Statistics/Data analysis

help robreg

Title

robreg —— Robust regression

Contents

Syntax

Estimation

robreg *subcmd* *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

where *subcmd* is

ls	least squares regression; equivalent to **regress**
q	quantile regression; similar to **qreg**
m	M regression
s	S regression
mm	MM regression
lts	least trimmed squares regression
lqs	least quantile of squares regression
lms	least median of squares regression

indepvars may contain factor variables; see fvvarlist.
depvar and *indepvars* may contain time-series operators; see tsvarlist.
fweights and **pweight**s are allowed; see weight.
The **svy** prefix is allowed with **robreg ls**, **robreg q**, and **robreg m**, unless **ivar()** or **absorb()**

Refitting MM after **robreg s** or **robreg mm**

robreg mm [, *mm_refit_options*]

Replaying results

robreg [, *reporting_options*]

Hausman test between models

robreg hausman [*model1* [*model2*]] [, *hausman_options*]

where *model1* and *model2* are names of stored estimates from **robreg** (or **.** for the active estim
used if *model1* is also omitted).

Prediction

predict [*type*] {*newvar* | *stub**** | *newvar1* *newvar2* ...} [*if*] [*in*] [, *predict_options*]

options	description
Main	
noconstant	suppress constant term
nor2	do not compute the (pseudo) R-squared

Subcommand	
ls_options	additional options for **robreg ls**
q_options	additional options for **robreg q**
m_options	additional options for **robreg m**
s_options	additional options for **robreg s**
mm_options	additional options for **robreg mm**
lts_options	additional options for **robreg lts**
lqs/lms_options	additional options for **robreg lqs/lms**
VCE/SE	
vce(*vcetype***)**	variance estimation method; *vcetype* may be **robust**, **cluster** *clustvar*, **boot**
cluster(*clustvar***)**	synonym for **vce(cluster** *clustvar***)**
ftest	report F tests rather then Wald tests
nose	do not compute standard errors
Reporting	
level(*#***)**	set confidence level; default is **level(95)**
noheader	suppress table header
notable	suppress table of results
all	report results from all equations (relevant for **robreg s** and **robreg mm**)
display_options	standard reporting options as described in **[R] estimation options**
Optimization	
tolerance(*#***)**	tolerance for iterative algorithms
iterate(*#***)**	maximum number of iterations
relax	do not return error if convergence is not reached
noquad	do not use quad precision when taking cross products
nolog	do not display progress information

ls_options	description
fe_options	fixed-effects options

q_options	description
Main	
quantile(*#***)**	estimate # quantile; the default is 0.5 (LAD or median regression)
VCE/SE	
fitted	alternative approach to estimate the sparsity function
bofinger	use Bofinger's bandwidth rather than the Hall-Sheather bandwidth
Starting values	
init(*matname***)**	use custom starting values

m_options	description
Main	
huber	use Huber objective function; the default
biweight	use biweight objective function
bisquare	synonym for **biweight**
efficiency(*#***)**	gaussian efficiency, in percent; default is **efficiency(95)**
k(*#***)**	tuning constant; alternative to **efficiency()**
fe_options	fixed-effects options
Scale	
scale(*#***)**	provide custom (starting) value for scale
updatescale	update scale in each iteration
center	center residuals when computing scale
Starting values	
init(*arg***)**	set the starting values; default is **init(lad)**

s_options	description
Main	
bp(*#***)**	breakdown point, in percent; default is **bp(50)**
k(*#***)**	tuning constant; alternative to **bp()**
nohausman	do not compute Hausman test of S against LS
m(*varlist* [**,** *opts*]**)**	variables to be partialled out in the subsampling algorithm
fe_options	fixed-effects options
Subsampling algorithm	
nsamp(*args***)**	number of trial samples; *args* may be *#* or *alpha* [*epsilon*]
rsteps(*#***)**	number of local improvement steps; default is **rsteps(2)**
nkeep(*#***)**	number of candidates to be kept for final refinement; default is **nkeep(5)**
naive	use naive subsampling algorithm
alt	use alternative nonsingular subsampling algorithm
nostd	do not standardize the data (not recommended)

mm_options	description
Main	
efficiency(*#***)**	gaussian efficiency, in percent; default is **efficiency(85)**
k(*#***)**	tuning constant; alternative to **efficiency()**
nohausman	do not compute Hausman test of MM against S
S estimate	
bp(*#***)**	breakdown point, in percent; default is **bp(50)**
sopts(*options***)**	other options passed through to the S algorithm

lts_options	description
Main	
bp(*#***)**	breakdown point, in percent; default is **bp(50)**
Subsampling algorithm	
nsamp(*args***)**	number of trial samples; *args* may be *#* or *alpha* [*epsilon*]
csteps(*#***)**	number of concentration steps; default is **csteps(2)**
nkeep(*#***)**	number of candidates to be kept for final refinement; default is **nkeep(10**
naive	use naive subsampling algorithm
alt	use alternative nonsingular subsampling algorithm
nostd	do not standardize the data (not recommended)

lqs/lms_options	description
Main	
bp(*#***)**	breakdown point, in percent (only allowed with **robreg lqs**)
Subsampling algorithm	
nsamp(*args***)**	number of trial samples; *args* may be *#* or *alpha* [*epsilon*]
naive	use naive subsampling algorithm
alt	use alternative nonsingular subsampling algorithm
nostd	do not standardize the data (not recommended)

fe_options	description
ivar(*varname***)**	variable identifying the groups; VCE like **xtreg,fe**
absorb(*varname***)**	variable identifying the groups; VCE like **areg**
[**no**]**usave**	whether to store fixed effects in **e()**
ugenerate(*newvar***)**	store fixed effects as a variable, rather than in **e()**
replace	allow overwriting existing variable

mm_refit_options	description
efficiency(*#***)**	gaussian efficiency, in percent; default is **efficiency(85)**
k(*#***)**	tuning constant; alternative to **efficiency()**
nohausman	do not compute Hausman test of MM against S
ftest	report F test rather then Wald test
reporting_options	reporting options
optimization_options	optimization options

hausman_options	description
constant	include constant in the test
common	restrict the test to coefficients that exist in both models
keep(*names***)**	coefficients to be included; may use ***** and **?** wildcards
drop(*names***)**	coefficients to be excluded; may use ***** and **?** wildcards
ftest	report F test rather then Wald test
level(*#***)**	set confidence level; default is **level(95)**
noheader	suppress header containing the overall test
notable	suppress table containing tests by individual coefficients
display_options	standard reporting options as described in **[R] estimation options**
post	post results in **e()**

predict_options	description
xb	linear prediction
* **xbu**	linear prediction plus fixed effect
* **u**	fixed effect
* **ue**	fixed effect plus residual
residuals	residuals
rstandard	standardized residuals
outlier[**(***#***)**]	outlier indicator
inlier[**(***#***)**]	inlier indicator
§ **weights**	RLS weight
# **subset**	H-subset indicator
+ **scores**	equation-level scores
+ **ifs**	coefficient-level influence functions
+ **rifs**	recentered influence functions

* only after **robreg ls**, **m**, or **s** with option **ivar()** or **absorb()**
§ only after **robreg m**, **s**, or **mm**
# only after **robreg lts**, **lqs**, or **lms**
+ only after **robreg ls**, **q**, **m**, **s**, or **mm**

Description

robreg provides a number of robust estimators for linear regression models (see, e.g., Maronna e

robreg ls fits classical (non-robust) least-squared models. Results are equivalent to results fr

robreg q fits quantile regression models (median or LAD regression by default) using the interio

robreg m fits M regression models (Huber 1973) using iteratively reweighted least squares (IRLS)

robreg s fits high breakdown S regression models (Rousseeuw and Yohai 1984) based on random subs

robreg mm fits MM regression models (Yohai 1987), combining an initial high breakdown S estimato

robreg lts fits least trimmed squares (LTS) regression models based on random subsampling with l
lts.

robreg lqs and **robreg lms** fit least quantile of squares (LQS) and least median of squares (LMS)
1997). Influence-function based variance estimation is not supported by **robreg lqs** and **robreg lm**

robreg hausman can be used after **robreg ls**, **robreg q**, **robreg m**, **robreg s**, and **robreg mm** to perfo

robreg requires **moremata**. See ssc describe moremata.

Options

Main

noconstant suppresses the constant term (intercept) in the model. With robust estimators, omitti
the model contains a factor variable without base level (see operator **ibn.** in help **fvvarlist**

Note that **robreg** always computes the R-squared with respect to a constant-only model, even i

nor2 skips the computation of the (pseudo) R-squared. Use this option to save computer time.

Additional options for robreg ls

fe_options are options to include fixed-effects in the model; see Fixed effects options below.

Additional options for robreg q

quantile(*#***)** specifies the quantile to be estimated, with # as a proportion between 0 and 1 (excl

fitted computes the sparsity function (which is needed for variance estimation) based on fitted
sparsity function based on nonparametric kernel density estimation (using a gaussian kernel)

bofinger uses Bofinger's bandwidth for the estimation of the sparsity function. The default is t
on the confidence level set by the **level()** option. For the kernel-based approach (i.e. if **fi**

init(*matname***)** provides a matrix containing custom starting values. Coefficients will be matched
used. The default is to obtain the starting values from a least-squares fit. The choice of s

Additional options for robreg m

huber uses the Huber objective function (monotone M estimator). This is the default.

biweight uses the biweight objective function (redescending M estimator). **bisquare** is a synonym

efficiency(*#***)** sets the gaussian efficiency (the asymptotic relative efficiency compared to the L
biweight. The default is **efficiency(95)**.

k(*#***)** specifies a custom tuning constant. The default is to use a tuning constant that is consist

fe_options are options to include fixed-effects in the model; see Fixed effects options below.

scale(*#***)** provides a custom (starting) value for the residual scale. The default is to use the MA

updatescale updates the scale in each iteration of the IRLS algorithm. The default is to hold th

center uses median-centered residuals to compute the MADN. The default is to use raw residuals.

init(*arg***)** determines the starting values of the IRLS algorithm. *arg* may be **lad** (median regressio
will be taken from the first row of matrix *matname* (coefficients will be matched by name; st

If option **ivar()** or **absorb()** is specified, the fixed-effects (quasi) quantile estimator impl
absorb().

Additional options for robreg s

bp(*#***)** sets the breakdown point, in percent, with # between 1 and 50. The default is **bp(50)**.

k(*#***)** specifies a custom tuning constant. The default is to use a tuning constant that is consist

nohausman suppresses the Hausman test of the S estimate against the least-squares fit.

m(*varlist* [**,** *options*]**)** identifies outlier-free variables to be partialled out when searching for provided, the dependent variable and the remaining independent variables are residualized by candidate fit from the residualized data. The candidate fit will be augmented by the coeffic Specifying **m()** can lead to significant improvements in computational speed and can also have categorical variables). *options* are **efficiency()** or **k()** to set the tuning constant of the Hu estimator proposed by Maronna and Yohai (2000), but it results in a true S estimator (in con trap; the candidates are then optimized in the same way as without **m()** option).

fe_options are options to include fixed-effects in the model; see Fixed effects options below.

nsamp(*args***)** determines the number of trial samples for the search algorithm. Specify **nsamp(***#***)** to

```
ceil(ln(alpha) / ln(1 - (1 - epsilon)^k))
```

with a minimum of 20 and a maximum of 1000, where *k* is the number of predictors included in *epsilon* is the assumed maximum fraction of contaminated data. The default is to determine th

rsteps(*#***)** specifies the number of local improvement (refinement) steps applied to the trial cand

nkeep(*#***)** specifies the number of best candidates to be kept for final refinement. The default is

naive uses a naive subsampling algorithm. In each replication, the algorithm keeps on drawing ra default is to use the fast nonsingular subsampling algorithm suggested by Koller (2012) that

alt uses an alternative nonsingular subsampling algorithm that expands the size of the random su the default algorithm, but it may be less robust.

nostd omits standardization of the data for the subsampling algorithm. To avoid precision proble (or only by dividing by MADNs if **noconstant** has been specified). Specifying **nostd** is not rec

Additional options for robreg mm

efficiency(*#***)** sets the gaussian efficiency of the bisquare M estimator, in percent. *#* must be be

k(*#***)** specifies a custom tuning constant for the bisquare M estimator. The default is to use a tu

nohausman suppresses the Hausman test of the MM estimate against the S estimate.

bp(*#***)** sets the breakdown point of the preliminary S estimator, in percent. *#* must be between 1 a

sopts(*options***)** are (other) options to be passed through to the preliminary S estimator. *options* Optimization (the default is to use the same optimization settings as for the bisquare M est

Additional options for robreg lts

bp(*#***)** sets the breakdown point, in percent, with *#* between 1 and 50. The default is **bp(50)**. **bp()**

```
h = (floor((1-#/100)*N) + floor(#/100*(K + 1))) / N
```

where *N* is the sample size and *K* is the number of (non-omitted) coefficients in the model.

nsamp(*args***)** determines the number of trial samples for the search algorithm. Specify **nsamp(***#***)** to (with a minimum of 50 and a maximum of 10'000).

csteps(*#***)** specifies the number of concentration steps applied to the trial candidates. The defau

nkeep(*#***)** specifies the number of best candidates to be kept for final refinement. The default is

naive, **alt**, and **nostd** are described above under Additional options for robreg s.

Additional options for robreg lqs/lms

bp(*#***)** sets the breakdown point, in percent, with *#* between 1 and 50 (only allowed with **robreg lq**

nsamp(*args***)** determines the number of trial samples for the search algorithm. Specify **nsamp(***#***)** to (with a minimum of 500 and a maximum of 10'000).

naive, **alt**, and **nostd** are described above under Additional options for robreg s.

VCE/SE

vce(*vcetype***)** determines how standard errors are computed. *vcetype* may be:

robust
cluster *clustvar*
bootstrap [, *bootstrap_options*]
jackknife [, *jackknife_options*]

vce(robust), the default, computes heteroscedasticity-robust standard errors based on influe
based on influence function allowing for intragroup correlation, where *clustvar* specifies to
respectively; see help *vce_option*. **vce(robust)** and **vce(cluster)** are not supported by **robreg**

cluster(*clustvar***)** can be used as a synonym for **vce(cluster** *clustvar***)**.

ftest requests that the overall model test and the Hausman test (in case of **robreg s** and **robreg**
the number of observations minus the number of (non-omitted) parameters in the (main equatio
are used for the t tests of the individual coefficients.

nose omits the computation of standard errors. Use this option to save computer time.

Reporting

level(*#***)** specifies the confidence level, as a percentage, for confidence intervals. The default

noheader suppresses the display of the table header.

notable suppresses the display of the table of results.

all reports results from all equations in case of **robreg s** and **robreg mm**. The default is to disp

display_options are standard reporting options such as **eform**, **cformat()**, or **coeflegend**; see **[R]**

Optimization

tolerance(*#***)** specifies the convergence tolerance for iterative estimation procedures. When the m
convergence criterion is satisfied. The default is **tolerance(1e-10)**.

An exception is **robreg q**, for which convergence is evaluated based on the duality gap (conve

iterate(*#***)** specifies the maximum number of iterations. Error will be returned if convergence is

relax causes **robreg** to proceed even if convergence is not reached.

noquad requests that double precision be used rather than quad precision when taking cross produ
loss if the data is not well behaved (unreasonable means, high collinearity). Note that **noqu**
subsampling algorithm does not use quad precision because it operates on standardized data;

nolog suppresses the display of progress information.

Fixed effects options

ivar(*varname***)** specifies a variable identifying groups for which fixed effects are to be included
behavior of **xtreg,fe** with option **robust**, that is, it implies clustering on the group variabl
must be nested within clusters).

absorb(*varname***)** is an alternative to **ivar()** that mimics the behavior of **areg** (no clustering impl
in which the number of groups does not increase with the sample size.

[**no**]**usave** decides whether a fixed-effects lookup table is stored in matrix **e(u)**. For **robreg ls** t
absorb(), the default is **usave**. Typing **nousave** for these models implies that *predict_options*

ugenerate(*newvar***)** stores the fixed effects as a variable in the data, rather than storing a look

replace allows overwriting an existing variable.

Refitting options for robreg mm

efficiency(*#***)** sets the gaussian efficiency of the bisquare M estimator, in percent. # must be be

k(*#***)** specifies a custom tuning constant for the bisquare M estimator. The default is to use a tu

nohausman suppresses the Hausman test of the MM estimate against the S estimate.

ftest requests that the overall model test and the Hausman test be reported as F tests rather th

reporting_options are reporting options as described above.

optimization_options are optimization options as described above.

Options for robreg hausman

constant includes the constant in the test. The default is to exclude the constant.

common restricts the test to coefficients that exist in both models. The default is to use all (other model.

keep(*names***)** provides a space-separated list of coefficients to be included in the test. Wildcard independently of **keep()**.

keep(*names***)** provides a space-separated list of coefficients to be excluded from the test. Wildca independently of **drop()**.

ftest requests that the overall test be reported as an F test rather than a Wald chi-squared tes models) minus the number of (non-omitted) parameters in the (main equation of the) larger mo freedom are also used for the t tests of the differences between coefficients

level(*#***)** specifies the confidence level, as a percentage, for the confidence intervals of the di

noheader suppresses the display of the header that contains the overall test.

notable suppresses the display of the table that contains the t-tests for the differences betwee

display_options are standard reporting options such as **eform**, **cformat()**, or **coeflegend**; see **[R]**

post stored the results in **e()**. The default is to store the results in **r()**.

Options for predict

xb calculates linear predictions.

xbu calculates linear predictions plus fixed effects. **xbu** is only allowed after option **ivar()** or

u calculates fixed effects. **u** is only allowed after option **ivar()** or **absorb()** has been applied.

ue calculates fixed effects plus residuals. **ue** is only allowed after option **ivar()** or **absorb()** h

residuals calculates residuals.

rstandard calculates standardized residuals (residuals divided by **e(scale)**).

outlier[**(***#***)**] generates a 0/1 variable identifying outliers (1 = outlier, 0 = inlier). Optional a with absolute standardized residuals larger than invnormal(1 - #/200) will be classified as

inlier[**(***#***)**] generates a 0/1 variable identifying inliers (1 = inlier, 0 = outlier). Optional arg absolute standardized residuals smaller than or equal to invnormal(.5 + #/200) will be class

weights generates a variable containing weights corresponding to the final RLS fit. **weights** is o

subset generates a 0/1 variable identifying the H-subset. The variable will be 1 for observation after **robreg lts**, **robreg lqs**, and **robreg lms**.

scores calculates equation-level scores. The scores can be used together with the information st scores generated after **robreg ls** deviate from the scores obtained by **predict** after **regress** b

ifs calculates coefficient-level influence functions. The influence functions are defined in a w
standard error of the coefficient. **ifs** is not allowed after **robreg lts**, **robreg lqs**, or **robre**

rifs calculates recentered influence functions. The recentered influence functions are defined i
provides an estimate of the standard error of the coefficient. **rifs** is not allowed after **rob**

Examples

Comparison of different estimators of the same model:

```
. sysuse auto, clear
. robreg ls price mpg weight headroom foreign
. robreg q price mpg weight headroom foreign
. robreg m price mpg weight headroom foreign
. robreg s price mpg weight headroom foreign
. robreg mm price mpg weight headroom foreign
```

We see, for example, that the effect of **headroom** has a p-value of 0.03 in the least-squares mode
relevant outliers.

Refitting MM with different efficiencies (without re-estimating the S):

```
. sysuse auto, clear
. robreg s price mpg weight headroom foreign
. robreg mm, efficiency(75)
. robreg mm, efficiency(80)
. robreg mm, efficiency(85)
. robreg mm, efficiency(90)
. robreg mm, efficiency(95)
```

Performing Hausman tests between models: The following example illustrates how one could do an i

```
. sysuse auto, clear
. robreg q price mpg weight headroom foreign, quantile(0.25)
. estimates store q25
. robreg q price mpg weight headroom foreign, quantile(0.75)
. estimates store q75
. robreg hausman q75 q25, constant
```

Using **predict** to flag outliers:

```
. sysuse auto, clear
. robreg mm price mpg weight headroom foreign
. predict outlier, outlier
. two scatter price weight if outlier==0 || scatter price weight if outlier==1
```

Using **predict** to generate recentered influence functions (the difference in standard errors is b

```
. sysuse auto, clear
. robreg q price mpg weight headroom foreign
. predict RIF*, rifs
. mean RIF*
```

Stored results

Depending on estimator and options, **robreg** saves a selection of the following results in **e()**.

Scalars

e(N)	number of observations
e(N_clust)	number of clusters (if **vce(cluster)**)
e(k_eq)	number of equations in **e(b)**
e(k_eform)	number of equations to be affected by **eform** option
e(k_omit)	number of omitted coefficients in **e(b)**
e(scale)	estimate of residual scale
e(scale0)	residual scale of empty model (unless **nor2**)
e(s0)	raw scale (**robreg lts/lqs/lms**)
e(s0_0)	raw scale of empty model (**robreg lts/lqs/lms**, unless **nor2**)
e(r2)	R-squared (**robreg ls**)
e(r2_p)	pseudo R-squared (all but **robreg ls**)
e(r2_w)	R-squared from final WLS fit (**robreg m/s/mm**)

e(k)	tuning constant (**robreg m/s/mm**)
e(m_k)	tuning constant of residualizing M (**robreg s/mm**, if **m()**)
e(kS)	tuning constant of S estimate (**robreg mm**)
e(bp)	breakdown point (**robreg s/mm/lts/lqs**)
e(efficiency)	gaussian efficiency (**robreg m/s/mm**)
e(m_efficiency)	gaussian efficiency of residualizing M (**robreg s/mm**, if **m()**)
e(effS)	gaussian efficiency of S estimate (**robreg mm**)
e(delta)	consistency parameter for scale estimation (**robreg s/mm**)
e(h)	relative size of H-subset (**robreg lts/lqs/lms**)
e(q_h)	h-quantile of squared residuals (**robreg lts/lqs/lms**)
e(crit)	value of optimization criterion (**robreg lts/lqs/lms**)
e(nsamp)	number of subsamples (**robreg s/mm/lts/lqs/lms**)
e(rsteps)	number of refinement steps (**robreg s/mm**)
e(csteps)	number of concentration steps (**robreg lts**)
e(nkeep)	number of candidates kept for final refinement (**robreg s/mm/lts**)
e(sum_adev)	sum of absolute deviations (**robreg q**)
e(sum_rdev)	sum of absolute deviations of empty model (**robreg q**, unless **nor2**)
e(q)	quantile requested (**robreg q**)
e(bwidth)	bandwidth (**robreg q**)
e(kbwidth)	kernel bandwidth (**robreg q**, unless **fitted**)
e(iterations)	number of iterations (**robreg q/m/mm**)
e(converged)	1 if converged, 0 else (**robreg q/m/mm**)
e(hausman_chi2)	chi-squared statistic of Hausman test (**robreg s/mm**)
e(hausman_F)	F statistic of Hausman test (**robreg s/mm**, if **ftest**)
e(hausman_p)	p value of Hausman test (**robreg s/mm**)
e(N_g)	number of groups (if relevant)
e(g_avg)	average group size (if relevant)
e(g_min)	minimum group size (if relevant)
e(g_max)	maximum group size (if relevant)
e(corr)	correlation between fixed effects and Xb (if relevant)
e(df_m)	model degrees of freedom
e(df_r)	residual degrees of freedom
e(chi2)	chi-squared statistic of model test (unless **nose**)
e(F)	F statistic of model test (if **ftest**, unless **nose**)
e(p)	p value of model test (unless **nose**)
e(rank)	rank of **e(V)** (unless **nose**)
e(level)	confidence level (unless **nose**)

Macros

e(cmd)	**robreg**
e(subcmd)	**ls**, **q**, **m**, **s**, **mm**, **lts**, **lqs**, or **lms**
e(predict)	**robreg predict**
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(indepvars)	names of independent variables
e(m)	variable names from **m()** option
e(noconstant)	**noconstant** or empty
e(ivar)	variable name from **ivar()** or empty
e(absorb)	variable name from **absorb()** or empty
e(ugenerate)	variable name from **ugenerate()** or empty
e(nor2)	**nor2** or empty
e(noquad)	**noquad** or empty
e(denmethod)	**kernel** or **fitted** (**robreg q**)
e(denmethod)	**gaussian** (**robreg q**)
e(bofinger)	**bofinger** or empty (**robreg q**)
e(vce)	*vcetype* specified in **vce()**
e(vcetype)	title used to label Std. Err.
e(clustvar)	name of cluster variable
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(properties)	**b** or **b V**

Matrices

e(b)	estimates
e(V)	variance-covariance matrix of estimates (unless **nose**)
e(omit)	vector identifying omitted coefficients
e(V_modelbased)	inverse of moment condition derivative matrix (unless **nose**)
e(u)	fixed-effects lookup table (if appropriate)
e(b0)	empty model fit (unless **nor2**)
e(b_init)	starting values (**robreg q/m**)
e(b_lo)	lower auxiliary fit (**robreg q**, if **fitted**)
e(b_up)	upper auxiliary fit (**robreg q**, if **fitted**)
e(IFoffset)	influence function offsets (**robreg q**)

Functions

e(sample)	estimation sample

If **vce()** is **bootstrap** or **jackknife**, additional results are stored in **e()**; see **bootstrap** and **jack**

robreg hausman saves the following results in **r()**. If option **post** is specified, the results are

Scalars

r(N)	number of observations
r(N_clust)	number of clusters (if **vce(cluster)**)
r(df_m)	test constraints degrees of freedom
r(df_r)	residual degrees of freedom
r(chi2)	chi-squared statistic of Hausman test
r(F)	F statistic of Hausman test (if **ftest**)
r(p)	p value of Hausman test

Macros

r(cmd)	**robreg**
r(subcmd)	**hausman**
r(wtype)	weight type
r(wexp)	weight expression
r(vce)	*vcetype* specified in **vce()**
r(vcetype)	title used to label Std. Err.
r(clustvar)	name of cluster variable
r(title)	title in estimation output

Matrices

r(b)	coefficient differences
r(V)	variance-covariance matrix

References

Croux, C., G. Dhaene, D. Hoorelbeke. 2003. Robust Standard Errors for Robust Estimators. Discuss
Hendricks, W., R. Koenker. 1992. Hierarchical Spline Models for Conditional Quantiles and the De
Huber, P.J. 1973. Robust Regression: Asymptotics, Conjectures and Monte Carlo. The Annals of Sta
Koenker, R. 2005. Quantile Regression. Cambridge: Cambridge University Press.
Koller, M. 2012. Nonsingular subsampling for S-estimators with categorical predictors. arXiv:120
Maronna, R.A., V.J. Yohai. 2000. Robust regression with both continuous and categorical predicto
Maronna, R.A., D.R. Martin, V.J. Yohai. 2006. Robust Statistics. Theory and Methods. Chichester:
Portnoy, S., R. Koenker. 1997. The Gaussian hare and the Laplacian tortoise: computability of sq
Powell, J.L. 1991. Estimation of monotonic regression models under quantile restrictions. Pp. 35
    Cambridge: Cambridge University Press.
Salibian-Barrera, M., V.J. Yohai. 2006. A Fast Algorithm for S-Regression Estimates. Journal of
Rousseeuw, P.J. 1984. Least Median of Squares Regression. Journal of the American Statistical As
Rousseeuw, P.J., M. Hubert. 1997. Recent developments in PROGRESS. Pp. 201-214 in: Y. Dodge (ed.
Rousseeuw, P.J., A.M. Leroy. 1987. Robust Regression and Outlier Detection. New York: Wiley.
Rousseeuw, P.J., K. van Driessen. 2002. Computing LTS regression for large data sets. Estadistic
Rousseeuw, P., V. Yohai. 1984. Robust Regression by Means of S-Estimators. Pp. 256-272 in: J. Fr
    Springer.
Yohai, V.J. 1987. High Breakdown-Point and High Efficiency Robust Estimates for Regression. The

Author

Ben Jann, University of Bern, ben.jann@unibe.ch

Thanks for citing this software as follows:

Jann, B. (2021). robreg: Stata module providing robust regression estimators. Available from

Also see

Official Stata: **regress**, **rreg**, **qreg**
SSC Archive: **xtrobreg**, **robstat**, **dstat**, **robbox**, **robmv**, **moremata**