.e. $b_{ij} = 1$) be used for the analysis. Again, the use of the F distribution for calculating p-values is documented in McKean and Sheather (1991) and Section 3.6 of HM (1998).

`redmod.` This function obtains the reduced model design matrix used by `droptest`. The calculation of this matrix is based on a QR decomposition of $\boldsymbol{A}^\intercal$ where $\boldsymbol{A}$ is defined in (11). See, for example, Theorem 3.7.2 of HM (1998).

`regrtest.` This function performs a Wilcoxon-based drop in dispersion test of regression significance (i.e. $H_1 : \boldsymbol{\beta} \neq \mathbf{0}$).

`cellmntest.` This function performs a Wilcoxon-based drop in dispersion test and returns a robust ANOVA table for the one-way analysis of variance model. The interested reader is referred to Chapter 4 of HM (1998) for the details.

`pwcomp.` This function is typically used in conjuntion with `cellmntest`. It examines all C_2^p pairwise comparisons using Wilcoxon-based drop in dispersion tests. Note that this is similar in nature to the protected LSD method. See, for example, Kuehl (2000, p.111).

`studres.gr.` This function calculates the studentized residuals corresponding to Mallows-based weighted Wilcoxon estimates (see (14)). These residuals can be used to construct residual plots and flag potential outliers. For instance, if the absolute value of the studentized residual exceeds two then one might declare the observation an outlier.

`studres.hbr`. This function is analogous to `studres.gr` except that it corresponds to WW-estimates based on Schweppe weights (see (16)).

`fitdiag`. This function returns the diganostics $TDBETAS_R$ and $CFITS_{R,i}$ which are defined in (17) and (18), respectively. The current implementation allows the following comparisons: WIL vs. GR, WIL vs. HBR, GR vs. HBR, and WIL vs. LS. All comparisons are standardized using the WIL fit.

`plotfitdiag`. This function produces a casewise plot of $CFITS_{R,i}$ based on the results of `fitdiag`. Information on the total change in fits (i.e. $TDBETAS_R$) is also reported.

5. Data set examples

We now present some data set examples which illustrate the use of our R functions and corresponding output. Other examples can be found in our help file (`wwhlp.r`).

5.1. Hawkins data set

For our first example we analyze the well-known data set constructed by Hawkins, Bradu, and Kass (1984). Briefly, this data set contains 75 observations on one response variable and three predictor variables. The first 10 observations are *bad* leverage points, observations 11-14 are *good* leverage points, and the remaining observations are consistent with the underlying model. This data set is typically used to illustrate the degree of robustness (or lack of) an estimator exhibits toward outlying data points.

The analysis will parallel that given in Section 5.3 of HM (1998); except that we will compare the WIL estimator to the HBR estimator (as opposed to the GR estimator). Figures 1 and 2 pertain to the WIL estimate while Figures 3 and 4 display the HBR results. Figure 5 compares the WIL and HBR fits directly.

To begin, note that `wwest` essentially serves as an all-purpose estimation and inference function. It makes calls to several of the functions described in Section 4.2. In particular, calls to `wwfit`, `wald`, and `regrtest` are made. Note that the type of weighted Wilcoxon analysis is controlled by the `bij` argument.

Figures 1 and 3 display the results for the WIL and HBR fits, respectively. The output is similar to that produced by many statistical software packages. That is, a test of general regression significance along with significance tests for individual model parameters are output. With the exception of the drop in dispersion test for the Wilcoxon estimate (i.e. F_R), all tests are Wald-based (i.e. (12)) procedures. Finally, the user is given an option to view a graphical residual analysis.

Figures 2 and 4 display the residual analyses for the WIL and HBR fits, respectively. The standard residual vs. fit and normal probability plots are given in positions (1,1) and (2,2), respectively. As a compliment to the normal probability plot, a histogram of the residuals appears in position (1,2). Lastly, a case plot of the studentized residuals (recall (14) and (16)) is given in position (2,1). Here, we can use the ± 2 benchmark to help identify potential outliers.

```
> wwest(x=hawk[,1:3],y=hawk[,4],bij="WIL")

Wald Test of H0: BETA1=BETA2=BETA3=0
TS: 419.2269 PVAL: 0

Drop Test of H0: BETA1=BETA2=BETA3=0
TS: 44.312 PVAL: 0

          EST     SE    TVAL   PVAL
BETA0 -0.7758 0.2032 -3.8177 0.0003
BETA1  0.1688 0.1103  1.5312 0.1302
BETA2  0.0180 0.0651  0.2766 0.7829
BETA3  0.2687 0.0541  4.9651 0.0000

Would you like to see residual plots (y/n)?
y
```

Figure 1: Wilcoxon-based `wwest` output for the Hawkins data.

Finally, a few words regarding the WIL and HBR fits are in order. It is clear from Figure 5 that the WIL and HBR fits are quite different; most notably, for the first 14 observations. It is evident from Figure 2 that the WIL fit favors the bad leverage points. For example, both residual plots identify the four good leverage points as outlying observations. On the other hand, the HBR-based residual plots in Figure 4 identify the 10 bad leverage points as outlying observations. Lastly, we note that the inference results in Figures 1 and 3 contradict one another. That is, the WIL test for regression significance is significant while the HBR test is not significant. Actually, all of this comes as no surprise given the well-known fact that WIL procedures are not robust against bad leverage points.

5.2. LDL cholesterol in quail

This data set contains the LDL Cholesterol levels of 39 different quail. The data were obtained using a completely randomized design with four treatments. Each treatment essentially corresponds to a different diet containing a different drug compound. To begin, note that this is a designed experiment. Therefore, the design matrix will not contain any leverage points. It does not make sense then, from an efficiency point of view, to use a weighted Wilcoxon estimate that downweights leverage points (e.g. GR or HBR). In general, we recommend that Wilcoxon-based procedures be used for experimental design situations. See Chapter 4 of HM (1998) for details.

We are interested in testing for a significant treatment (i.e. diet) effect. This can be accomplished using the `cellmntest` function. Figure 6 illustrates the steps involved as well as the corresponding output. Briefly, `cellmntest` performs a WIL-based drop in dispersion test and returns a robust ANOVA table for the one-way analysis of variance model. From Figure 6 we see that $F_R = 3.79$ and the corresponding p-value is 0.0187. Hence, the diets are significantly different at the 0.05 level.

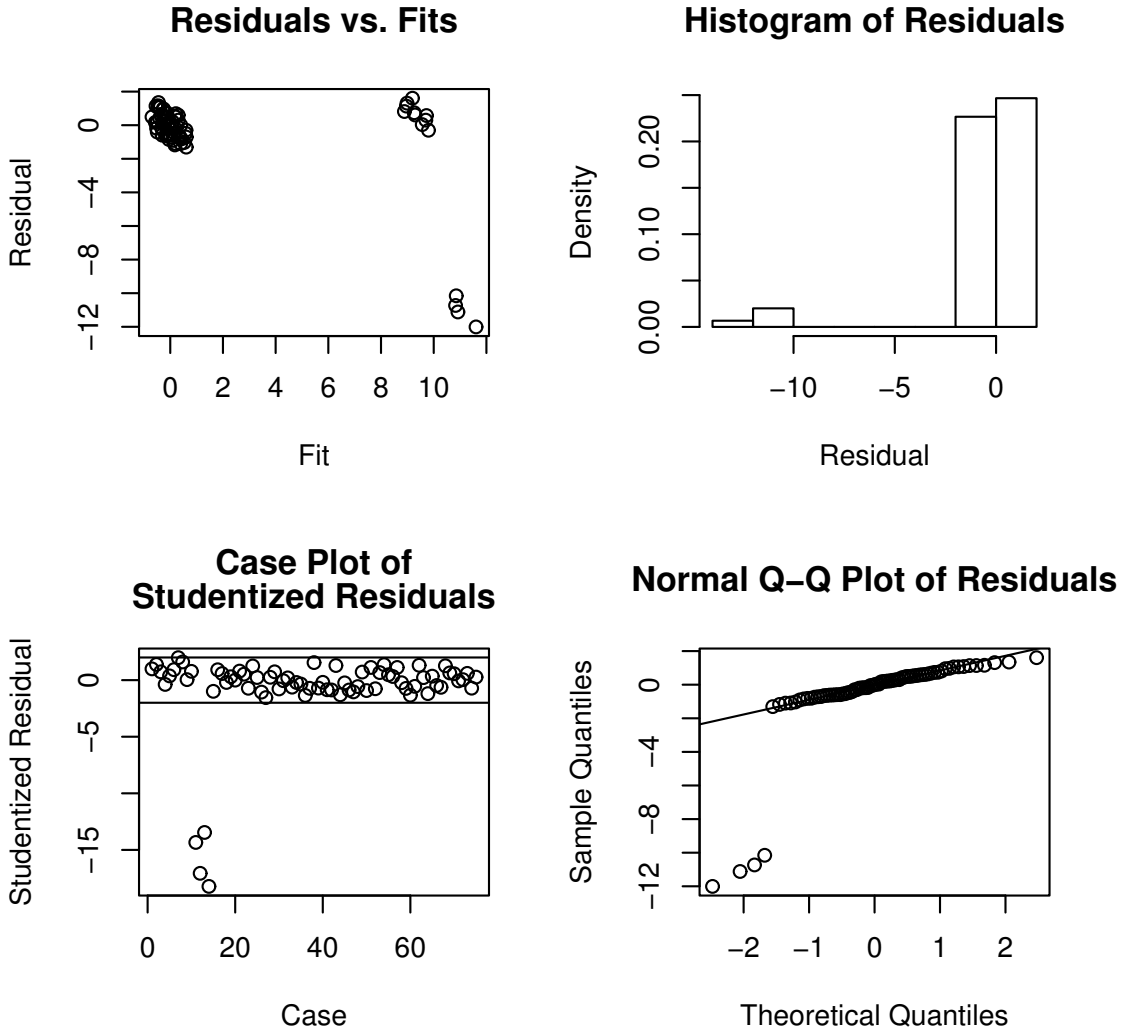

Figure 2: Wilcoxon-based residual analysis for the Hawkins data.

```
> wwest(x=hawk[,1:3],y=hawk[,4],bij="HBR")

Wald Test of H0: BETA1=BETA2=BETA3=0
TS: 0.5651 PVAL: 0.6398

          EST     SE    TVAL   PVAL
BETA0 -0.1547 0.2166 -0.7144 0.4773
BETA1  0.0960 0.1179  0.8145 0.4181
BETA2  0.0385 0.0683  0.5630 0.5752
BETA3 -0.0458 0.0586 -0.7813 0.4372

Would you like to see residual plots (y/n)?
y
```

Figure 3: HBR-based `wwest` output for the Hawkins data.

Now, given a significant result, it is customary to examine the pairwise differences in order to help assess the nature of the rejection. To this end, we note that `cellmntest` has an argument which allows one to test

$$H_0 : \boldsymbol{A}\boldsymbol{\mu} = \mathbf{0} \quad \text{versus} \quad H_1 : \boldsymbol{A}\boldsymbol{\mu} \neq \mathbf{0}$$

where $\boldsymbol{\mu}$ is the $p \times 1$ vector of cell locations and $\boldsymbol{A}$ is an arbitrary $q \times p$ contrast matrix. For example, if $p = 4$, calling `cellmntest` with $\boldsymbol{A} = (1, 0, -1, 0)$ tests for location differences between populations one and three. Thus, we can use `cellmntest` to examine each of the C_2^p pairwise comparisons by defining an appropriate $\boldsymbol{A}$ matrix. When used in this manner we essentially have a Wilcoxon-based drop in dispersion version of the well-known protected LSD method. See, for example, Kuehl (2000, p.111). In fact, this is exactly what the `pwcomp` function is used for. From Figure 6 then, we see that diet number 2 yields the lowest cholesterol levels and this diet is significantly different from the others.

5.3. Residential extensions data

A widely used model in time series analysis is the (stationary) autoregressive model of order p, which we denote as AR(p). In short, an AR(p) model is a linear regression model where the response variable corresponds to the current time series value and the independent variables represent the previous p values of the time series. See, for example, Fuller (1996) for a more thorough description. Thus, from this perspective, our suite of R functions is equally applicable to autoregressive time series analysis. In fact, WW-estimates are particularly appealing because outlying observations play a dual role as both response and explanatory variables. Terpstra *et al.* (2001) provides a good overview of WW-estimates and their application to autoregressive time series modeling.

A widely cited example in the robust time series literature is a monthly time series (RESX), which originated at Bell Canada. The data set can be found in Rousseeuw and Leroy (1987, p.278–280). The series consists of the number of telephone installations in a given region and

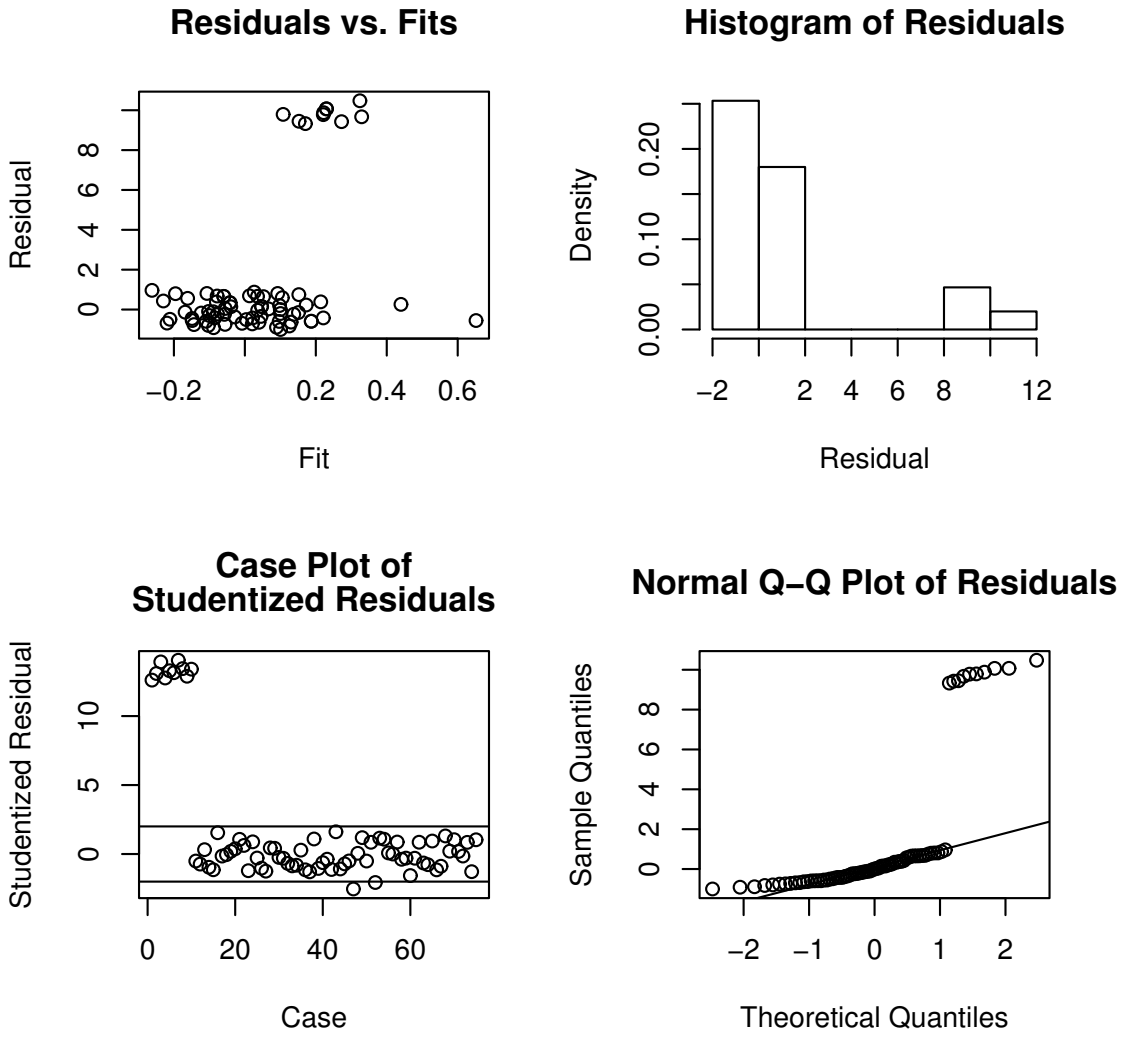

Figure 4: HBR-based residual analysis for the Hawkins data.

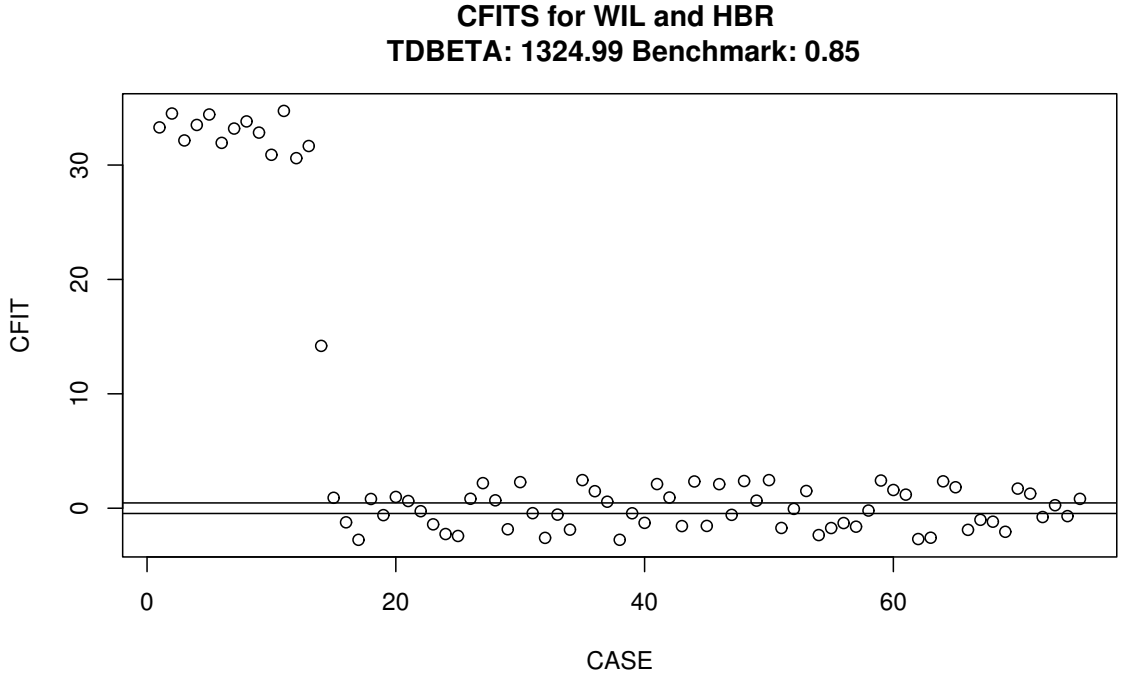

Figure 5: Comparison of the WIL and HBR fits of the Hawkins data.

Table 1: Parameter estimates for the RESX data.

Estimate	$\widehat{\beta}_1$	SE($\widehat{\beta}_1$)	$\widehat{\beta}_2$	SE($\widehat{\beta}_2$)
WIL	0.5032	0.0298	-0.1512	0.0298
GR	0.3605	0.1008	0.4048	0.0956
HBR	0.4126	0.0691	0.2903	0.0755

has two obvious outliers. The outliers are essentially attributed to bargain months where telephone installations were free. Historically, the stationary zero mean AR(2) has been used to model the seasonally differenced series.

The WIL, GR, and HBR parameter estimates and standard errors for the AR(2) model are given in Table 1. Again, we used the function `wwest` to obtain these results. However, we have chosen to suppress the output in the interest of space. Note that there appears to be some discrepancy between the three estimates. In particular, the sign of $\widehat{\beta}_2$ is negative for the WIL estimate and positive for the GR and HBR estimates. This difference between the WIL estimate and the other WW-estimates can provide valuable insight into the type of outliers present.

For example, let $\{Y_t\}$ denote the observed time series where $Y_t = X_t + a_t$, $\{X_t\}$ is an underlying AR(p) time series, and $\{a_t\}$ is an iid sequence of random variables that possess a mixture distribution, say $(1-q)\delta_0 + qH$. Here, q represents the proportion of outliers, δ_0 is a degenerate distribution at zero, and H is some contaminating distribution function. Now, when $q = 0$ this model corresponds to the Type II or Innovation Outlier (IO) model of Fox (1972). For

```
> #The data...
> quail[,1]
 [1] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
[30] 3 4 4 4 4 4 4 4 4 4

> quail[,2]
 [1]  52  67  54  69 116  79  68  47 120  73  36  34  47 125
[15]  30  31  30  59  33  98  52  55  66  50  58 176  91  66
[29]  61  63  62  71  41 118  48  82  65  72  49

> #Test for equal cell locations...
> z=cellmntest(y=quail[,2],levels=quail[,1])

           RD DF     MRD     TS   PVAL
H0    108.611  3 36.2037 3.7896 0.0187
Error         35  9.5535

> #Cell location estimates...
> z$full$coef
[1] 67 42 63 62

> #Pairwise comparisons...
> t(pwcomp(quail[,2],quail[,1]))
      G1-G2  G1-G3  G1-G4  G2-G3  G2-G4  G3-G4
PVAL 0.0031 0.5984 0.5433 0.0131 0.0173 0.8472
```

Figure 6: WIL-based one-way analysis of variance output for the quail data.

the IO model, outliers are introduced through the error distribution corresponding to $\{X_t\}$ and consequently produce good leverage points in the sense that the fit yields a small residual at these points. In Type I or Additive Outlier (AO) model (e.g. $q > 0$) of Fox (1972) outliers do not become part of the underlying model and, as a consequence, result in bad leverage points. Of course, any given time series may contain both IOs and AOs either in isolation and/or patches.

Now, as demonstrated by Terpstra *et al.* (2001), the WIL estimate is highly efficient under an IO model. However, when AOs are present the WIL estimate is not robust and can differ from other WW-estimates; in particular, the GR and HBR estimates. Thus, for time series analysis, it is important to both identify and distinguish between the different types of outliers. Indeed, the $TDBETAS_R$ and $CFITS_{R,i}$ diagnostics discussed in Section 3.4 are suitable for these types of comparisons.

Figure 7 presents these diagnostics for the following three comparisons: WIL vs. GR, WIL vs. HBR, and GR vs. HBR. The figure was created with the `fitdiag` and `plotfitdiag` functions. Based on the benchmark value of 0.48, all three fits appear to be different. However, note that the two $TDBETAS_R$ diagnostics which feature the WIL fit are much larger than the GR vs.

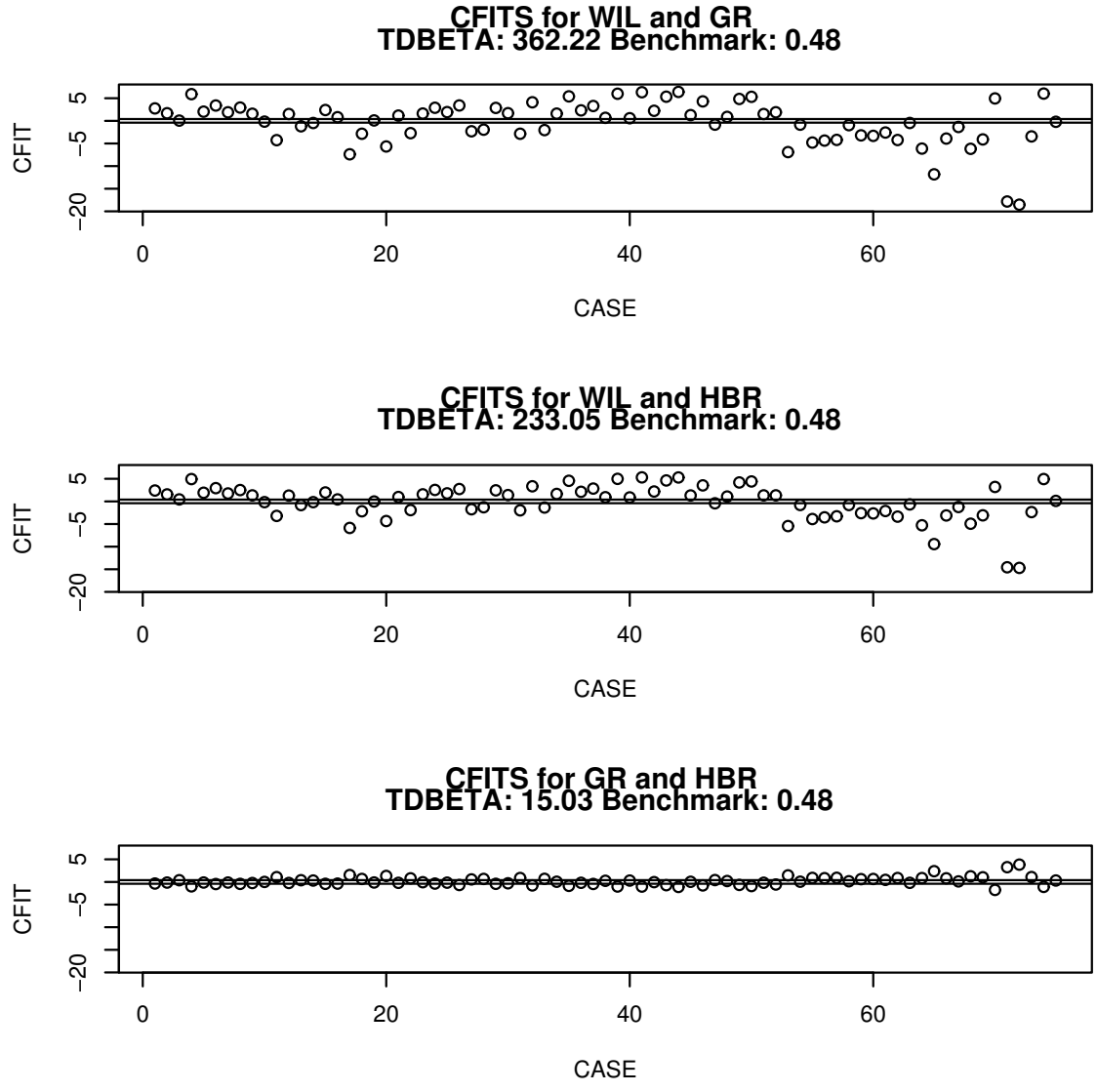

Figure 7: Comparison of WIL, GR, and HBR fits of the RESX series.

HBR $TDBETAS_R$. That is, the GR and HBR fits are much more similar. This indicates the presence of additive outliers. For example, recall that the WIL estimate does not downweight any observations, the GR estimate downweights *all* leverage points, and the HBR estimate attempts to downweight only *bad* leverage points. Since additive outliers typically produce bad leverage points, the GR and HBR estimates will tend to be similar, but different from the WIL estimate. On the other hand, if all fits appear to be similar then any potential outliers will tend to be of the innovation variety. Lastly, we note that the $CFITS_{R,i}$ diagnostics clearly indicate the two previously mentioned outliers.

6. Simulation applications

In this section, we present two statistical applications of our software. We would like to point out that, using our R functions, the effort to code these applications was quite minimal. The functions pertaining to these applications (`bootsim.s`, `polysim1.s`) are also available on our web site. The address is given in Section 4.1.

6.1. The bootstrap

For our first application, we consider bootstrapping the p-value of the test statistic (i.e. F_R) defined after (13). We have chosen to use the bootstrap where the model is rebuilt based on a sample of full model residuals; see Efron and Tibshirani (1993) for a discussion. A brief description of the general algorithm is as follows.

Algorithm 6.1 (Bootstrap algorithm for p-value) *Consider the linear model in (1) and the hypotheses in (11). Let NB denote the number of bootstrap samples.*

(1) *Fit model (1). Then obtain the full model residuals, say $\{\widehat{\varepsilon}_i\}$, and the value of the test statistic, F_R.*

(2) *Set $j = 1$.*

(3) *Obtain a bootstrap sample of size n, say $\{\widehat{\varepsilon}_i^*\}$, by sampling with replacement from $\{\widehat{\varepsilon}_i\}$. Now let $Y_i^* = \widehat{\varepsilon}_i^*$.*

(4) *Fit model (1) using $\{(Y_i^*, \boldsymbol{X}_i^\intercal)\}$ and obtain $F_{R,j}^*$, the test statistic for the hypotheses in (11).*

(5) *If $j < NB$, set $j = j + 1$ and return to step (3); else, stop.*

The bootstrap p-value is then given by

$$p^* \quad = \quad \frac{1}{NB} \sum_{j=1}^{NB} I(F_{R,j}^* \geq F_R)$$

where $I(\cdot)$ denotes the indicator function.

Note that the regression equivariance property of the estimate and step (3) imply that the fitting, and hence testing, is performed under the assumption that H_0 is true. The R code for this bootstrap consists of a wrapper function (`bootsim.s`) around `droptest`.

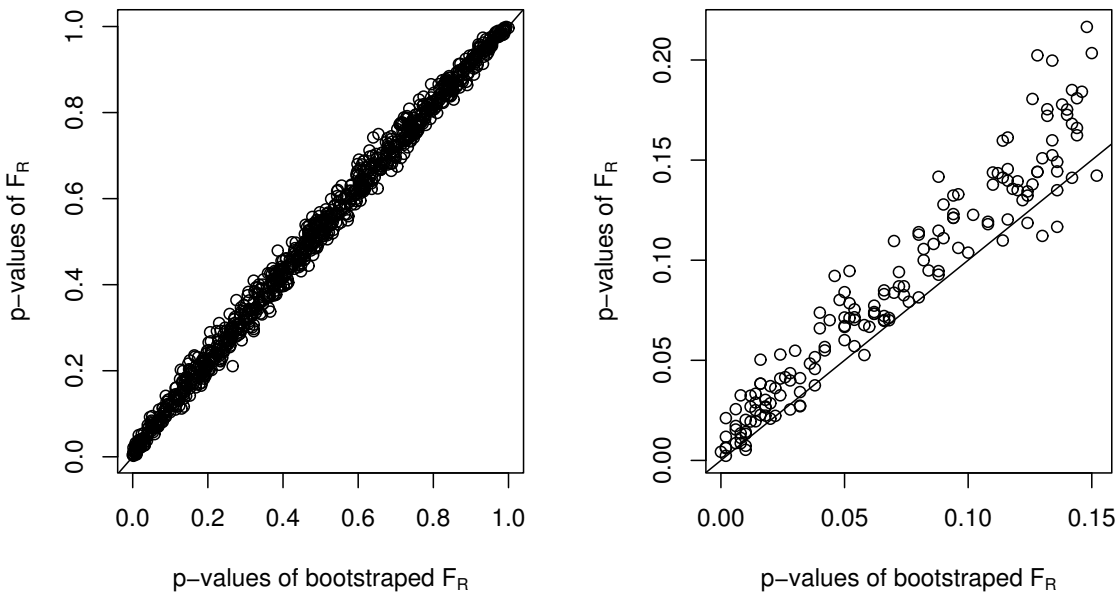

Figure 8: Scatterplot of p-values based on F_R and F_R^* for 1000 simulations with the identity line superimposed.

We used the above algorithm on an example involving generated data from the model

$$Y_i \quad = \quad \beta_0 + \sum_{j=1}^{3} \beta_j X_{ji} + \varepsilon_i$$

where the $\{\varepsilon_i\}$ and $\{X_{ji}\}$, $j = 1, 2, 3$, were iid $N(0, 1)$ variates. We set all of the β_j equal to zero and used $n = 30$ for the sample size. The hypotheses considered were

$$H_0 : \beta_2 = \beta_3 = 0 \quad \text{versus} \quad H_1 : \beta_2 \neq 0 \text{ and/or } \beta_3 \neq 0.$$

The actual data set (`eg.dat`) can be obtained from our web site given in Section 4.1. For the data, the value of F_R was 1.217 with a p-value of 0.312 (based on the approximate F-distribution). Based on $NB = 500$ bootstraps, we calculated the bootstrap p-value to be 0.294.

Next, we simulated the process 1000 times, drawing new data each time. Figure 8 displays the asymptotic p-values versus the bootstrap p-values for the test statistic F_R. In addition, Table 2 displays the empirical α levels for the nominal α values listed. We note that the α levels for the asymptotic version of the test are quite close to the nominal values whereas the α values for the bootstrap version of the test appear to be somewhat liberal. Generally speaking, the plot on the left side of Figure 8 indicates the p-values for the two tests are quite close. However, upon closer inspection (see right side of Figure 8) of the region which contains smaller p-values, we see that the asymptotic p-values tend to be slightly larger

than the bootstrap p-values. In summary though, this simulation suggests that both tests performed reasonably well.

6.2. The order of a polynomial model

In this application, we investigate an algorithm for the determination of the order of a polynomial model. Consider a polynomial model of the form

$$Y_i \quad = \quad \beta_0 + \sum_{j=1}^{p} \beta_j (X_i - \overline{X})^j + \varepsilon_i \tag{19}$$

where $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ are iid with pdf f. Graybill (1976) presents the following algorithm for determining the order (i.e. p) of model (19).

Algorithm 6.2 (Graybill) *Select a super order P so that the true order p is less than or equal to P. Select a significance level α.*

(0) *Set $p = P$.*

(1) *While $p > 0$, fit model (19) with order p.*

(2) *Test the hypotheses $H_0 : \beta_p = 0$ versus $H_1 : \beta_p \neq 0$ at level α.*

(3) *If H_0 is rejected, declare p to be the order and stop; else, set $p = p - 1$ and return to step (1).*

We conducted a pilot study of the powers of five tests for this algorithm: WIL (both the Wald test and the drop in dispersion test), GR, HBR, and LS. Using the functions described in Section 4.2, the coding for the simulation was straightforward. For example, the three weighted Wilcoxon Wald tests involved calls to `wwest` and `wald`, while the drop in dispersion test was performed with the `droptest` function. For LS, we used the `lsfit` and `wald` functions. The R wrapper (`polysim1.s`) to do the simulations can be downloaded from our web site given in Section 4.1.

For the pilot study, we only considered eight situations. Each situation used a sample size of $n = 30$. Four of the situations used $N(0, 1)$ simulated errors, while the remaining four used contaminated normal simulated variates. For the contaminated variates, we set the contamination at 20% and the ratio of standard deviations, contaminated to good, at 10. The regression predictors were iid $N(0, 1)$ variables. The first through fourth situations within each distribution consisted of a polynomial of degree 1 through 4, respectively. We set $\beta_i = 0.4$ for situation i of the normal errors and $\beta_j = 0$ for $j \neq i$. The settings for the contaminated normals were the same, except 0.7 was substituted for 0.4. Finally, we used $P = 4$ for all situations.

Table 2: Empirical α levels for the test statistic F_R and the corresponding bootstrap test.

	Nominal α		
	0.010	0.050	0.100
F_R	0.008	0.051	0.099
F_R^*	0.019	0.067	0.115

We ran 1000 simulations for each situation. Our interest was in how well the algorithm worked for each procedure. Table 3 reports the percentage of times that a particular procedure chose the correct order of the polynomial. The LS test did the best for normal errors, followed closely by the Wilcoxon drop in dispersion test, F_R. For contaminated normal variates, the LS test performed poorly versus the Wilcoxon drop in dispersion test, the Wilcoxon Wald test, and the HBR test. Of the Wilcoxon procedures, the drop in dispersion test did the best. The poor behavior of the GR test confirms the discussion in McKean, Sheather, and Hettmansperger (1994) on the poor efficiency of high breakdown estimators in detecting the curvature in polynomial models. However, note that the HBR estimator recovers much of the efficiency that the GR estimator lost in the presence of curvature; although, in general, it did poorer than the Wilcoxon procedures.

7. Discussion and conclusion

As discussed in this paper, weighted Wilcoxon analyses can range from highly efficient to highly robust, depending on the weights employed. Nevertheless, this class of estimators has yet to be implemented in any mainstream statistical software packages. This paper addresses this issue with a suite of R functions. However, in principle, the algorithm used to obtain these analyses can be adapted to any statistical software package that has L_1 regression capabilities. For example, in S-PLUS, L_1 regression estimates are computed via the `l1fit` function. Therefore, upon making the appropriate substitutions to `wwfit`, one can readily use the functions in S-PLUS as well. In SAS, L_1 regression estimates can be computed using the `IML` procedure and the `LAV` subroutine. Furthermore, the weights defined in (4) and (5) can be computed using calls to `MAD`, `MCD`, and `LTS`.

However, some vigilance is necessary when considering this approach. For example, as discussed in Section 2.1, the Wilcoxon estimate minimizes the L_1 objective function applied to the differences of the residuals. In contrast, the L_1 regression estimate minimizes the sum of the absolute values of the residuals. Thus, the Wilcoxon and L_1 estimators are quite different. As outlined in Section 3.8 of HM (1998), the L_1 estimator is equivalent to an R-estimator using sign scores. Hence, the efficiency properties of the L_1 estimator are the same as those associated with the median and sign test in location problems. In particular, the L_1 estimator for linear models has an asymptotic relative efficiency (ARE) of 0.637 relative to the least squares estimator when the errors have a normal distribution, while the Wilcoxon regression estimator has an ARE of 0.955. Of course, for very heavy tailed data, the L_1 estimator is generally more efficient.

Table 3: Correct order identification percentages.

Errors	Normal				Contaminated Normal			
Situation	1	2	3	4	1	2	3	4
WIL Drop	40.6	59.4	65.0	65.6	45.5	56.3	65.3	68.7
WIL Wald	38.3	56.3	61.1	61.4	39.9	55.4	62.3	65.5
GR	40.4	12.1	1.3	0.0	45.8	14.2	1.2	0.1
HBR	31.3	51.4	45.3	35.3	41.8	59.0	56.4	37.4
LS	43.4	61.0	65.7	66.9	10.8	16.5	28.3	34.6

Furthermore, suppose a user naively uses the inference results produced by an L_1 computing package for the corresponding WW-inferences. While such inference is appropriate for the L_1 estimator, it is not in general appropriate for the WW-estimator. For example, recall that $D_{WR}(\boldsymbol{\beta})$ is invariant to location and therefore, β_0 can not be directly estimated via the minimization. Hence, any displayed inference results for an intercept parameter would not be valid. This can be further illustrated by considering the standard errors for a simple linear regression model, say $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$, $i = 1, 2, \ldots, n$. For instance, the asymptotic standard error for the Wilcoxon estimate of β_1 is given by

$$\mathrm{SE}(\widehat{\beta}_1) \quad = \quad \frac{\tau}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}} \tag{20}$$

where $\tau = (\sqrt{12}\mathsf{E}\,[f(\varepsilon_1)])^{-1}$. However, in using an L_1 package for the computation of the denominator of (20), the X_is would be replaced by the pairwise differences of the X_is. Thus, an adjustment would need to be made. In general, the estimate of τ would also be different. For example, the L_1 estimate is essentially an estimator of the reciprocal of the mode of the error distribution. However, weighted differences of the residuals are used for the WW-estimate. Hence, the user would have to show that the estimate produced by the package is consistent for τ; which is unlikely for non-constant weighting schemes. Instead, we simply recommend that the user obtain the inference appropriate for the WW-estimator; that is, the inference produced by our R code.

Acknowledgment

We would like to thank the referees for many helpful suggestions that led to a much improved paper.

References

Chang WH, McKean JW, Naranjo JD, Sheather SJ (1999). "High-Breakdown Rank Regression." *Journal of the American Statistical Association*, **94**(445), 205–219.

Conover WJ (1999). *Practical Nonparametric Statistics*. Wiley Series in Probability and Statistics: Applied Probability and Statistics Section. John Wiley & Sons Inc., New York, third edition.

Daniel WW (1990). *Applied Nonparametric Statistics*. Boston: PWS-KENT Publishing Company, second edition.

Efron B, Tibshirani RJ (1993). *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, New York. ISBN 0-412-04231-2.

Fox AJ (1972). "Outliers in Time Series." *Journal of the Royal Statistical Society B*, **34**(3), 350–363.

Fuller WA (1996). *Introduction to Statistical Time Series*. John Wiley and Sons, New York, second edition.

Graybill FA (1976). *Theory and Application of the Linear Model*. Duxbury Press, North Scituate, Mass.

Handschin E, Kohlas J, Fiechter A, Schweppe F (1975). "Bad Data Analysis for Power System State Estimation." *IEEE Transactions on Power Apparatus and Systems*, **2**, 329–337.

Hawkins DM, Bradu D, Kass GV (1984). "Location of Several Outliers in Multiple-Regression Data using Elemental Sets." *Technometrics*, **26**(3), 197–208. ISSN 0040-1706.

Hettmansperger TP (1984). *Statistical Inference Based on Ranks*. John Wiley and Sons, New York.

Hettmansperger TP, McKean JW (1983). "A Geometric Interpretation of Inferences Based on Ranks in the Linear Model." *Journal of the American Statistical Association*, **78**(384), 885–893. ISSN 0162-1459.

Hettmansperger TP, McKean JW (1998). *Robust Nonparametric Statistical Methods*. Great Britain: Arnold.

Hollander M, Wolfe DA (1999). *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics: Texts and References Section. John Wiley & Sons Inc., New York, second edition. ISBN 0-471-19045-4. A Wiley-Interscience Publication.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**, 299–314.

Jaeckel LA (1972). "Estimating Regression Coefficients by Minimizing the Dispersion of the Residuals." *The Annals of Mathematical Statistics*, **43**(5), 1449–1458.

Koenker R, Bassett Jr G (1978). "Regression Quantiles." *Econometrica*, **46**(1), 33–50.

Kuehl RO (2000). *Design of Experiments: Statistical Principles of Research Design and Analysis*. Duxbury Press, Pacific Grove, CA, second edition.

Mallows CL (1975). "On Some Topics in Robustness." *Unpublished Memorandum*, Bell Telephone Laboratories, Murray Hill, NJ.

McKean JW, Naranjo JD, Sheather SJ (1996a). "Diagnostics to Detect Differences in Robust Fits of Linear Models." *Computational Statistics*, **11**, 223–243.

McKean JW, Naranjo JD, Sheather SJ (1996b). "An Efficient and High Breakdown Procedure for Model Criticism." *Communications in Statistics, Theory and Methods*, **25**(11), 2575–2595.

McKean JW, Sheather SJ (1991). "Small Sample Properties of Robust Analyses of Linear Models Based on R-estimates: A Survey." In "Directions in Robust Statistics and Diagnostics, Part II," volume 34 of *IMA Vol. Math. Appl.*, pp. 1–19. Springer, New York.

McKean JW, Sheather SJ, Hettmansperger TP (1994). "Robust and High-Breakdown Fits of Polynomial Models." *Technometrics*, **36**(4), 409–415. ISSN 0040-1706.

Naranjo JD, Hettmansperger TP (1994). "Bounded Influence Rank Regression." *Journal of the Royal Statistical Society B*, **56**(1), 209–220.

Naranjo JD, McKean JW, Sheather SJ, Hettmansperger TP (1994). "The use and Interpretation of Rank-Based Residuals." *Journal of Nonparametric Statistics*, **3**, 323–341.

R Development Core Team (2005). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3, URL http://www.R-project.org.

Rousseeuw PJ, Leroy AM (1987). *Robust Regression and Outlier Detection*. John Wiley and Sons, New York.

Rousseeuw PJ, Van Driessen K (1999). "A Fast Algorithm for the Minimum Covariance Determinant Estimator." *Technometrics*, **41**, 212–223.

Rousseeuw PJ, Van Driessen K (2002). "Computing LTS Regression for Large Data Sets." *Estadística*, **54**(162-163), 163–190 (2003). ISSN 0014-1135. Special issue on robust statistics.

Sievers GL (1983). "A Weighted Dispersion Function for Estimation in Linear Models." *Communications in Statistics, Theory and Methods*, **12**(10), 1161–1179.

Terpstra JT, McKean JW, Naranjo JD (2001). "Weighted Wilcoxon Estimates for Autoregression." *Australian & New Zealand Journal of Statistics*, **43**(4), 399–419. ISSN 1369-1473.

Theil H (1950). "A Rank-Invariant Method of Linear and Polynomial Regression Analysis (Parts 1-3)." *Ned. Akad. Wetensch. Proc. Ser. A*, **53**, 386–392, 521–525, 1397–1412.

Affiliation:

Jeff Terpstra
Department of Statistics
North Dakota State University
Fargo, ND 58105
E-mail: Jeff.Terpstra@ndsu.edu
URL: http://www.ndsu.nodak.edu/statistics/

Joseph McKean
Department of Statistics
Western Michigan University
Kalamazoo, MI 49008
E-mail: joe@stat.wmich.edu
URL: http://www.stat.wmich.edu/

Journal of Statistical Software
July 2005, Volume 14, Issue 7.
http://www.jstatsoft.org/

Submitted: 2004-06-04
Accepted: 2005-07-01