THE RESTRICTED LEAST SQUARES STEIN-RULE IN GRETL

Lee C. Adkins
Oklahoma State University

THE RESTRICTED LEAST SQUARES STEIN-RULE IN GRETL

LEE C. ADKINS

ABSTRACT. The paper documents a Gretl function package that is used for the Restricted Least Squares (RLS) Stein-rule estimator. Judge and Bock (1981, pp. 240-42) proposed a family of Stein-rule estimators that dominates the MLE of β in the CNLRM under weighted quadratic loss. The estimator is a linear combination of the unrestricted and restricted MLEs, where the degree of shrinkage is controlled using a conventional Wald test of the implied hypothesis restrictions. The Gretl function computes the positive-part version of the RLS Stein-rule that allows users to specify the desired linear restrictions on the model and to select a loss function under which to compute the RLS-Stein rule. In the absence of specific prior information about parameter values Lindley's version of the James-Stein rule is particularly attractive; accordingly, it is offered as a user specified option. The final version of the paper will also provide options for computing bootstrap standard errors [see Adkins (1990); Adkins and Hill (1990a)].

A simple Monte Carlo simulation is performed to explore the risk characteristics of the RLS-Stein rule vs. those of pretest, restricted mle, and unrestricted mle. All of the computations are preformed in gretl.

1. RLS Stein-rule Estimator

The classical normal linear regression model (CNLRM) is represented by

$$(1) \qquad\qquad y = X\beta + e \ \ e \sim N(0, \sigma^2 I_T)$$

where y is a $T \times 1$ vector of observable random variables, X is a nonstochastic $T \times K$ matrix of rank K, 3 is a $K \times 1$ vector of unknown parameters, and e is a $T \times 1$ vector of unobservable normally and independently distributed random variables having zero mean and finite variance σ^2. The ordinary least squares (OLS) and maximum likelihood estimator of β is $b = (X'X)^{-1}X'y \sim N(\beta, \sigma^2(X'X)^{-1})$, and the minimum variance unbiased estimator of σ^2 is $\hat{\sigma}^2 = (y - Xb)'(y - Xb)/(T - K)$, with $(T - K)\hat{\sigma}^2/\sigma^2 \sim \chi^2_{T-K}$ and independent of b.

Judge and Bock (1981) (pp. 240-42) proposed a family of Stein-rule estimators that dominates the MLE of β in the CNLRM under weighted quadratic loss. The loss associated with

---

using an estimator $\hat{\beta}$ to estimate a vector β with weight matrix W is:

$$(2) \qquad\qquad L(\hat{\beta}, \beta, W) = (\hat{\beta} - \beta)' W (\hat{\beta} - \beta)$$

For squared error loss $W = I_K$ and for mean square error of prediciton loss $W = X'X$.

The Judge and Bock estimator is a linear combination of the unrestricted and restricted MLEs and has the form

$$(3) \qquad\qquad \delta(b) = (1 - c/u)b + (c/u)b*$$

where $u = (Rb - r)'(RS^{-1}R')^{-1}(Rb - r)/J\hat{\sigma}^2 \sim F_{J,T-K,\lambda}$ is the conventional F-statistic used to test the hypothesis restrictions HO: $R\beta = r$, $S = X'X$; R is a known $J \times K$ nonstochastic matrix of rank J; r is a $J \times 1$ vector of known constants; $b* = b - S^{-1}R'(RS^{-1}R')^{-1}(Rb - r)$ is the restricted least squares estimator (RLS); $\lambda = (R\beta - r)'(RS^{-1}R')^{-1}(R\beta - r)/2\sigma^2$ is the noncentrality parameter; and $c = a(T - K)/J$. The estimator is minimax if the scalar a is chosen to lie within the interval $[0, a_{max}]$, where

$$(4) \qquad a_{max} = [2/(T - K + 2)]\{\lambda_L^{-1}tr[(RS^{-1}R')^{-1}RS^{-1}WS^{-1}R'] - 2\},$$

and λ_L is the largest characteristic root of $[(RS^{-1}R')^{-1}RS^{-1}WS^{-1}R']$. The value of the constant a that minimizes quadratic risk is the interval's midpoint.

In many circumstances, the usual Stein estimator is dominated by a simple modification called the positive-part rule. The positive-part rule associated with (3) is denoted

$$(5) \qquad\qquad \delta(b)^+ = \begin{cases} b*, & \text{if } c > u \\ \delta(b), & c \leq u. \end{cases}$$

Adkins and Hill (1990b) show that this general family of Stein-rules (3) mentioned in Judge and Bock, extended by Mittelhammer and Young (1981), and generalized by Mittelhammer (1984, 1985) is dominated by the positive-part rule, (5) which sets the Stein estimator equal to the RLS estimator whenever the shrinkage factor $(c/u) > 1$.

The positive-part rule takes a convex combination of the unrestricted and restricted MLEs; geometrically, this means that (5) lies between b and $b*$. The positive-part rule is intuitively appealing because it draws, or "shrinks," the unrestricted estimates toward, but not past, the restricted estimates. The usual Stein estimator (3) is less attractive because it changes the sign of the unrestricted MLEs whenever the value of the test statistic, u, is smaller than the constant, c. The positive-part Stein rule can be used whenever one has uncertain nonsample information involving three or more linear parameter restrictions.

2. GRETL FUNCTIONS

The main function used to estimate a linear model via the RLS Stein-rule is RLSStein. The function takes five arguments. The first is the **series** used as the dependent variable, the second is a **list** of regressors, third is a $J \times K$ **matrix** of linear restrictions to impose on the parameters of the model, fourth is a $J \times 1$ **matrix** of constants, the next is an integer (0, 1) that selects the loss function. The last two arguments are also integers, the first chooses whether to print the results and the last selects the number of bootstrap iterations to use for computation of standard errors.

```
1  function bundle RLSStein (series y "Dependent Variable",
2                            list EXOG "Regressors",
3                            matrix R "R for linear hypotheses RB=r",
4                            matrix r "r for linear hypotheses",
5                            int Loss[0:1:1] "Loss function" {"SEL", "MSEP"} ,
6                            int verb[0:1:1] "Verbosity" {"no print","print"} ,
7                            int B[100] "Bootstrap Replications")
8      # first thing, drop all obs with missing values anywhere
9      list EVERYTHING = y || EXOG
10     smpl EVERYTHING --no-missing
11
12     bundle rr = Stein_setup(y, EXOG, R, r, Loss, verb, B)
13     scalar err = aw(&rr)
14     scalar err = Stein_estimate(&rr)
15     scalar err = bootStein(&rr)
16     if verb == 1
17         Stein_printout(&rr)
18         endif
19     return rr
20 end function
```

The contents of the function are private, which means that they require no user input other than what comes from the RLSStein. The first private function, Stein_setup uses the inputs to initialize the bundle used to save and pass arguments to the remaining functions. It also estimates the unrestricted and restricted models.

The aw function is a wrapper for two other functions, Wmat and amax. Wmat interprets the string used to determine the loss function and produces a corresponding weight matrix, W. The amax function determines the maximum shrinkage allowed a_{max}. The Stein_estimate function computes the RLS Stein-rule and returns more arguments to the bundle and Stein_printout prints the unrestricted, restricted and Stein-rule estimates. At this point, zeros are reported for Stein-rule standard errors. The last function is bootStein and it

computes bootstrap covariance and standard errors using the procedure described in section 3. The conditional statement uses the stored verbosity parameter to control the amount of printout. This is especially useful for simulations and to limit the amount of output produced during the bootstrap iterations.

Finally, the function returns a bundle. A bundle is a container for various objects namely, scalars, series, matrices, strings and bundles. Notice that a bundle cannot contain a list. A bundle takes the form of a hash table or associative array: each item placed in the bundle is associated with a key string that can used to retrieve it subsequently. The resulting bundle includes various matrices, scalars, and strings associated with estimating the model. The bundle includes the following:

Bundle Key	Description
depvar	series: dependent variable
depvarname	string: name of the dependent variable
mEXOGnames	string: names of the exogenous variables
Loss	string: W loss function name
mEXOG	matrix: exogenous variables
mW	matrix: Weight matrix
unrest	matrix: OLS estimates
unrest_se	matrix: OLS standard errors
Rmat	matrix: R in $R\beta - r$
r	matrix: r in $R\beta - r$
J	scalar: number of restrictions
res	matrix: RLS estimates
res_se	matrix: RLS std errors
SRCoeff	matrix: $\delta(b)^+$ Stein-rule estimates
max_eigen	scalar: λ_L the maximum eigenvalue
complex	scalar: 1 if any char roots are complex; 0 otherwise
amax	scalar: a_{max}
collinear	scalar: 1 if $a_{max} < 0$; 0 otherwise
u	scalar: F-statistic for testing $R\beta - r$
c	scalar: c shrinkage constant
shrinkage	scalar: c/u the amount of shrinkage
SR_se	matrix: Bootstrap standard errors for Stein-rule
bootCov	matrix: Bootstrap covariance estimate
B	scalar: number of bootstrap samples
Sresidual	matrix: Stein-rule residuals
N	scalar: N Sample size
mk	scalar: K number of regressors
t1	scalar: Beginning obs. number
t2	scalar: ending obs. number
verbosity	scalar: $0, 1$; 0 turns off printing results

The dialog box produced by the GUI is shown in Figure 1. Once can choose SEL or MSEP for the loss function, and to print or not to print output in verbosity. Selecting 0 in the bootstrap field suppresses computation of the bootstrap standard errors. This option is useful in using the function for simulations. Otherwise, the user may choose the number of boostrap replications to use for computation of standard errors. To save the results in a bundle, then one may do so in the last field.
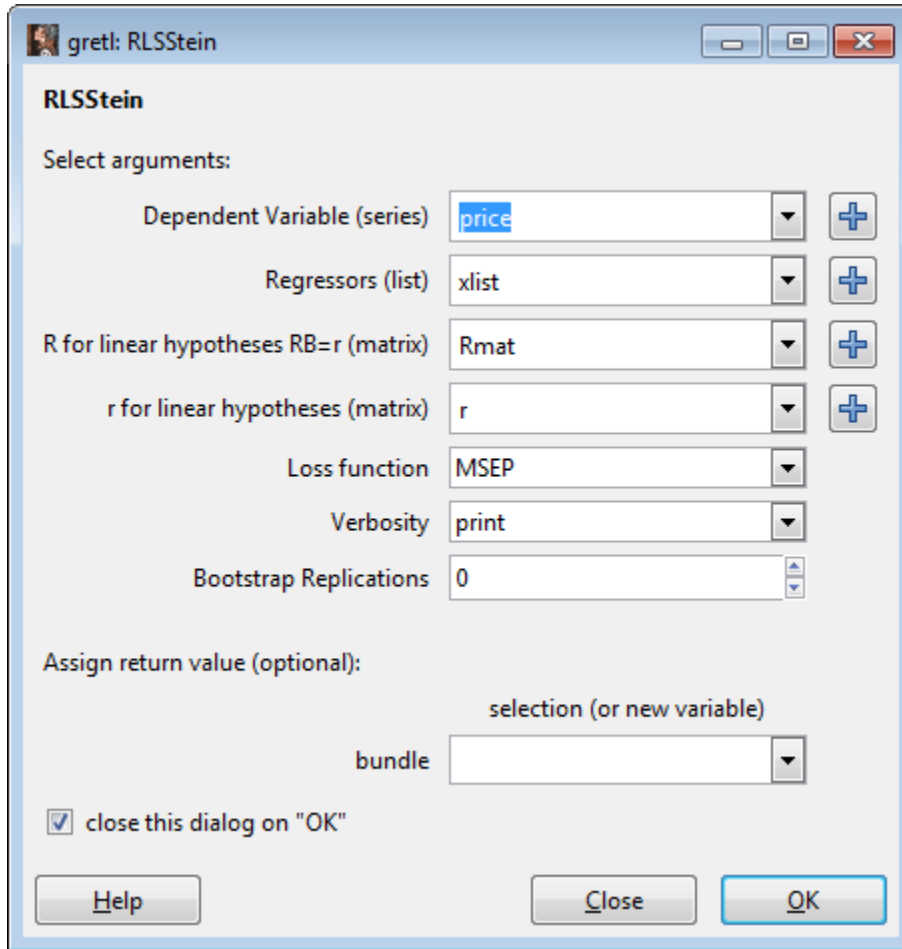
FIGURE 1. The RLS Stein-rule dialog box obtained using the GUI

.

The R and r fields are particularly useful for building the necessary restrictions. Clicking on the plus sign produces the build matrix dialog and users can choose to build a restriction matrix manually by specifying the desired number of rows (J) and columns (K) and then typing in the desired coefficients on the parameters. The matrix and build matrix dialogs are shown in Figure 2

3. BOOTSTRAP

The bootstrap can be used in a variety of ways [see Efron (1982)]. The most common form of the bootstrap is nonparametric and uses the least squares estimates b to obtain the set of residuals $e = y - Xb$; these serve as the estimates of the true disturbances of the model and are thought to capture its underlying structure.
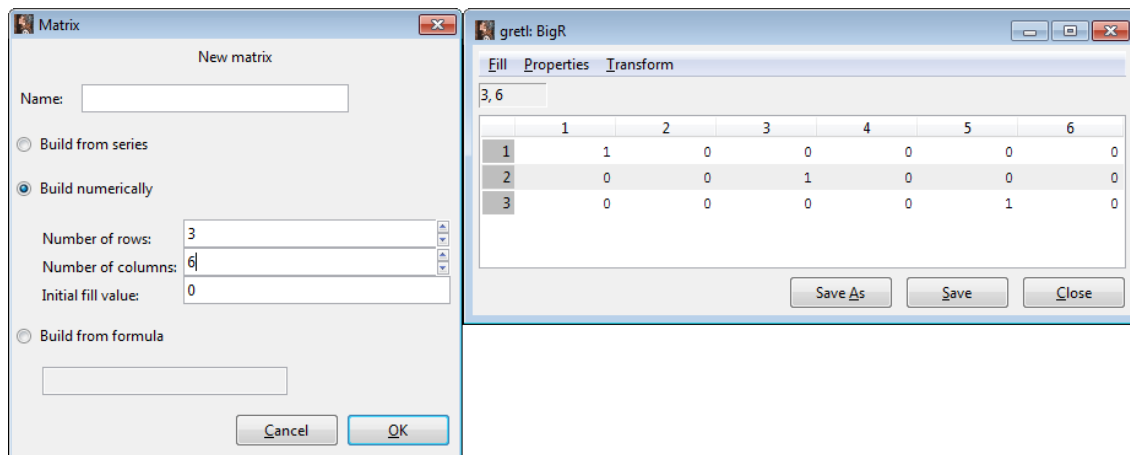
FIGURE 2. The Matrix and Build Matrix dialogs

.

One method of generating semiparametric bootstrap standard errors is based on using the empirical distribution of the least squares residuals. The least squares residuals are rescaled using $\hat{e}_t^* = (T/(T-K))^{1/2}\hat{e}_t$, $t = l, \ldots, T$, [see Wu (1986) for discussion] and a bootstrap sample of size T is drawn randomly and with replacement from $\hat{e}$, denoted e^*. Then the bootstrap sample $y^* = Xb + e^*$ is obtained and the bootstrap estimate δ^* is computed. A large number, N, of size T random samples are drawn from the empirical distribution $\hat{e}^*$ and the sequences $\{y^*\}_1^N$ and $\{\delta^*\}_1^N$ are computed. The sample covariance of this sequence could then be used to estimate the covariance of the RLS Stein-rule.

Adkins (1992) shows that this method tends to overstate the size of standard errors, at least for the simpler James-Stein estimator, and suggests ways to reduce this bias. The approach adopted here is to resample randomly from the RLS Stein-rule residuals $e_\delta = y - X\delta$ and generate bootstrap samples using $y^* = X\delta + e_\delta^*$ where e_δ^* represents a random resample from the RLS Stein-rule residuals e_δ. The sequence $\{\delta^*\}_1^N$ are computed and the sample covariance of this sequence is used to estimate the covariance of the RLS Stein-rule. This approach is similar in spirit to that taken by Brownstone (1990). This modification can be justified on theoretical grounds since δ is consistent for β.

It is debateable whether the Stein-rule residuals should be rescaled. In the current version of the package, they are. The example given below contains only 14 observations and rescaling residuals increases the estimated standard errors by a substantial amount. The effect as T increases will diminish.

7

4. Example

The example uses data on single family homes in the University City community of San Diego, in 1990. It is available in the base distribution of gretl as `data4-1.gdt`.

The basic model to be estimated is:

$$(6) \qquad \text{price} = \beta_1 + \beta_2 \text{sqft} + \beta_3 \text{sqft}^2 + \beta_4 \text{bedrms} + \beta_5 \text{baths} + e$$

The following restrictions are considered: $\beta_2 = 360$; $\beta_3 = 0$; $\beta_4 = -2$; and, $\beta_5 = -51$.

```
1  open data4-1
2  series sqft = sqft/1000
3  square sqft bedrms
4  list xlist = const sqft sq_sqft bedrms baths
5  matrix Rmat = zeros(4,1)~I(4)
6  matrix r = { 350 ; -50 ; 0 ; 0}
7
8  bund = RLSStein(price, xlist, Rmat, r, 1, 1, 100)
```

Estimating the model under squared error loss produces the message:

```
1  Squared Error Loss
2
3  The data are highly collinear and no shrinkage
4  will occur under the current loss function.
```

Hence, the RLS Stein estimates are the same as those of OLS, except for the standard errors, which are based on the bootstrap. The RLS Stein-rule is estimated under mean square error of prediction loss.

The results from the example script if found below. Notice in line 12 that the chosen weight matrix for the loss function is reported. Also, there is a note give in lines 14-15 that no shrinkage is occurring under this loss function due to high collinearity among the regressors. Hence, the Stein estimates are the same as those for OLS. Finally, in line 50 the computed values of the shrinkage constants a and c are printed, the amount of shrinkage toward RLS, and the value of the F-statistic used to control the amount of shrinkage are given.

```
1  gretl version 1.9.10cvs
2  Current session: 2012-11-17 12:13
```

```
 3  ? set echo off
 4
 5  Read datafile C:\Program Files (x86)\gretl\data\data4-1.gdt
 6  periodicity: 1, maxobs: 14
 7  observations range: 1-14
 8
 9  Listing 5 variables:
10    0) const      1) price      2) sqft      3) bedrms    4) baths
11
12  Squared Error Loss
13
14  The data are highly collinear and no shrinkage
15  will occur under the current loss function.
16
17  Stein-Rule Estimation of a linear regression
18  using observations 1-14
19  Dependent Variable y
20  Unrestricted OLS
21
22              coefficient    std. error      z       p-value
23    --------------------------------------------------------
24    const      -14.8037       138.026     -0.1073    0.9146
25    sqft       367.990        163.896      2.245     0.0248    **
26    sq_sqft    -51.1936        38.6554    -1.324     0.1854
27    bedrms     -43.7401        30.9703    -1.412     0.1579
28    baths       -3.71536       42.1948    -0.08805   0.9298
29
30  Restricted LS
31
32              coefficient    std. error      z       p-value
33    --------------------------------------------------------
34    const      -153.252        10.2323     -14.98     1.03e-050 ***
35    sqft        350.000         0.000000    NA        NA
36    sq_sqft     -50.0000        0.000000    NA        NA
37    bedrms        0.000000      0.000000    NA        NA
38    baths         0.000000      0.000000    NA        NA
39
40  Stein-Rule estimates
41              Stein          SE
42    const   -14.8037   131.8333
43     sqft   367.9898   139.9846
44  sq_sqft   -51.1936    33.4086
45   bedrms   -43.7401    26.2731
46    baths    -3.7154    31.0739
47
48  a=0.0000, c=0.0000, shrinkage is 0.0000 and F=0.8177
```

Under MSEP loss the Stein estimates are:

```
1  Stein-Rule estimates
2               Stein          SE
3    const    -84.0725    81.3650
4     sqft    358.9891    92.5592
5  sq_sqft    -50.5964    21.0457
6   bedrms    -21.8559    21.1038
7    baths     -1.8565    25.3871
8
9  a=0.3636, c=0.4091, shrinkage is 0.5003 and F=0.8177
```

In this case, the RLS estimates are approximately midway between the OLS and RLS estimates.

5. SIMULATION

In this section I will report the results from a small simulation of the estimators considered above. Their risk characteristics under mean square error of prediction loss will be examined. The results confirm that the RLS Stein-rule dominates least squares and performs better than pretest and restricted estimators over large areas of the parameter space.

The experimental design is adopted with minor modification from Adkins (1992).

For the Monte Carlo each of 8 regressors in the X matrix has been standardized to have zero mean, variance 1, and to be mutually orthogonal. This design is referred to as the orthonormal regression model (i.e., $X'X = I$) and corresponds to the model of the mean of a multivariate population that is widely studied in statistics. The orthonormal model is computationally convenient and the results generalize to nonorthogonal models for which it is a canonical form. The weight matrix W, which appears in the scalar 'a,' is chosen to be $W = X^T X$). This choice corresponds to mean square error of prediction loss; 'a' is chosen to be the midpoint of the interval $(0, a_{max})$.

A total of 500 pseudo-random samples of size T=30 were drawn from the $N(0, 1)$ density for each of the parameter vectors. The design points $\beta = \ell j$, are generated using

$$(7) \qquad\qquad \ell = R^2 T \sigma^2 / ((1 - R^2) j^T j)^{1/2}$$

where j is $8 \times l$ vector of ones, and population goodness-of-fit $R^2 = [\, 0.001, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]$. Thus, the risk of the Stein-rule is studied over various degrees of specification error inherent in the hypothesis restrictions. Larger values of R^2 could be considered, but
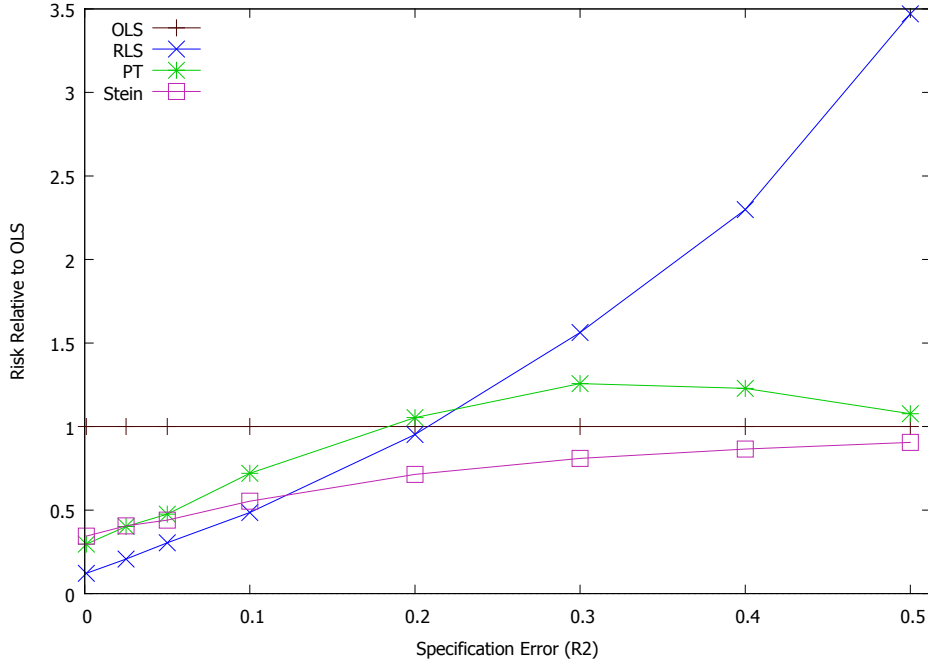
FIGURE 3. Relative Risk Profiles of OLS, RLS, Pretest and Stein-rule estimators

the largest here is set at $R^2 = 0.5$ where the Pretest, OLS and Stein-rule estimators' risks converge.

The RLS-Stein estimator is studied for several points in the parameter space that lie along a ray extending from the origin. As R^2 increases, the degree of noncentrality increases as well.

A pretest of the restrictions is tested at the 10% level and computed:

$$(8) \qquad\qquad b_{PT} = I(p \leq 0.1)b + I(p > 0.1)b^*$$

where p is the p-value associated with the Wald test of the hypothesis restrictions and $I(p)$ is the indicator function, taking the value of 1 if expression in parentheses is true and 0 otherwise. The restrictions condisered are

$$(9) \qquad\qquad \beta_2 = \beta_3 = \ldots = \beta_K = 0$$

that is, all of the parameters except the first is equal zero. This amounts to $J = K - 1$ restrictions; In the restricted model this leaves a single parameter to estimate.

The risks of RLS, Pretest (PT) and the RLS Stein-rule (Stein) are plotted in Figure 3. All risks are measured relative to that of OLS (which theoretically, has a constant risk equal to K in the orthonormal model). RLS risk increases without limit as error in the hypotheses

11

grows. The pretest estimator has risk lower than OLS when the hypotheses are nearly true, but exceeds that of OLS (though only by a small amount, roughly 26% at its maximum where $R^2 = 0.3$) as the probability of rejecting the restrictions increases. The RLS-Stien rule dominates OLS and nearly dominates the pretest.

6. SIMULATION CODE

```
 1  nulldata 30
 2  include loss.inp
 3  include Steinbundle_BS.inp
 4  scalar n=30          # sample size
 5  scalar K=8           # number of regressors
 6  scalar sig2=1        # variance of y
 7  scalar alpha = .1    # level of pretest
 8  scalar nmc=500       # number of simulations
 9  # Generating regressors
10  matrix x = mrandgen(n,0,1,n,K)
11  # Rescale regressors to unit length
12  matrix dd = 1/sqrt(diag(x'x))
13  matrix xm=x.*(ones(n,1).*dd')
14  matrix U V
15  matrix d=svd(x, &U, &V)
16  d=ones(1,K)
17  # set the singular values to 1 and reform x, xm now orthonormal
18  matrix xm = (U .* d)
19  # create a list to use in the RLSStein function
20  list xlist = null
21  loop i = 1..K
22      series x_$i =  xm[,i]
23      xlist += x_$i
24  endloop
25  # Generation of design points
26  matrix R2={ .001, .025, .05, .1, .2, .3, .4, .5}
27  scalar D = nelem(R2)
28  # Generate quantities used repeatedly in computations
29  matrix xmx=xm'xm
30  matrix j=ones(K,1)
31  # Initialize risk matrix
32  matrix RiskMat=zeros(D,4)
33  # Create Restrictions
34  matrix BigR=zeros(K-1,1)~I(K-1)
35  matrix Lr=zeros(K-1,1)
36  # Main Monte Carlo Loop
37  loop i=1..D
38      # Initialize Loss Matrix
```

```
39      matrix LossMat=zeros(nmc,4)
40      # Set beta
41      matrix r2 = R2[i]
42      matrix L=sqrt(r2*n*sig2/((1-r2)*j'j))
43      matrix beta = L .* j
44      # Generate samples using beta
45      loop jj = 1..nmc --quiet
46          y=xm*beta+normal(0,sig2)
47          # Convert the matrix y to a series, which RLSStein requires
48          series y_s = y
49          # Estimate RLSStein
50          Stein=RLSStein(y_s, xlist, BigR, Lr, 1, 0, 0 )
51          # retrieve OLS, Stein, u, and RLS from the bundle
52          b=Stein["unrest"]
53          bStein=Stein["SRCoeff"]
54          bRes = Stein["res"]
55          u=Stein["u"]
56          # Use the F-stat to compute PT estimator
57          scalar p=pvalue(F,Stein["J"],n-K,u)
58          matrix bpt = bRes
59          if p < alpha
60              bpt = b
61          endif
62          # Calculate Loss for each estimator
63          LossMat[jj,1]=loss(beta,b,xmx)
64          LossMat[jj,2]=loss(beta,bRes,xmx)
65          LossMat[jj,3]=loss(beta,bpt,xmx)
66          LossMat[jj,4]=loss(beta,bStein,xmx)
67
68      end loop
69      # Average the Losses to obtain Risk
70      matrix risk = meanc(LossMat)
71      # save the risk of OLS to normalize the others
72      matrix rols = risk[1]
73      matrix RiskMat[i,]=risk./rols
74  end loop
75  # Form the matricies for printing and then plot results
76  matrix RR=RiskMat ~ R2'
77  colnames(RR, "OLS RLS PT Stein R2" )
78  RR
79  gnuplot --matrix=RR --output=display --with-lp \
80    { set xlabel 'Specification Error (R2)'; \
81      set ylabel 'Risk Relative to OLS'; }
```

References

Adkins, Lee C. (1990), 'Small sample performance of jackknife confidence intervals for the James-Stein estimator', *Communications in Statistics* .

Adkins, Lee C. (1992), 'Finite sample moments of a bootstrap estimator of the James-Stein rule', *Econometric Reviews* **11**, 173–193.

Adkins, Lee C. and R. Carter Hill (1990*a*), 'An improved confidence ellipsoid for the linear regression model', *Journal of Statistical Computation and Simulation* **36**.

Adkins, Lee C. and R. Carter Hill (1990*b*), 'The RLS positive-part Stein estimator', *American Journal of Agricultural Economics* **72**, 727–730.

Brownstone, David (1990), 'Bootstrapping improved estimators for linear regression models', *Journal of Econometrics* **44**, 171–188.

Efron, Bradley (1982), *Jackknife. Bootstrap. and Other Resamp1ing Plans*, Society for Industrial and Applied Mathematics, Philadelphia.

Judge, G. G. and M. E. Bock (1981), *The Statistical Implications of Pre-Test and Stein-Rule Estimators in Econometrics*, North-Holland Publishing Co., Amsterdam.

Mittelhammer, R. C. (1984), 'Restricted least squares, pre-test, OLS, and Stein rule estimators: Risk comparisons under model misspecification', *Journal of Econometrics* **25**, 151–64.

Mittelhammer, R. C. (1985), 'Quadratic risk domination of restricted least squares estimators via Stein-ruled auxiliary constraints', *Journal of Econometrics* **29**, 289–303.

Mittelhammer, R. C. and D. L. Young (1981), 'Mitigating the effects of multicollinearity using exact and stochastic restrictions: Reply', *American Journal of Agricultural Economics* **63**, 301–4.

Wu, C. J. (1986), 'Jackknife, bootstrap, and other resampling methods in regression analysis', *Annals of Statistics* **14**, 1261–1295.

Lee C. Adkins, Professor of Economics, College of Business Administration, Oklahoma State University, Stillwater OK 74078

E-mail address: lee.adkins@okstate.edu