Title

stata.com

areg — Linear regression with a large dummy-variable set

Description

`areg` fits a linear regression absorbing one categorical factor. `areg` is designed for datasets with many groups, but not a number of groups that increases with the sample size. See the `xtreg, fe` command in [XT] **xtreg** for an estimator that handles the case in which the number of groups increases with the sample size.

Quick start

Linear regression of `y` on `x`, absorbing an indicator variable for each level of `cvar`

```
areg y x, absorb(cvar)
```

As above, but add categorical variable `a`

```
areg y x i.a, absorb(cvar)
```

With cluster–robust standard errors

```
areg y x i.a, absorb(cvar) vce(cluster cvar2)
```

Using `svyset` data

```
svy: areg y x i.a, absorb(cvar)
```

Menu

Statistics > Linear models and related > Other > Linear regression absorbing one cat. variable

Syntax

```
areg depvar [indepvars] [if] [in] [weight] , absorb(varname) [options]
```

options	Description
Model	
* absorb(*varname*)	categorical variable to be absorbed
SE/Robust	
vce(*vcetype*)	*vcetype* may be ols, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

*absorb(*varname*) is required.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bootstrap, by, fp, jackknife, mi estimate, rolling, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] **mi estimate**.

Weights are not allowed with the bootstrap prefix; see [R] **bootstrap**.

aweights are not allowed with the jackknife prefix; see [R] **jackknife**.

aweights, fweights, and pweights are allowed; see [U] **11.1.6 weight**.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

absorb(*varname*) specifies the categorical variable, which is to be included in the regression as if it were specified by dummy variables. absorb() is required.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (ols), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] ***vce_option***.

vce(ols), the default, uses the standard variance estimator for ordinary least-squares regression.

Exercise caution when using the vce(cluster *clustvar*) option with areg. The effective number of degrees of freedom for the robust variance estimator is $n_g - 1$, where n_g is the number of clusters. Thus the number of levels of the absorb() variable should not exceed the number of clusters.

Reporting

`level(#)`; see [R] **estimation options**.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] **estimation options**.

The following option is available with `areg` but is not shown in the dialog box:

`coeflegend`; see [R] **estimation options**.

Remarks and examples

stata.com

Suppose that you have a regression model that includes among the explanatory variables a large number, k, of mutually exclusive and exhaustive dummies:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{d}_1\gamma_1 + \mathbf{d}_2\gamma_2 + \cdots + \mathbf{d}_k\gamma_k + \boldsymbol{\epsilon}$$

For instance, the dummy variables, $\mathbf{d}_i$, might indicate countries in the world or states of the United States. One solution would be to fit the model with `regress`, but this solution is possible only if k is small enough so that the total number of variables (the number of columns of $\mathbf{X}$ plus the number of $\mathbf{d}_i$'s plus one for $\mathbf{y}$) is sufficiently small—meaning less than `matsize` (see [R] **matsize**). For problems with more variables than the largest possible value of `matsize` (800 for Stata/IC and 11,000 for Stata/SE and Stata/MP), `regress` will not work. `areg` provides a way of obtaining estimates of $\boldsymbol{\beta}$—but not the γ_i's—in these cases. The effects of the dummy variables are said to be absorbed.

▷ Example 1

So that we can compare the results produced by `areg` with Stata's other regression commands, we will fit a model in which k is small. `areg`'s real use, however, is when k is large.

In our automobile data, we have a variable called `rep78` that is coded 1, 2, 3, 4, and 5, where 1 means poor and 5 means excellent. Let's assume that we wish to fit a regression of `mpg` on `weight`, `gear_ratio`, and `rep78` (parameterized as a set of dummies).

```
. use http://www.stata-press.com/data/r15/auto2
(1978 Automobile Data)
. regress mpg weight gear_ratio b5.rep78
```

Source	SS	df	MS	Number of obs	=	69
				F(6, 62)	=	21.31
Model	1575.97621	6	262.662702	Prob > F	=	0.0000
Residual	764.226686	62	12.3262369	R-squared	=	0.6734
				Adj R-squared	=	0.6418
Total	2340.2029	68	34.4147485	Root MSE	=	3.5109

mpg	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
weight	-.0051031	.0009206	-5.54	0.000	-.0069433	-.003263
gear_ratio	.901478	1.565552	0.58	0.567	-2.228015	4.030971
rep78						
Poor	-2.036937	2.740728	-0.74	0.460	-7.515574	3.4417
Fair	-2.419822	1.764338	-1.37	0.175	-5.946682	1.107039
Average	-2.557432	1.370912	-1.87	0.067	-5.297846	.1829814
Good	-2.788389	1.395259	-2.00	0.050	-5.577473	.0006939
_cons	36.23782	7.01057	5.17	0.000	22.22389	50.25175

To fit the `areg` equivalent, we type

```
. areg mpg weight gear_ratio, absorb(rep78)
```

Linear regression, absorbing indicators

Number of obs	=	69	
F(2, 62)	=	41.64	
Prob > F	=	0.0000	
R-squared	=	0.6734	
Adj R-squared	=	0.6418	
Root MSE	=	3.5109	

mpg	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
weight	-.0051031	.0009206	-5.54	0.000	-.0069433	-.003263
gear_ratio	.901478	1.565552	0.58	0.567	-2.228015	4.030971
_cons	34.05889	7.056383	4.83	0.000	19.95338	48.1644
rep78	F(4, 62) =	1.117	0.356	(5 categories)		

Both `regress` and `areg` display the same R^2 values, root mean squared error, and—for `weight` and `gear_ratio`—the same parameter estimates, standard errors, t statistics, significance levels, and confidence intervals. `areg`, however, does not report the coefficients for `rep78`, and, in fact, they are not even calculated. This computational trick makes the problem manageable when k is large. `areg` reports a test that the coefficients associated with `rep78` are jointly zero. Here this test has a significance level of 35.6%. This F test for `rep78` is the same that we would obtain after `regress` if we were to specify `test 1.rep78 2.rep78 3.rep78 4.rep78`; see [R] **test**.

The model F tests reported by `regress` and `areg` also differ. The `regress` command reports a test that all coefficients except that of the constant are equal to zero; thus, the dummies are included in this test. The `areg` output shows a test that all coefficients excluding the dummies and the constant are equal to zero. This is the same test that can be obtained after `regress` by typing `test weight gear_ratio`.

◁

❑ Technical note

areg is designed for datasets with many groups, but not a number that grows with the sample size. Consider two different samples from the U.S. population. In the first sample, we have 10,000 individuals and we want to include an indicator for each of the 50 states, whereas in the second sample we have 3 observations on each of 10,000 individuals and we want to include an indicator for each individual. areg was designed for datasets similar to the first sample in which we have a fixed number of groups, the 50 states. In the second sample, the number of groups, which is the number of individuals, grows as we include more individuals in the sample. For an estimator designed to handle the case in which the number of groups grows with the sample size, see the xtreg, fe command in [XT] **xtreg**.

Although the point estimates produced by areg and xtreg, fe are the same, the estimated VCEs differ when vce(cluster *clustvar*) is specified because the commands make different assumptions about whether the number of groups increases with the sample size.

❑

❑ Technical note

The intercept reported by areg deserves some explanation because, given k mutually exclusive and exhaustive dummies, it is arbitrary. areg identifies the model by choosing the intercept that makes the prediction calculated at the means of the independent variables equal to the mean of the dependent variable: $\overline{\mathbf{y}} = \overline{\mathbf{x}}\,\widehat{\boldsymbol{\beta}}$.

```
. predict yhat
(option xb assumed; fitted values)
. summarize mpg yhat if rep78 != .
```

Variable	Obs	Mean	Std. Dev.	Min	Max
mpg	69	21.28986	5.866408	12	41
yhat	69	21.28986	4.383224	11.58643	28.07367

We had to include if rep78 < . in our summarize command because we have missing values in our data. areg automatically dropped those missing values (as it should) in forming the estimates, but predict with the xb option will make predictions for cases with missing rep78 because it does not know that rep78 is really part of our model.

These predicted values do not include the absorbed effects (that is, the $\mathbf{d}_i\gamma_i$). For predicted values that include these effects, use the xbd option of predict (see [R] **areg postestimation**) or see [XT] **xtreg**.

❑

▷ Example 2

areg, vce(robust) is a Huberized version of areg; see [P] **_robust**. Just as areg is equivalent to using regress with dummies, areg, vce(robust) is equivalent to using regress, vce(robust) with dummies. You can use areg, vce(robust) when you expect heteroskedastic or nonnormal errors. areg, vce(robust), like ordinary regression, assumes that the observations are independent, unless the vce(cluster *clustvar*) option is specified. If the vce(cluster *clustvar*) option is specified, this independence assumption is relaxed and only the clusters identified by equal values of *clustvar* are assumed to be independent.

Assume that we were to collect data by randomly sampling 10,000 doctors (from 100 hospitals) and then sampling 10 patients of each doctor, yielding a total dataset of 100,000 patients in a cluster sample. If in some regression we wished to include effects of the hospitals to which the doctors belonged, we would want to include a dummy variable for each hospital, adding 100 variables to our model. `areg` could fit this model by

```
. areg depvar patient_vars, absorb(hospital) vce(cluster doctor)
```

◁

Stored results

`areg` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(tss)`	total sum of squares
`e(df_m)`	model degrees of freedom
`e(rss)`	residual sum of squares
`e(df_r)`	residual degrees of freedom
`e(r2)`	R-squared
`e(r2_a)`	adjusted R-squared
`e(df_a)`	degrees of freedom for absorbed effect
`e(rmse)`	root mean squared error
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(N_clust)`	number of clusters
`e(F)`	F statistic
`e(F_absorb)`	F statistic for absorbed effect (when `vce(robust)` is not specified)
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`areg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(absvar)`	name of `absorb` variable
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. Err.
`e(datasignature)`	the checksum
`e(datasignaturevars)`	variables used in calculation of checksum
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(footnote)`	program used to implement the footnote display
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

Methods and formulas

areg begins by recalculating *depvar* and *indepvars* to have mean 0 within the groups specified by absorb(). The overall mean of each variable is then added back in. The adjusted *depvar* is then regressed on the adjusted *indepvars* with regress, yielding the coefficient estimates. The degrees of freedom of the variance–covariance matrix of the coefficients is then adjusted to account for the absorbed variables—this calculation yields the same results (up to numerical roundoff error) as if the matrix had been calculated directly by the formulas given in [R] **regress**.

areg with vce(robust) or vce(cluster *clustvar*) works similarly, calling _robust after regress to produce the Huber/White/sandwich estimator of the variance or its clustered version. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. The model F test uses the robust variance estimates. There is, however, no simple computational means of obtaining a robust test of the absorbed dummies; thus this test is not displayed when the vce(robust) or vce(cluster *clustvar*) option is specified.

The number of groups specified in absorb() are included in the degrees of freedom used in the finite-sample adjustment of the cluster–robust VCE estimator. This statement is only valid if the number of groups is small relative to the sample size. (Technically, the number of groups must remain fixed as the sample size grows.) For an estimator that allows the number of groups to grow with the sample size, see the xtreg, fe command in [XT] **xtreg**.

References

Blackwell, J. L., III. 2005. Estimation and testing of fixed-effect panel-data systems. *Stata Journal* 5: 202–207.

McCaffrey, D. F., K. Mihaly, J. R. Lockwood, and T. R. Sass. 2012. A review of Stata commands for fixed-effects estimation in normal linear models. *Stata Journal* 12: 406–432.

Also see

[R] **areg postestimation** — Postestimation tools for areg

[R] **regress** — Linear regression

[MI] **estimation** — Estimation commands for use with mi estimate

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[U] **20 Estimation and postestimation commands**