# Robust Statistics in Stata

Ben Jann

University of Bern, ben.jann@soz.unibe.ch

2017 London Stata Users Group meeting
London, September 7–8, 2017

# Contents


1. The robstat command
2. The robreg command
3. The robmv command
4. The roblogit command
5. Outlook

# The robstat command

- Computes various (robust) statistics of location, scale, skewness and kurtosis (classical, M, quantile-based, pairwise-based).
- Provides various generalized Jarque-Bera tests for normality as suggested by Brys et al. (2008).
- Variance estimation based on influence functions; full support for complex survey data.
- Simultaneous estimation of multiple statistics for multiple outcomes and multiple subpopulations (including full variance matrix).
- Using fast algorithms for the pairwise-based estimators (based on Johnson and Mizoguchi 1978; also see Croux and Rousseeuw 1992, Brys et al. 2004).

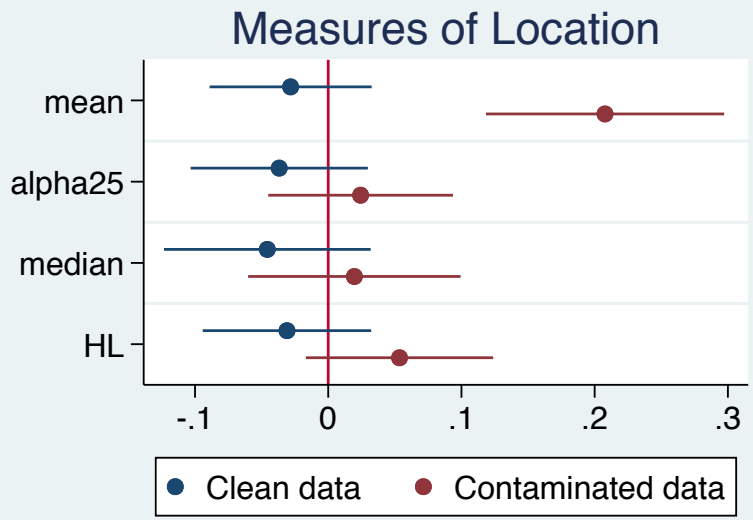
# Examples for robstat

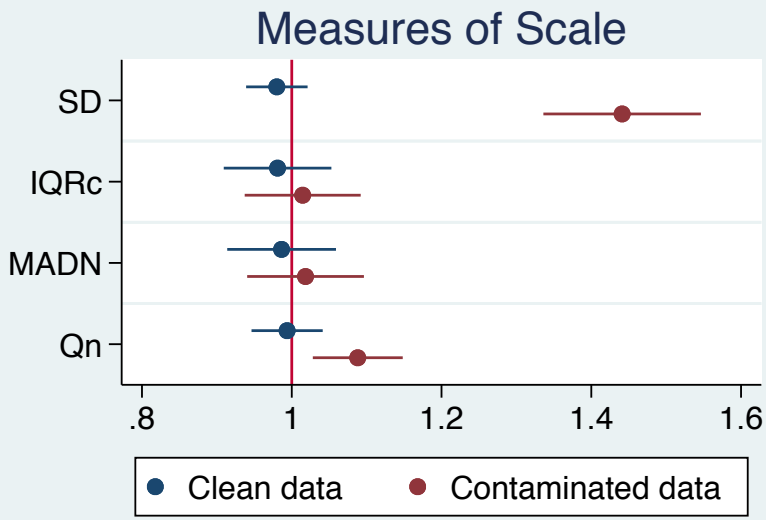
```
. // Generate some data
. clear all
. set seed 64334
. set obs 1000
number of observations (_N) was 0, now 1,000
. // Normally distributed variable (mean 0, standard deviation 1)
. generate z  = rnormal(0, 1)
. // Contaminated data (5% point mass at value 5)
. generate zc = z
. replace  zc = 5 if uniform()<.05
(48 real changes made)
```

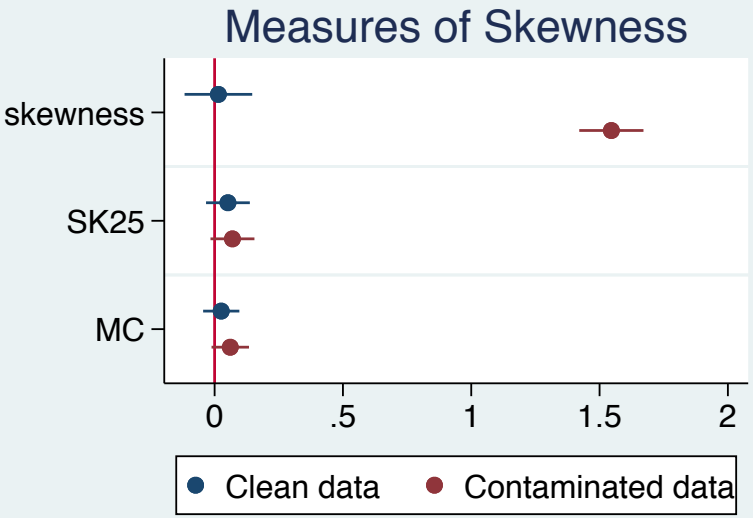
```
. robstat z zc, statistics(/*Location:*/ mean alpha25 median HL ///
>                          /*Scale:   */ SD IQRc MADN Qn ///
>                          /*Skewness:*/ skewness SK25 MC)
```

Robust Statistics  Number of obs = 1,000

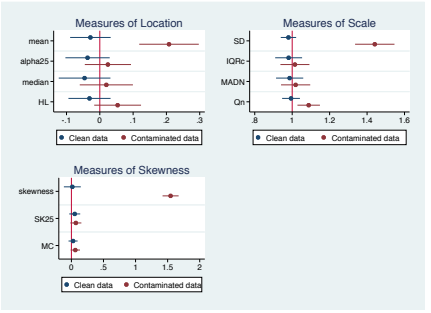
| | Coef. | Std. Err. | [95% Conf. | Interval] |
|---|---|---|---|---|
| z | | | | |
| mean | -.0282395 | .0309938 | -.0890599 | .0325809 |
| alpha25 | -.0367917 | .0338939 | -.1033031 | .0297196 |
| median | -.0457205 | .0395328 | -.1232973 | .0318563 |
| HL | -.0309803 | .0322082 | -.0941838 | .0322232 |
| SD | .9801095 | .0208968 | .9391027 | 1.021116 |
| IQRc | .9810724 | .0366146 | .9092221 | 1.052923 |
| MADN | .9864664 | .0370098 | .9138406 | 1.059092 |
| Qn | .9938275 | .0242308 | .9462783 | 1.041377 |
| skewness | .0146799 | .067181 | -.1171522 | .1465121 |
| SK25 | .0518805 | .0434621 | -.033407 | .1371679 |
| MC | .0257808 | .0359978 | -.0448592 | .0964208 |
| zc | | | | |
| mean | .2078149 | .0455753 | .1183806 | .2972493 |
| alpha25 | .0243169 | .0353438 | -.0450396 | .0936735 |
| median | .0196103 | .0406538 | -.0601664 | .099387 |
| HL | .0535221 | .0358125 | -.0167543 | .1237984 |
| SD | 1.441218 | .0535933 | 1.33605 | 1.546387 |
| IQRc | 1.014596 | .0394691 | .9371445 | 1.092048 |
| MADN | 1.018404 | .0397156 | .9404688 | 1.09634 |
| Qn | 1.088171 | .0305887 | 1.028146 | 1.148197 |
| skewness | 1.546024 | .0637939 | 1.420838 | 1.671209 |
| SK25 | .0694487 | .0436647 | -.0162363 | .1551338 |
| MC | .0609388 | .0371472 | -.0119568 | .1338343 |


Measures of Location
mean
alpha25
median
HL
-.1
0
.1
.2
.3
Clean data
Contaminated data


Measures of Scale
SD
IQRc
MADN
Qn
.8
1
1.2
1.4
1.6
Clean data
Contaminated data


Measures of Skewness
skewness
SK25
MC
0
.5
1
1.5
2
Clean data
Contaminated data

# Robust Statistics in Stata
## The robstat command

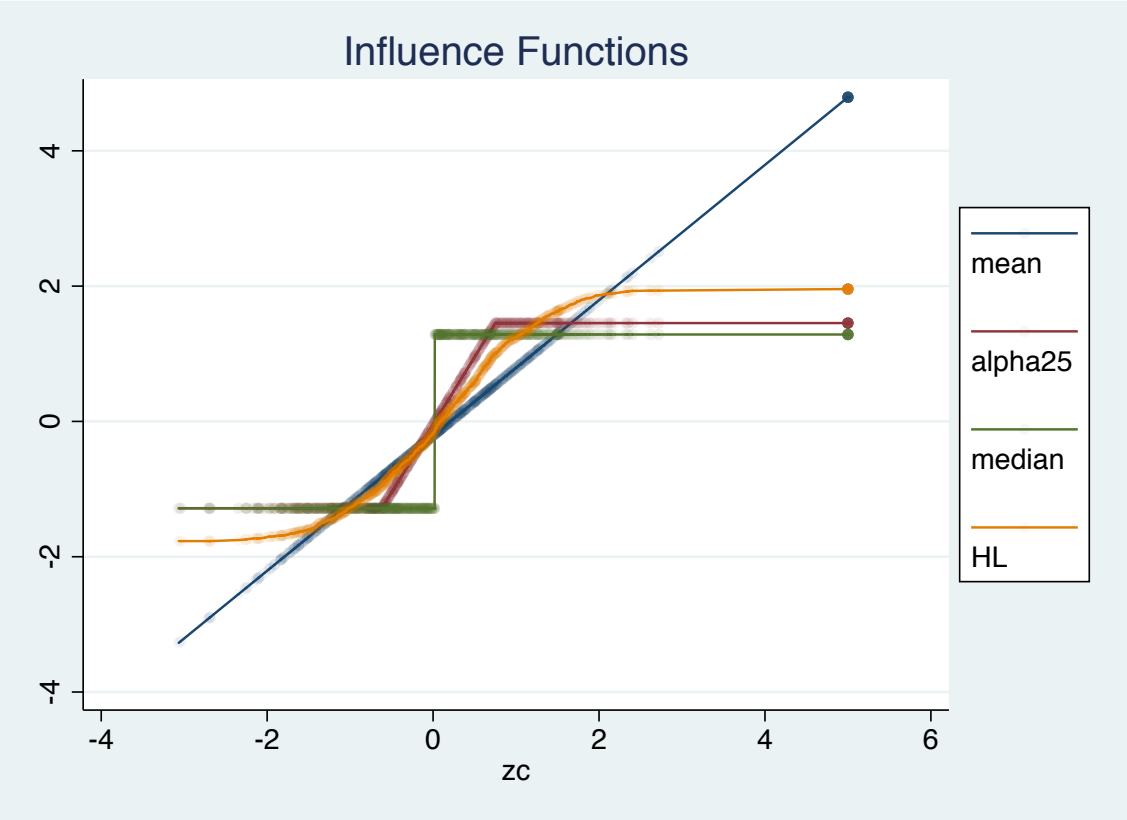



```
. coefplot (., drop(zc:)) (., drop(z:)), keep(mean alpha25 median HL) ///
>     xline(0) plotlabels("Clean data" "Contaminated data") ///
>     title("Measures of Location") nodraw name(loc, replace)

. coefplot (., drop(zc:)) (., drop(z:)), keep(SD IQRc MADN Qn) ///
>     xline(1) plotlabels("Clean data" "Contaminated data") ///
>     title("Measures of Scale") nodraw name(sc, replace)

. coefplot (., drop(zc:)) (., drop(z:)), keep(skewness SK25 MC) ///
>     xline(0) plotlabels("Clean data" "Contaminated data") ///
>     title("Measures of Skewness") nodraw name(sk, replace)

. graph combine loc sc sk
```

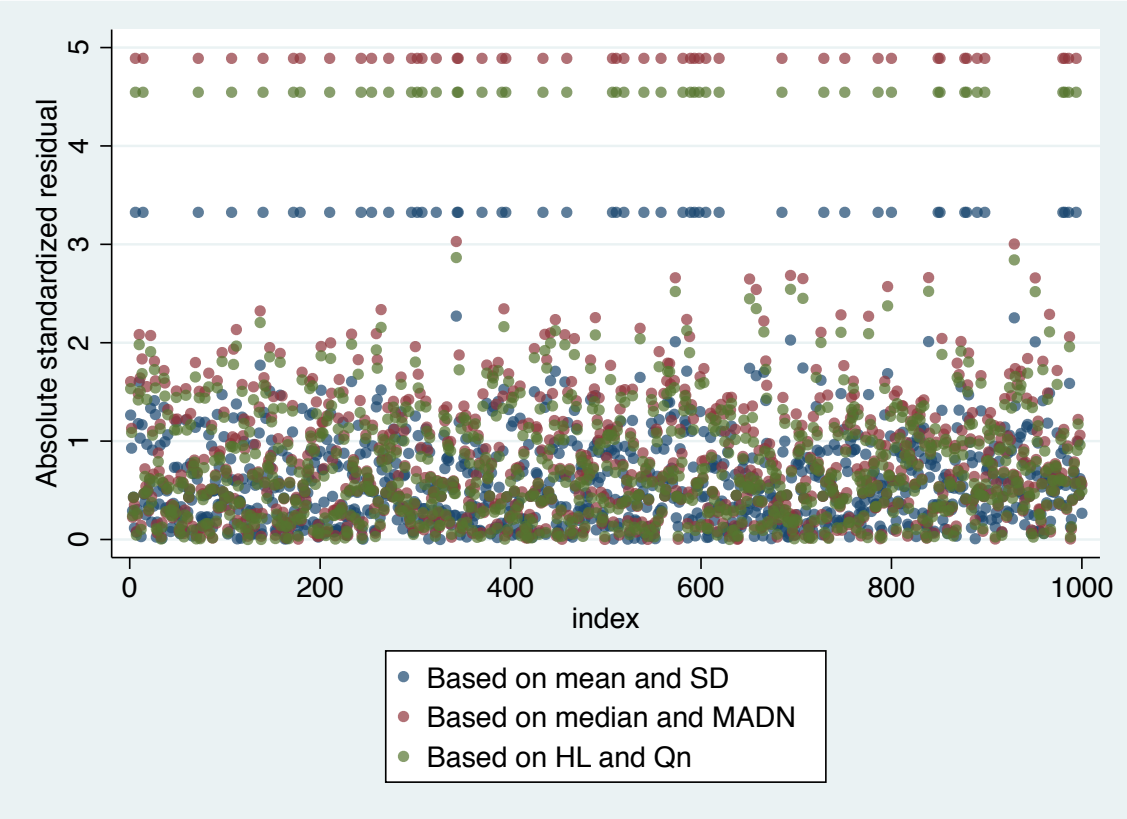
```
. // May use -generate()- to store the estimated influence functions
. robstat zc, statistics(mean alpha25 median HL) generate
```

(*output omitted*)

```
. two connect _IF* zc, sort ms(o ..) mc(%5 ..) mlc(%0 ..) ///
>     legend(order(1 "mean" 2 "alpha25" 3 "median" 4 "HL") ///
>     cols(1) stack pos(3) keygap(0) rowgap(5)) ///
>     ti("Influence Functions")
```



```
. // Identifying outliers
. robstat zc, statistics(mean SD median MADN HL QN)
  (output omitted)
. generate o_classic  = abs((zc-_b[mean])/_b[SD])
. generate o_quantile = abs((zc-_b[median])/_b[MADN])
. generate o_pairwise = abs((zc-_b[HL])/_b[Qn])
. generate index = _n
. scatter o_classic o_quantile o_pairwise index, ///
>     ms(o ..) mc(%70 ..) mlc(%0 ..) yti("Absolute standardized residual") ///
>     legend(order(1 "Based on mean and SD" 2 "Based on median and MADN" 3 "Based on HL and Qn") cols(1))
```



```
. // Normality tests
. robstat z zc, jbtest
```

Robust Statistics  Number of obs = 1,000

| | Coef. | Std. Err. | [95% Conf. | Interval] |
|---|---|---|---|---|
| z | | | | |
| skewness | .0146799 | .067181 | -.1171522 | .1465121 |
| kurtosis | 2.820148 | .1134951 | 2.597432 | 3.042865 |
| SK25 | .0518805 | .0434621 | -.033407 | .1371679 |
| QW25 | 1.263147 | .0583604 | 1.148624 | 1.37767 |
| MC | .0257808 | .0359978 | -.0448592 | .0964208 |
| LMC | .2686648 | .0500956 | .1703602 | .3669694 |
| RMC | .1773274 | .0537777 | .0717972 | .2828577 |
| zc | | | | |
| skewness | 1.546024 | .0637939 | 1.420838 | 1.671209 |
| kurtosis | 6.536769 | .3352475 | 5.8789 | 7.194639 |
| SK25 | .0694487 | .0436647 | -.0162363 | .1551338 |
| QW25 | 1.37593 | .0644609 | 1.249435 | 1.502424 |
| MC | .0609388 | .0371472 | -.0119568 | .1338343 |
| LMC | .317669 | .0503196 | .2189247 | .4164133 |
| RMC | .3022334 | .0568872 | .1906013 | .4138655 |

Normality Tests

| | chi2 | df | Prob>chi2 |
|---|---|---|---|
| z | | | |
| JB | 1.38 | 2 | 0.5007 |
| MOORS | 1.81 | 2 | 0.4040 |
| MC-LR | 2.21 | 3 | 0.5304 |
| zc | | | |
| JB | 919.56 | 2 | 0.0000 |
| MOORS | 9.40 | 2 | 0.0091 |
| MC-LR | 12.46 | 3 | 0.0060 |

```
. // Survey estimation
. webuse nhanes2f, clear
. svyset psuid [pweight=finalwgt], strata(stratid)

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: stratid
         SU 1: psuid
        FPC 1: <zero>
. robstat copper, statistics(mean median huber95 HL) svy

Survey: Robust Statistics

Number of strata =        31        Number of obs    =         9,118
Number of PSUs   =        62        Population size  =   103,505,700
                                     Design df        =            31
```

| copper | Coef. | Linearized Std. Err. | [95% Conf. | Interval] |
|---|---|---|---|---|
| mean | 124.7232 | .6657517 | 123.3654 | 126.081 |
| median | 118 | .5894837 | 116.7977 | 119.2023 |
| Huber95 | 119.897 | .5378589 | 118.8001 | 120.994 |
| HL | 120 | .5502105 | 118.8778 | 121.1222 |

```
. robstat copper, statistics(mean median huber95 HL) svy over(sex) total
Survey: Robust Statistics
Number of strata =      31      Number of obs    =        9,118
Number of PSUs   =      62      Population size  =  103,505,700
                               Design df        =           31
            1: sex = Male
            2: sex = Female
```

| copper | Linearized Coef. | Std. Err. | [95% Conf. | Interval] |
|---|---|---|---|---|
| 1 | | | | |
| mean | 111.1328 | .2743469 | 110.5733 | 111.6924 |
| median | 109 | .2282107 | 108.5346 | 109.4654 |
| Huber95 | 109.9958 | .2613545 | 109.4627 | 110.5288 |
| HL | 110 | .2574198 | 109.475 | 110.525 |
| 2 | | | | |
| mean | 137.3958 | .5440111 | 136.2863 | 138.5053 |
| median | 129 | .4460584 | 128.0903 | 129.9097 |
| Huber95 | 131.5431 | .4857928 | 130.5523 | 132.5339 |
| HL | 131.5 | .4789198 | 130.5232 | 132.4768 |
| total | | | | |
| mean | 124.7232 | .6657517 | 123.3654 | 126.081 |
| median | 118 | .5894837 | 116.7977 | 119.2023 |
| Huber95 | 119.897 | .5378589 | 118.8001 | 120.994 |
| HL | 120 | .5502105 | 118.8778 | 121.1222 |

# The robreg command

- Supports various robust regression estimators (M, S, MM, and some other high breakdown estimators).
- Hausman-type tests (S against least-squares, MM against S).
- Robust standard errors (Croux et al. 2003).
- S-estimator: Fast subsampling algorithm (Salibian-Barrera and Yohai 2006) with speed improvements for categorical predictors (Koller 2012).

# Examples for robstat

```
. // diabetes data from https://www.cdc.gov/diabetes/data/countydata/countydataindicators.html
. set seed 312003
. use diabetes, clear
. drop county
. qui drop if percphys>=.
. qui drop if percob>=.
. qui sample 1000, count
. describe
Contains data from diabetes.dta
  obs:         1,000
 vars:             3                          7 Sep 2017 17:55
 size:        12,000
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
perdiabet       float   %8.0g                 Diabetes prevalence
percob          float   %8.0g                 Obesity prevalence
percphys        float   %8.0g                 Physical inactivity prevalence
-------------------------------------------------------------------------------
Sorted by:
     Note: Dataset has changed since last saved.
```

```
. // classic regression using clean data
. reg perdiabet percphys percob
```

| Source | SS | df | MS | | |
|---|---|---|---|---|---|
| Model | 1675.80732 | 2 | 837.903662 | | |
| Residual | 847.376638 | 997 | .849926417 | | |
| Total | 2523.18396 | 999 | 2.52570967 | | |

| | | | | | |
|---|---|---|---|---|---|
| Number of obs | = | 1,000 | | | |
| F(2, 997) | = | 985.85 | | | |
| Prob > F | = | 0.0000 | | | |
| R-squared | = | 0.6642 | | | |
| Adj R-squared | = | 0.6635 | | | |
| Root MSE | = | .92191 | | | |

| perdiabet | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1332684 | .0076287 | 17.47 | 0.000 | .1182982 | .1482385 |
| percob | .2196914 | .0120911 | 18.17 | 0.000 | .1959645 | .2434182 |
| _cons | -.7546425 | .2251135 | -3.35 | 0.001 | -1.196393 | -.3128918 |

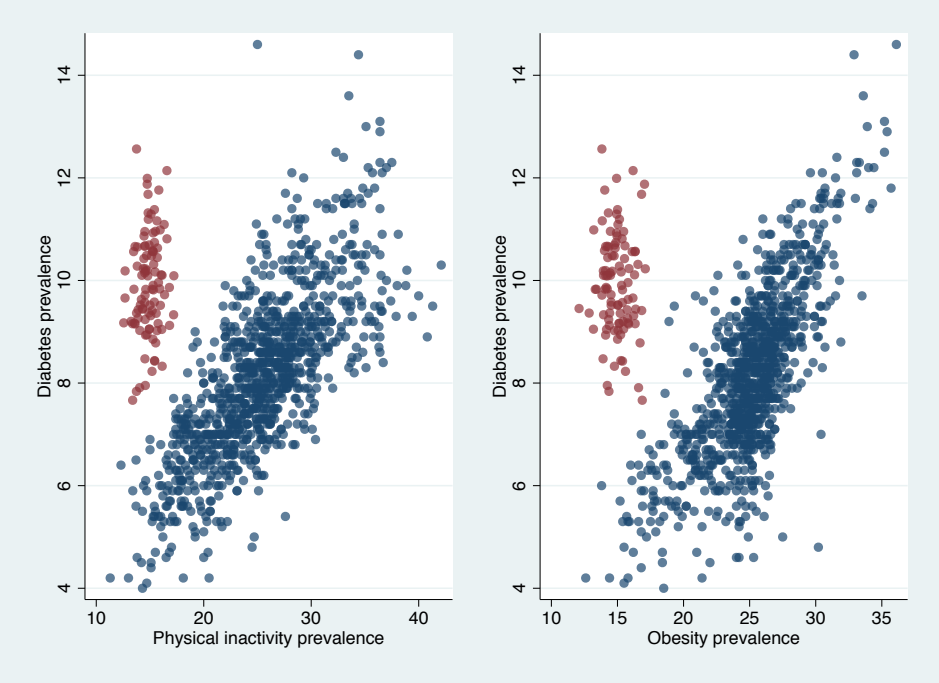
```
. est sto CLEAN
```

```
. // robust regression using clean data (no significant difference)
. robreg s perdiabet percphys percob, hausman

enumerating 50 candidates (percent completed)
0 ------ 20 ------ 40 ------ 60 ------ 80 ------ 100
..................................................

refining 2 best candidates .. done

S-Regression (28.7% efficiency)                 Number of obs     =      1,000
                                                  Subsamples      =         50
                                                  Breakdown point =         50
                                                  Bisquare k      =   1.547645
                                                  Scale estimate  =   .84987186
```

| perdiabet | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1513016 | .0163233 | 9.27 | 0.000 | .1193085 | .1832948 |
| percob | .1907011 | .02477 | 7.70 | 0.000 | .1421528 | .2392494 |
| _cons | -.5645537 | .5152196 | -1.10 | 0.273 | -1.574366 | .4452581 |

```
Hausman test of S against LS:    chi2(2) = 1.9259508      Prob > chi2 = 0.3818
```

```
. // contaminate the data
. set obs 1100
number of observations (_N) was 1,000, now 1,100
. gen byte clean    = perdiabet<.
. replace perdiabet = rnormal(10,1) if perdiabet>=.
(100 real changes made)
. replace percphys  = rnormal(15,1) if percphys>=.
(100 real changes made)
. replace percob    = rnormal(15,1) if percob>=.
(100 real changes made)
. two (scatter perdiabet percphys if clean==1, mc(%70) mlc(%0)) ///
>     (scatter perdiabet percphys if clean==0, mc(%70) mlc(%0)) ///
>     , nodraw name(a, replace) legend(off)
. two (scatter perdiabet percob if clean==1, mc(%70) mlc(%0)) ///
>     (scatter perdiabet percob if clean==0, mc(%70) mlc(%0)) ///
>     , nodraw name(b, replace) legend(off)
. graph combine a b,
```



```
. // result using contaminated data contaminated data
. // - classic regression
. reg perdiabet percphys percob
```

| Source | SS | df | MS | | |
|---|---|---|---|---|---|
| Model | 542.552632 | 2 | 271.276316 | Number of obs | = 1,100 |
| Residual | 2367.3518 | 1,097 | 2.15802351 | F(2, 1097) | = 125.71 |
| | | | | Prob > F | = 0.0000 |
| | | | | R-squared | = 0.1865 |
| | | | | Adj R-squared | = 0.1850 |
| Total | 2909.90443 | 1,099 | 2.64777473 | Root MSE | = 1.469 |

| perdiabet | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1395938 | .0120617 | 11.57 | 0.000 | .1159271 | .1632604 |
| percob | -.0347372 | .0163833 | -2.12 | 0.034 | -.0668833 | -.0025911 |
| _cons | 5.728702 | .2542869 | 22.53 | 0.000 | 5.229758 | 6.227646 |

```
. est sto OLS
. // - (semi-robust) M estimator
. robreg m perdiabet percphys percob
fitting initial LAV estimate ... done
iterating RWLS estimate ..................... done
```

M-Regression (95% efficiency)

| | |
|---|---|
| Number of obs | = 1,100 |
| Huber k | = 1.3449975 |
| Scale estimate | = 1.0614389 |
| Robust R2 (w) | = .34529835 |
| Robust R2 (rho) | = .19178653 |

| perdiabet | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1404872 | .0095426 | 14.72 | 0.000 | .121784 | .1591903 |
| percob | .034564 | .0343107 | 1.01 | 0.314 | -.0326837 | .1018116 |
| _cons | 3.825278 | .7621694 | 5.02 | 0.000 | 2.331454 | 5.319103 |

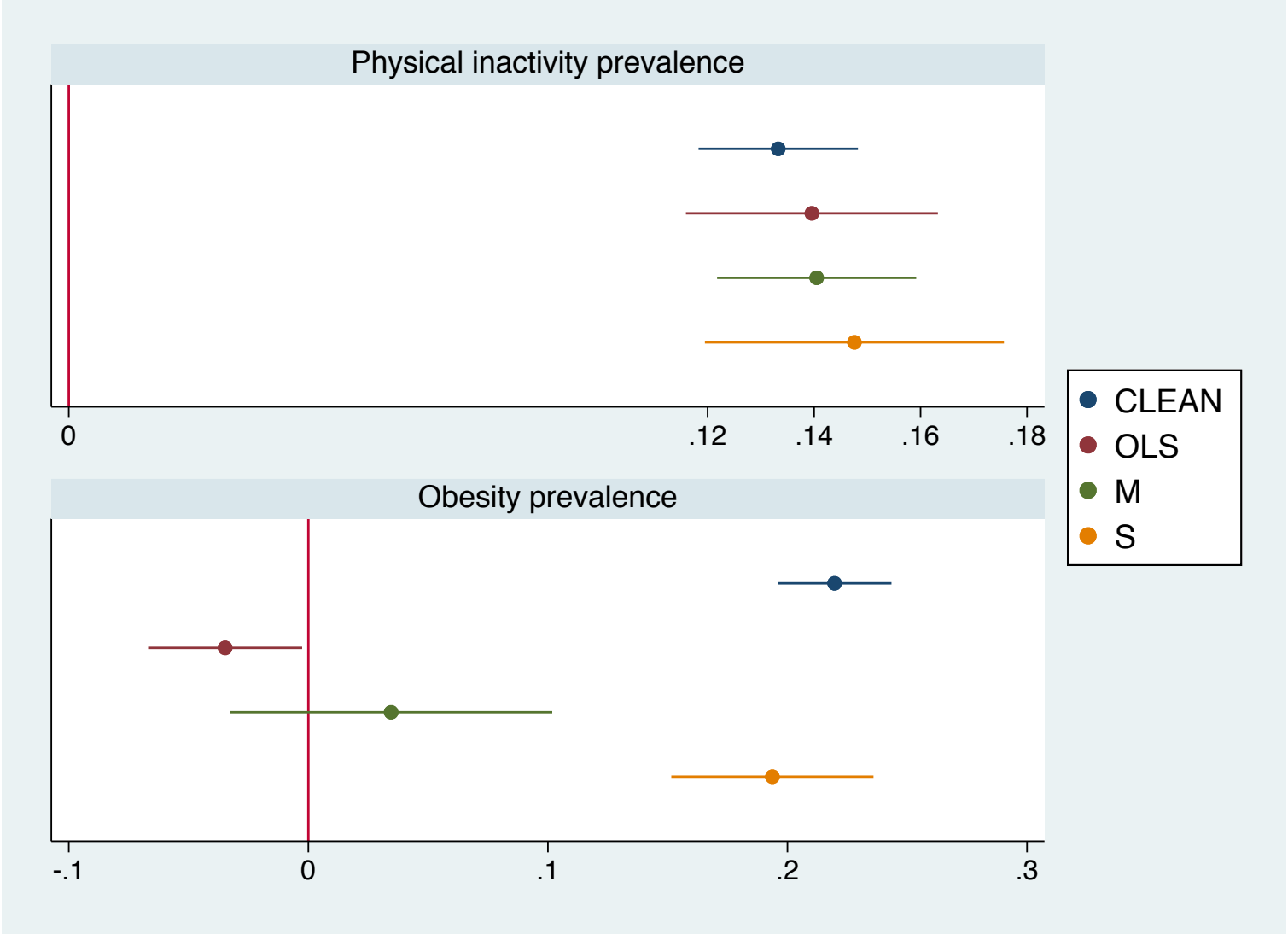
```
. est sto M
```

```
. // - (fully robust) S estimator
. robreg s perdiabet percphys percob, hausman
enumerating 50 candidates (percent completed)
0 ------ 20 ------ 40 ------ 60 ------ 80 ------ 100
..................................................
refining 2 best candidates .. done
S-Regression (28.7% efficiency)                 Number of obs     =      1,100
                                                  Subsamples      =         50
                                                  Breakdown point =         50
                                                  Bisquare k      =   1.547645
                                                  Scale estimate  =   .97331348
```

| perdiabet | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1475783 | .0143318 | 10.30 | 0.000 | .1194885 | .1756681 |
| percob | .1937171 | .0215297 | 9.00 | 0.000 | .1515196 | .2359145 |
| _cons | -.5402307 | .4590932 | -1.18 | 0.239 | -1.440037 | .3595755 |

```
Hausman test of S against LS:    chi2(2) = 102.56961      Prob > chi2 = 0.0000
. est sto S
```

```
. coefplot CLEAN OLS M S, drop(_cons) ///
>     bycoefs xlabel(0, add) xline(0) legend(cols(1)) ///
>     byopts(cols(1) noiylabel noiytick xrescale legend(pos(3)))
```



```
. // improving on the S-estimate as much as possible
. // - 85% efficiency (still ok)
. robreg mm perdiabet percphys percob, hausman
Step 1: fitting S-estimate
enumerating 50 candidates (percent completed)
0 ------ 20 ------- 40 ------- 60 ------- 80 ------ 100
..................................................
refining 2 best candidates .. done
Step 2: fitting redescending M-estimate
iterating RWLS estimate ............ done
MM-Regression (85% efficiency)                  Number of obs     =      1,100
                                                  Subsamples      =         50
                                                  Breakdown point =         50
                                                  M-estimate: k   =  3.4436898
                                                  S-estimate: k   =   1.547645
                                                  Scale estimate  =  .97331348
                                                  Robust R2 (w)   =  .73742992
                                                  Robust R2 (rho) =  .34644654
------------------------------------------------------------------------------
             |               Robust
   perdiabet |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    percphys |   .1337519   .0087649    15.26   0.000     .1165731    .1509308
      percob |    .214233   .0135823    15.77   0.000     .1876121    .2408538
       _cons |  -.6563609   .2460445    -2.67   0.008    -1.138599   -.1741225
------------------------------------------------------------------------------
Hausman test of MM against S:     chi2(2) = 2.7984133      Prob > chi2 = 0.2468
. est sto MM85
```

```
. // - 95% efficiency (still ok)
. robreg mm perdiabet percphys percob, hausman eff(95)
Step 1: fitting S-estimate
enumerating 50 candidates (percent completed)
0 ------ 20 ------- 40 ------- 60 ------- 80 ------ 100
..................................................
refining 2 best candidates .. done
Step 2: fitting redescending M-estimate
iterating RWLS estimate ......... done
MM-Regression (95% efficiency)                  Number of obs     =      1,100
                                                  Subsamples      =         50
                                                  Breakdown point =         50
                                                  M-estimate: k   =  4.6850649
                                                  S-estimate: k   =   1.547645
                                                  Scale estimate  =  .97331348
                                                  Robust R2 (w)   =  .69765832
                                                  Robust R2 (rho) =   .2958162
```

| perdiabet | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1334119 | .0083695 | 15.94 | 0.000 | .117008 | .1498157 |
| percob | .2092841 | .0144597 | 14.47 | 0.000 | .1809436 | .2376246 |
| _cons | -.5044031 | .2823584 | -1.79 | 0.074 | -1.057815 | .0490091 |

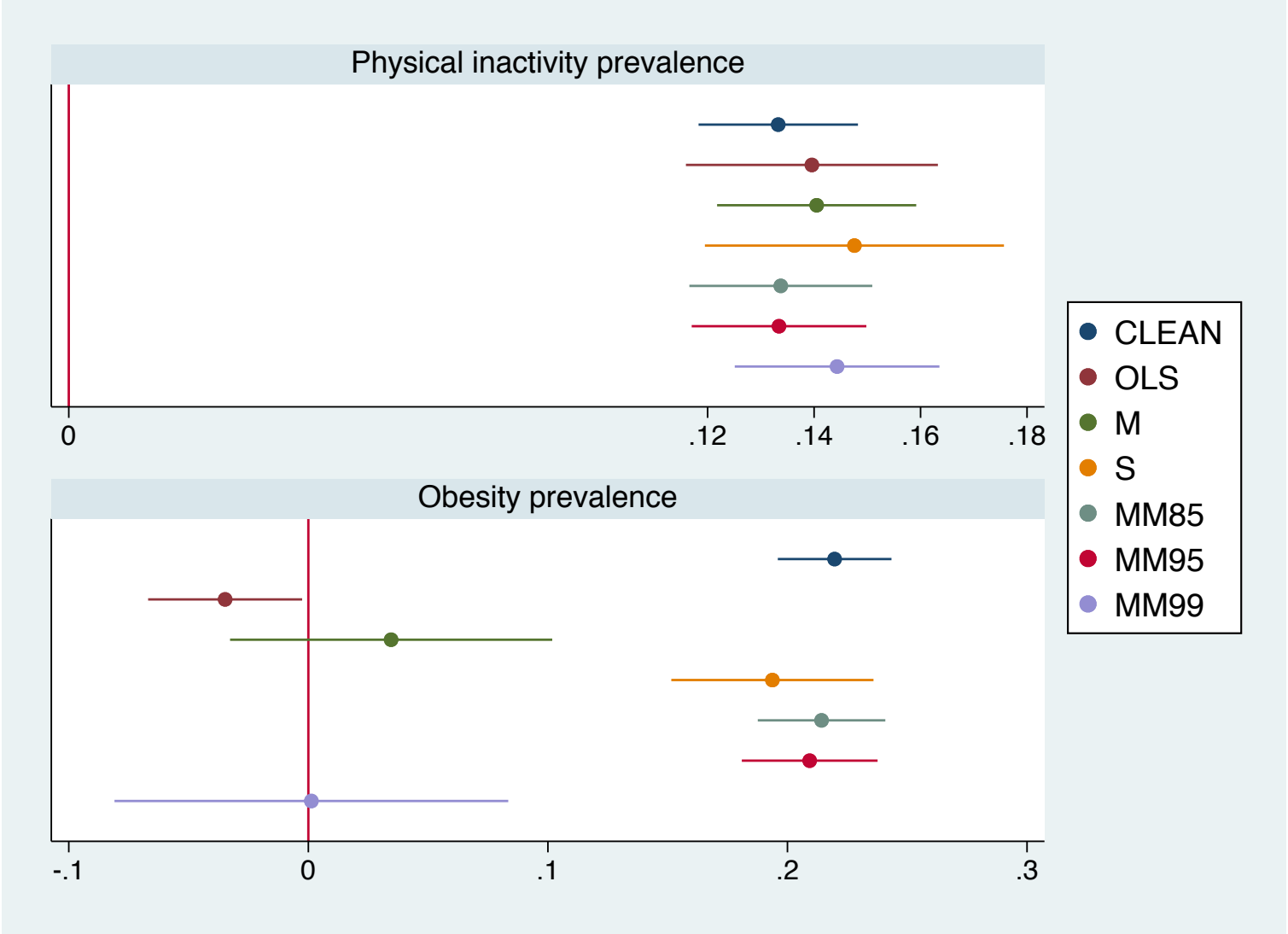
```
Hausman test of MM against S:    chi2(2) = 1.8866681      Prob > chi2 = 0.3893
. est sto MM95
```

```
. // - 99% efficiency (no longer ok)
. robreg mm perdiabet percphys percob, hausman eff(99)
Step 1: fitting S-estimate
enumerating 50 candidates (percent completed)
0 ------ 20 ------- 40 ------- 60 ------- 80 ------ 100
..................................................
refining 2 best candidates .. done
Step 2: fitting redescending M-estimate
iterating RWLS estimate ........................... done
MM-Regression (99% efficiency)                  Number of obs     =      1,100
                                                  Subsamples      =         50
                                                  Breakdown point =         50
                                                  M-estimate: k   =  7.0413916
                                                  S-estimate: k   =   1.547645
                                                  Scale estimate  =  .97331348
                                                  Robust R2 (w)   =  .29117091
                                                  Robust R2 (rho) =  .18696675
```

| perdiabet | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | .1443221 | .0098078 | 14.71 | 0.000 | .125099 | .1635451 |
| percob | .0012655 | .0419327 | 0.03 | 0.976 | -.080921 | .083452 |
| _cons | 4.609422 | .9643243 | 4.78 | 0.000 | 2.719381 | 6.499463 |

```
Hausman test of MM against S:    chi2(2) = 24.758765      Prob > chi2 = 0.0000
. est sto MM99
```

```
. coefplot CLEAN OLS M S MM*, drop(_cons) ///
>     bycoefs xlabel(0, add) xline(0) legend(cols(1)) ///
>     byopts(cols(1) noiylabel noiytick xrescale legend(pos(3)))
```



# The robmv command

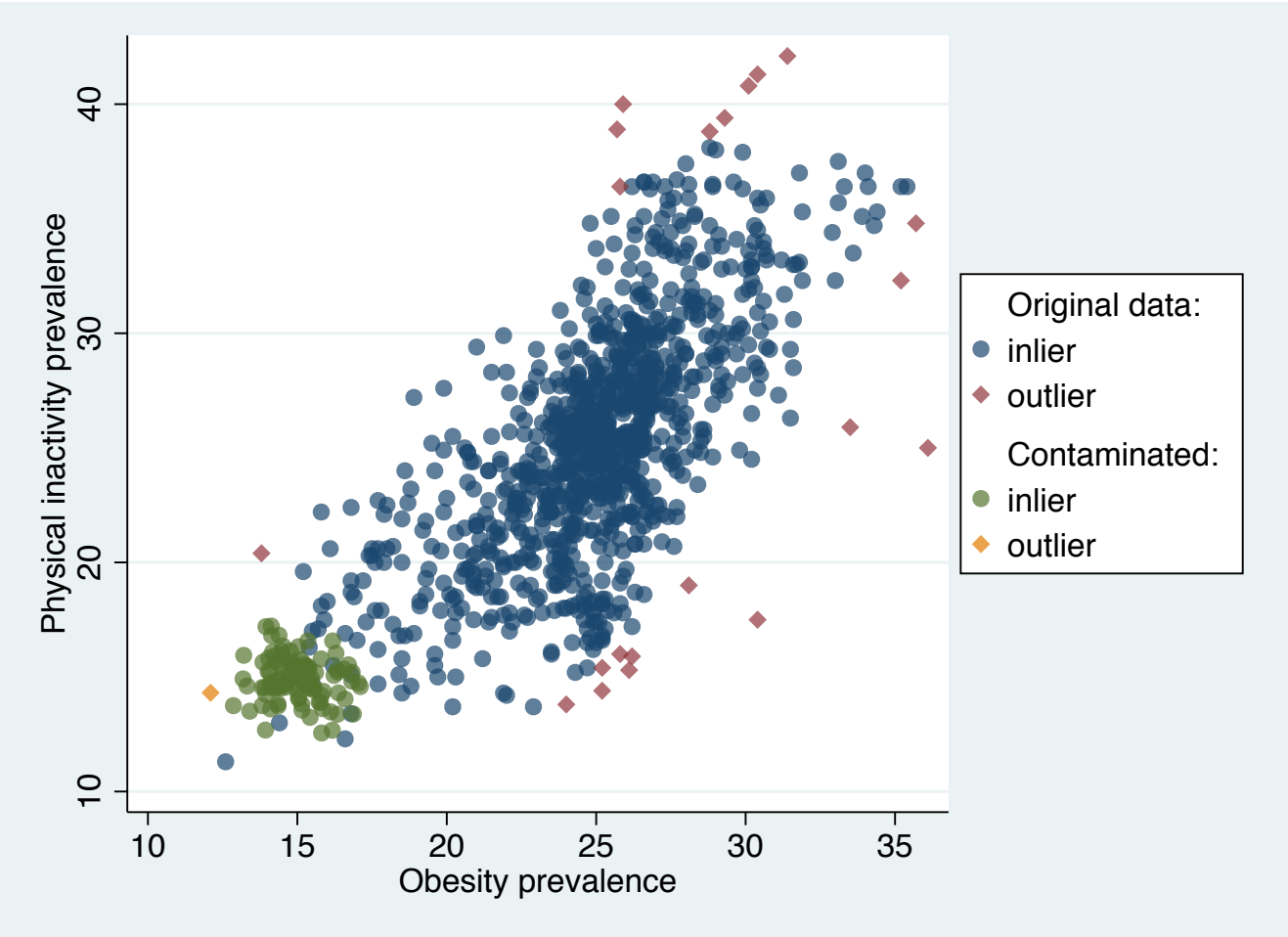
- Supports various robust estimators of location and covariance (M, MVE, MCD; S and SD yet to come).
- Efficiency-improving reweighting (Maronna et al. 2006) as well as small sample (Pison et al. 2002) and consistency correction (Croux and Haesbroeck 1999).
- Fast H-subset search algorithms (Rousseeuw and Van Driessen 1999).
- Postestimation computation of robust distances, outlier identification, etc.

# Examples for robmv

```
. // classical estimate
. robmv m percphys percob, ptrim(0) vce(boot)
(running robmv on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Huber M-estimate (.% BP)                        Number of obs     =      1,100
                                                Replications      =         50
                                                Winsorizing (%)   =          0
                                                Tuning constant   =          .
```

| Cov | Observed Coef. | Bootstrap Std. Err. | z | P>\|z\| | Normal-based [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | | | | | | |
| percphys | 33.79778 | 1.309019 | 25.82 | 0.000 | 31.23215 | 36.36341 |
| percob | 19.28447 | .8406838 | 22.94 | 0.000 | 17.63676 | 20.93218 |
| percob | | | | | | |
| percob | 18.31911 | .7867742 | 23.28 | 0.000 | 16.77706 | 19.86116 |
| _location | | | | | | |
| percphys | 24.44981 | .1947152 | 125.57 | 0.000 | 24.06817 | 24.83144 |
| percob | 24.15853 | .1362999 | 177.25 | 0.000 | 23.89139 | 24.42567 |

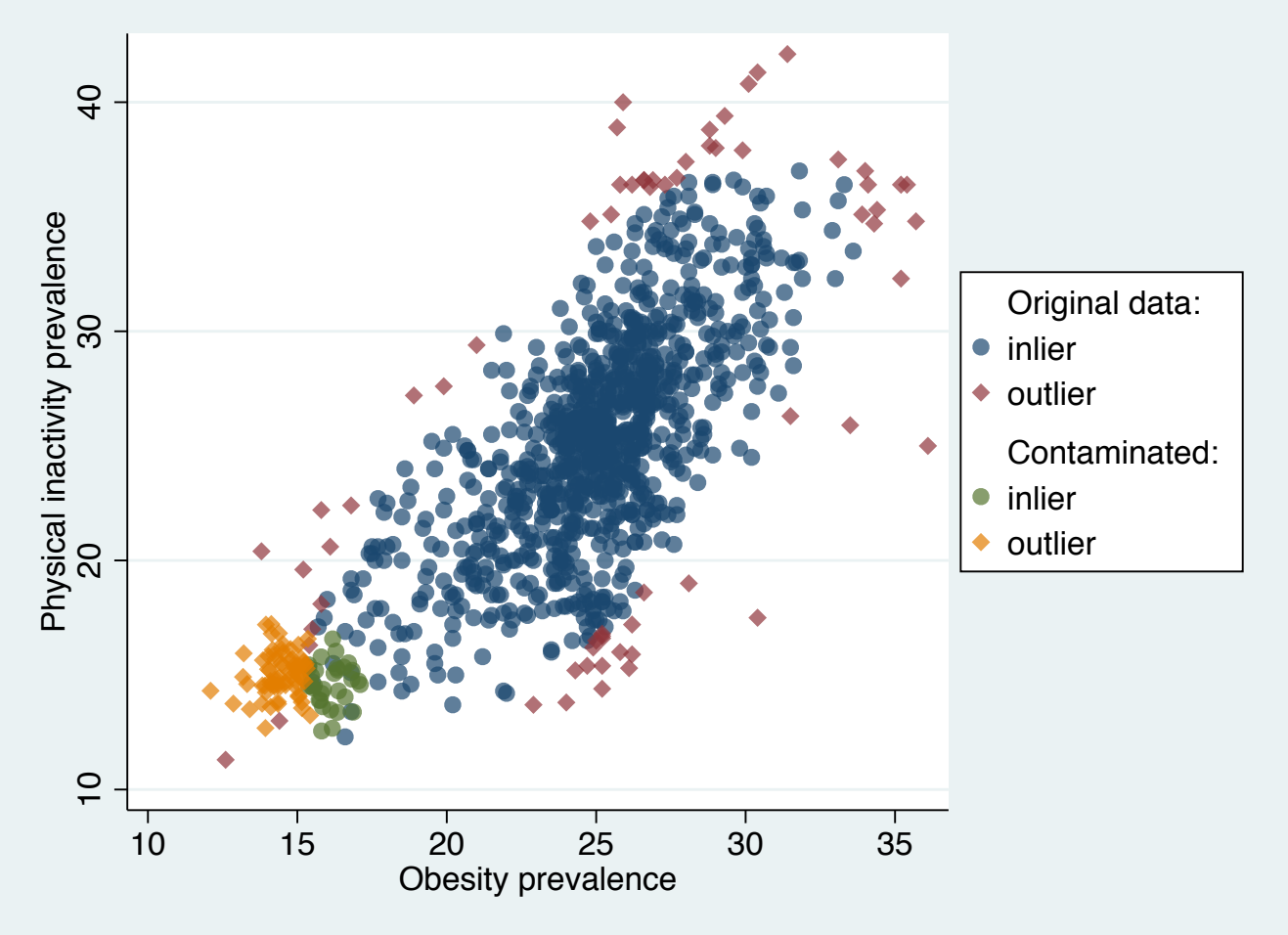
```
. predict _o, outlier
. two (scatter percphys percob if _o==0 & clean==1, mc(%70) mlc(%0)) ///
>     (scatter percphys percob if _o==1 & clean==1, mc(%70) mlc(%0) ms(d)) ///
>     (scatter percphys percob if _o==0 & clean==0, mc(%70) mlc(%0)) ///
>     (scatter percphys percob if _o==1 & clean==0, mc(%70) mlc(%0) ms(d)) ///
>     , legen(cols(1) pos(3) order(- "Original data:" 1 "inlier" 2 "outlier" ///
>         - - "Contaminated:" 3 "inlier" 4 "outlier"))
. drop _o
```



```
. // M-estimate (with maximum breakdown of 33%)
. robmv m percphys percob, vce(boot)
(running robmv on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Huber M-estimate (33.3% BP)                     Number of obs     =      1,100
                                                Replications      =         50
                                                Winsorizing (%)   =  22.313016
                                                Tuning constant   =  1.7320508
```

| Cov | Observed Coef. | Bootstrap Std. Err. | z | P>\|z\| | Normal-based [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | | | | | | |
| percphys | 24.26213 | 1.47271 | 16.47 | 0.000 | 21.37567 | 27.14859 |
| percob | 12.66472 | .8738919 | 14.49 | 0.000 | 10.95192 | 14.37752 |
| percob | | | | | | |
| percob | 11.46751 | .8394741 | 13.66 | 0.000 | 9.822175 | 13.11285 |
| _location | | | | | | |
| percphys | 24.81437 | .1422826 | 174.40 | 0.000 | 24.5355 | 25.09324 |
| percob | 24.52607 | .1078412 | 227.43 | 0.000 | 24.3147 | 24.73743 |

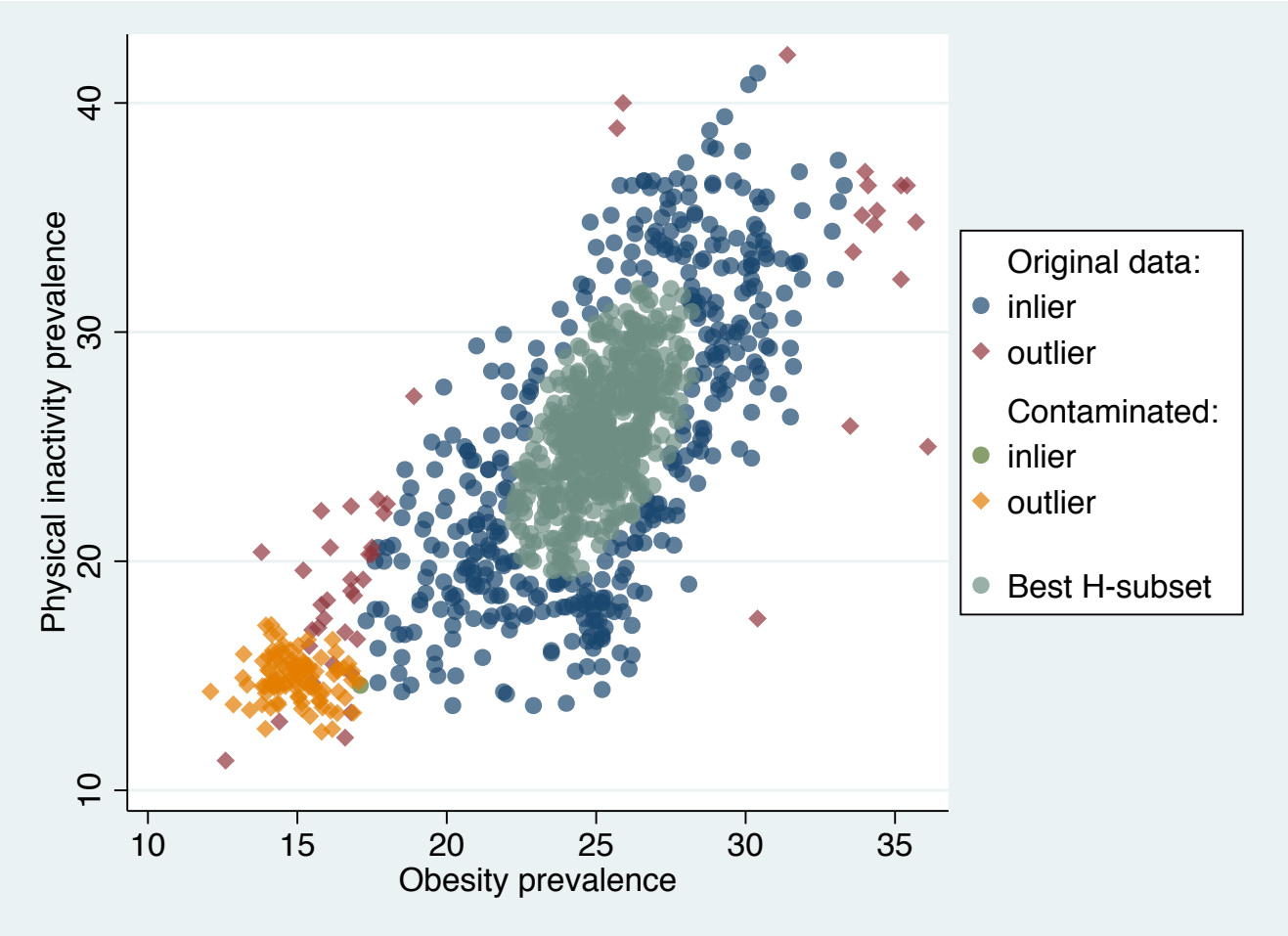
```
. predict _o, outlier
. two (scatter percphys percob if _o==0 & clean==1, mc(%70) mlc(%0)) ///
>     (scatter percphys percob if _o==1 & clean==1, mc(%70) mlc(%0) ms(d)) ///
>     (scatter percphys percob if _o==0 & clean==0, mc(%70) mlc(%0)) ///
>     (scatter percphys percob if _o==1 & clean==0, mc(%70) mlc(%0) ms(d)) ///
>     , legen(cols(1) pos(3) order(- "Original data:" 1 "inlier" 2 "outlier" ///
>         - - "Contaminated:" 3 "inlier" 4 "outlier"))
. drop _o
```



```
. // Reweighted MCD-estimate (50% breakdown)
. robmv mcd percphys percob, vce(boot)
(running robmv on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

MCD estimate (50% BP; 97.5% reweighting)      Number of obs     =      1,100
                                              Replications      =         50
                                              Size of H-subset  =        551
                                              MCD (log)         =  2.1429965

Algorithm = random    Candidates  =     500   Candidate C-steps =          2
                      Final cand. =      10   Final C-steps     =   converge
                      Subsamples  =       3   Merged subs. size =       1100
```

| Cov | Observed Coef. | Bootstrap Std. Err. | z | P>\|z\| | Normal-based [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| percphys | | | | | | |
| percphys | 35.55331 | 2.246265 | 15.83 | 0.000 | 31.15071 | 39.95591 |
| percob | 11.62844 | 1.015272 | 11.45 | 0.000 | 9.638549 | 13.61834 |
| percob | | | | | | |
| percob | 9.146134 | .777975 | 11.76 | 0.000 | 7.621331 | 10.67094 |
| _location | | | | | | |
| percphys | 25.53709 | .2116731 | 120.64 | 0.000 | 25.12221 | 25.95196 |
| percob | 25.28146 | .0950847 | 265.88 | 0.000 | 25.09509 | 25.46782 |

```
. predict _o, outlier
. predict _s, subset
. two (scatter percphys percob if _s==0 & _o==0 & clean==1, mc(%70) mlc(%0)) ///
>     (scatter percphys percob if _s==0 & _o==1 & clean==1, mc(%70) mlc(%0) ms(d)) ///
>     (scatter percphys percob if _s==0 & _o==0 & clean==0, mc(%70) mlc(%0)) ///
>     (scatter percphys percob if _s==0 & _o==1 & clean==0, mc(%70) mlc(%0) ms(d)) ///
>     (scatter percphys percob if _s==1                   , mc(%70) mlc(%0)) ///
>     , legen(cols(1) pos(3) order(- "Original data:" 1 "inlier" 2 "outlier" ///
>         - - "Contaminated:" 3 "inlier" 4 "outlier" - - - 5 "Best H-subset"))
. drop _o _s
```



# The roblogit command

- Implementation of several robust logistic regression estimators, relying on robust estimation of location and covariance (`robmv`) to identify high-leverage points.
  - Weighted maximum likelihood.
  - M-estimator as proposed by Bianco and Yohai (1996).
  - M-estimator as proposed by Croux and Haesbroeck (2003).

# Examples for roblogit

```
. clear all
. use titanic.dta
. generate female = sex=="female" if inlist(sex,"male","female")
(1 missing value generated)
. tab pclass, gen(class) nofreq
. lab var class1 "first class"
. lab var class2 "second class"
. lab var class3 "third class"
. gen lnfare = ln(fare+1)
(2 missing values generated)
```

```
. // classic logit
. logit survived age female class1 class3 lnfare

Iteration 0:   log likelihood = -706.78527
Iteration 1:   log likelihood = -494.16661
Iteration 2:   log likelihood = -490.65117
Iteration 3:   log likelihood = -490.63585
Iteration 4:   log likelihood = -490.63585

Logistic regression                             Number of obs     =      1,045
                                                LR chi2(5)        =     432.30
                                                Prob > chi2       =     0.0000
Log likelihood = -490.63585                     Pseudo R2         =     0.3058

------------------------------------------------------------------------------
    survived |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.0354682   .0064402    -5.51   0.000    -.0480908   -.0228457
      female |   2.532131   .1697653    14.92   0.000     2.199397    2.864864
      class1 |   1.437476   .2712539     5.30   0.000     .9058277    1.969123
      class3 |  -1.079723   .2102919    -5.13   0.000    -1.491887   -.6675583
      lnfare |   -.128688   .1221912    -1.05   0.292    -.3681783    .1108024
       _cons |   .1471451   .4536999     0.32   0.746    -.7420904    1.036381
------------------------------------------------------------------------------

. est sto logit
```

```
. // Weighted maximum likelihood
. roblogit survived age female class1 class3 lnfare, ml
determining leverage outliers for: age lnfare ... done

Iteration 0:   Maximum-Likelihood f() =  643.14639
Iteration 1:   Maximum-Likelihood f() =  459.38352
Iteration 2:   Maximum-Likelihood f() =  455.67669
Iteration 3:   Maximum-Likelihood f() =  455.67068
Iteration 4:   Maximum-Likelihood f() =  455.67068

Maximum-likelihood logistic regression          Number of obs     =      1,045
                                                Wald chi2(5)      =     260.67
                                                Prob > chi2       =     0.0000
                                                  Minimized f()   =   455.6707
                                                  Weighting step  =        mcd
                                                  Cutoff p-value  =       .975
                                                  Leverage points =         85
```

| survived | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| age | -.037228 | .0073552 | -5.06 | 0.000 | -.051644 | -.0228121 |
| female | 2.488124 | .1731912 | 14.37 | 0.000 | 2.148675 | 2.827572 |
| class1 | 1.559394 | .2840594 | 5.49 | 0.000 | 1.002648 | 2.116141 |
| class3 | -1.13753 | .2198864 | -5.17 | 0.000 | -1.5685 | -.7065606 |
| lnfare | -.2206801 | .1785078 | -1.24 | 0.216 | -.570549 | .1291888 |
| _cons | .4938637 | .6212811 | 0.79 | 0.427 | -.7238249 | 1.711552 |

```
. est sto WML
```

```
. // Bianco-Yohai M-estimator
. roblogit survived age female class1 class3 lnfare, byohai
determining leverage outliers for: age lnfare ... done

Iteration 0:   Bianco-Yohai f() =  400.58539
Iteration 1:   Bianco-Yohai f() =  269.25704
Iteration 2:   Bianco-Yohai f() =  260.31399
Iteration 3:   Bianco-Yohai f() =  259.61052
Iteration 4:   Bianco-Yohai f() =  259.59776
Iteration 5:   Bianco-Yohai f() =  259.59771
Iteration 6:   Bianco-Yohai f() =  259.59771

Bianco-Yohai logistic regression                Number of obs     =      1,045
                                                Wald chi2(5)      =     147.79
                                                Prob > chi2       =     0.0000
                                                  Tuning constant =   3.506558
                                                  Minimized f()   =   259.5977
                                                  Weighting step  =        mcd
                                                  Cutoff p-value  =       .975
                                                  Leverage points =         84
```

| survived | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| age | -.0395028 | .0086657 | -4.56 | 0.000 | -.0564873 | -.0225182 |
| female | 2.628858 | .256361 | 10.25 | 0.000 | 2.1264 | 3.131316 |
| class1 | 1.566123 | .3009247 | 5.20 | 0.000 | .9763215 | 2.155925 |
| class3 | -1.48135 | .3475514 | -4.26 | 0.000 | -2.162539 | -.8001623 |
| lnfare | -.3024237 | .1955151 | -1.55 | 0.122 | -.6856262 | .0807789 |
| _cons | .7779488 | .6697682 | 1.16 | 0.245 | -.5347728 | 2.09067 |

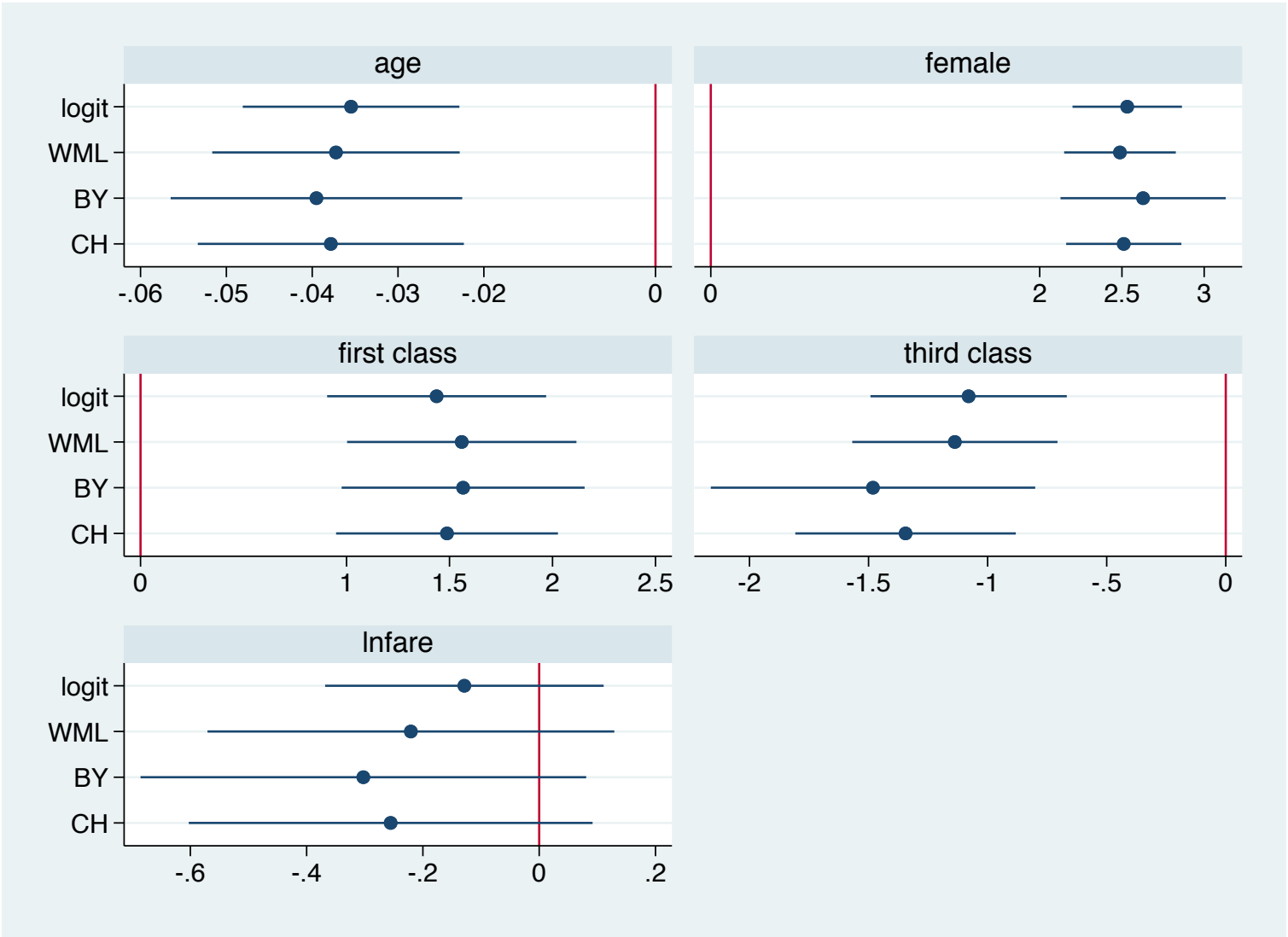
```
. est sto BY
```

```
. // Croux-Haesbroeck M-estimator
. roblogit survived age female class1 class3 lnfare
determining leverage outliers for: age lnfare ... done

Iteration 0:   Croux-Haesbroeck f() =  1082.5906
Iteration 1:   Croux-Haesbroeck f() =  1020.5806
Iteration 2:   Croux-Haesbroeck f() =  1018.1186
Iteration 3:   Croux-Haesbroeck f() =  1007.8863
Iteration 4:   Croux-Haesbroeck f() =  1007.5209
Iteration 5:   Croux-Haesbroeck f() =  1007.5201
Iteration 6:   Croux-Haesbroeck f() =  1007.5201

Croux-Haesbroeck logistic regression          Number of obs     =      1,045
                                              Wald chi2(5)      =     235.72
                                              Prob > chi2       =     0.0000
                                                Tuning constant =         .5
                                                Minimized f()   =    1007.52
                                                Weighting step  =        mcd
                                                Cutoff p-value  =       .975
                                                Leverage points =         85
```

| survived | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| age | -.0378282 | .0079094 | -4.78 | 0.000 | -.0533303 | -.022326 |
| female | 2.511185 | .1787137 | 14.05 | 0.000 | 2.160913 | 2.861457 |
| class1 | 1.487809 | .2747335 | 5.42 | 0.000 | .9493408 | 2.026276 |
| class3 | -1.34463 | .2362298 | -5.69 | 0.000 | -1.807632 | -.8816285 |
| lnfare | -.2554476 | .1771504 | -1.44 | 0.149 | -.6026559 | .0917608 |
| _cons | .6409812 | .6052633 | 1.06 | 0.290 | -.5453131 | 1.827275 |

```
. est sto CH
```

```
. // not much of a differenc between models; results are fairly robust
. coefplot logit || WML || BY || CH, drop(_cons) ///
>     bycoefs byopts(xrescale cols(2)) xlabel(0, add) xline(0)
```



# Still some work to do ...

- `robreg`: some housekeeping and cleaning up, support for `svy`
- `robmv`: S and Stahel-Donoho estimators, influence functions/standard errors
- `roblogit`: some housekeeping and cleaning up (e.g. factor variables), support for `svy`
- Additional commands that are in the pipeline:
  - Robust instrumental variables regression
  - Robust fixed-effects panel regression

# References I


- Bianco, A.M., V.J. Yohai (1996). Robust Estimation in the Logistic Regression Model. Pp. 17-34 in Helmut Rieder (Ed.), Robust Statistics, Data Analysis, and Computer Intensive Methods. In Honor of Peter Huber's 60th Birthday. New York: Springer.
- Brys, G., M. Hubert, A. Struyf (2004). A Robust Measure of Skewness. Journal of Computational and Graphical Statistics 13(4): 996-1017.
- Brys, G., M. Hubert, A. Struyf (2008). Goodness-of-fit tests based on a robust measure of skewness. Computational Statistics 23: 429-442.
- Croux, C., G. Dhaene, D. Hoorelbeke (2003). Robust Standard Errors for Robust Estimators. Discussions Paper Series (DPS) 03.16. Center for Economic Studies.
- Croux, C., G. Haesbroeck (1999). Influence Function and Efficiency of the Minimum Covariance Determinant Scatter Matrix Estimator. Journal of Multivariate Analysis 71: 161 190.
- Croux, C., G. Haesbroeck (2003). Implementing the Bianco and Yohai estimator for logistic regression. Computational Statistics and Data Analysis 44(1-2): 273-295.

# References II


- Croux, C., P. J. Rousseeuw (1992). Time-efficient algorithms for two highly robust estimators of scale. P. 411-428 in: Y. Dodge and J. Whittaker (eds.). Computational Statistics. Heidelberg: Physica-Verlag.
- Johnson, D. B., T. Mizoguchi (1978). Selecting the $K$th element in $X + Y$ and $X_1 + X_2 + ... + X_m$. SIAM Journal on Scientific Computing 7(2): 147–153.
- Koller, M. 2012. Nonsingular subsampling for S-estimators with categorical predictors. ETH Zurich. arXiv:1208.5595v1
- Maronna, R. A., D. R. Martin, V. J. Yohai (2006). Robust Statistics. Theory and Methods. Chichester: John Wiley & Sons.
- Pison, G., S. Van Aelst, G. Willems (2002). Small sample corrections for LTS and MCD. Metrika 55: 111-123.
- Rousseeuw, P.J., K. Van Driessen (1999). A Fast Algorithm for the Minimum Covariance Determinant Estimator. Technometrics 41(3): 212-223.
- Salibian-Barrera, M., V. J. Yohai (2006). A Fast Algorithm for S-Regression Estimates. Journal of Computational and Graphical Statistics 15: 414-427.