Title

stata.com

predict treatment — predict for treatment statistics

Description

`predict` has options to predict potential-outcome means, treatment effects, and treatment effects on the treated after models fit using the `entreat()` or `extreat()` option. The `predict` options are described below.

For standard use of `predict`, see

[ERM] **eregress predict**

[ERM] **eintreg predict**

[ERM] **eprobit predict**

[ERM] **eoprobit predict**

For advanced use of `predict`, see

[ERM] **predict advanced**

Also see [ERM] **estat teffects** for reports of average treatment statistics.

Syntax

You previously fit a model by using the `entreat()` or `extreat()` option,

```
eregress   y    x1 ..., ... entreat(treated = ...) ...
eintreg  yl yu  x1 ..., ... entreat(treated = ...) ...
eprobit    y    x1 ..., ... entreat(treated = ...) ...
eoprobit   y    x1 ..., ... entreat(treated = ...) ...
eregress   y    x1 ..., ... extreat(treated) ...
eintreg  yl yu  x1 ..., ... extreat(treated) ...
eprobit    y    x1 ..., ... extreat(treated) ...
eoprobit   y    x1 ..., ... extreat(treated) ...
```

In these cases, `predict` has extra features. `predict`'s extra syntax for these features is

`predict` [*type*] *newvar* [*if*] [*in*] , *treatstatistic* [*treatmodifier* *oprobitmodifier*]

In some cases, more than one new variable needs to be specified:

`predict` [*type*] { *stub*`*` | *newvarlist* } [*if*] [*in*] , *treatstatistic* [*treatmodifier* *oprobitmodifier*]

treatstatistic	Description
pomean	potential-outcome mean (POM)
te	treatment effect (TE)
tet	treatment effect on the treated (TET)

treatmodifier	Description
tlevel(#)	treatment level for which *treatstatistic* is calculated

may be specified as a value recorded in variable treated, such as 1, 2, ... or such as 1, 5, ..., depending on the values recorded.

may also be specified as #1, #2, ..., meaning the first, second, ... values recorded in treated.

oprobitmodifier	Description
outlevel(#)	ordered outcome for which *treatstatistic* is calculated

When used after models fit with eoprobit, *treatstatistic* is calculated for the specified outcome, or for the first outcome if you do not specify otherwise.

outlevel(#) specifies the outcome for which statistics are to be calculated. # is specified in the same way as with tlevel(), but the meaning is different. In the case of outlevel(), you are specifying the outcome, not the treatment level.

Options

The options for the statistic to be calculated—pomean, te, and tet—are mutually exclusive. You calculate one treatment statistic per predict command.

pomean calculates the POMs for each treatment level. The POMs are the expected value of y that would have been observed if everyone was assigned to each of the treatment levels.

If there were two treatment levels (a control and a treatment), you would type

```
. predict pom1 pom2, pomean
```

If there were three levels, you would type

```
. predict pom1 pom2 pom3, pomean
```

pomean can alternatively be used with tlevel() to produce individual POMs:

```
. predict pom1, pomean tlevel(#1)
. predict pom2, pomean tlevel(#2)
```

If you have fit the model using eoprobit, the POMs calculated for the examples above would be for y's first outcome. You can change that. See *Predicting treatment effects after eoprobit* in *Remarks and examples* below.

te calculates the TEs for each treatment level. The TEs are the differences in the POMs. For instance, if there were two treatment levels—a control and a treatment—there would be one treatment effect and it would be pom2-pom1. If there were three levels, there would be two treatment effects, pom2-pom1 and pom3-pom1.

If there were two treatment levels—a control and a treatment—you would type

```
. predict te2, te
```

If there were three levels, you would type

```
. predict te2 te3, te
```

te can alternatively be used with tlevel() to produce individual TEs:

```
. predict te2, te tlevel(#2)
. predict te3, te tlevel(#3)
```

If you have fit the model using eoprobit, the TEs calculated for the examples above would be for y's first outcome. You can change that. See *Predicting treatment effects after eoprobit* in *Remarks and examples* below.

tet calculates the TETs. The TETs are the differences in the POMs conditioned on treatment level.

If there were two treatment levels—a control and a treatment—you would type

```
. predict tet2, tet
```

If there were three levels, you would type

```
. predict tet2 tet3, tet
```

tet can alternatively be used with tlevel() to produce individual TETs:

```
. predict tet2, tet tlevel(#2)
. predict tet3, tet tlevel(#3)
```

If you have fit the model using eoprobit, the TETs calculated for the examples above would be for y's first outcome. You can change that. See *Predicting treatment effects after eoprobit* in *Remarks and examples* below.

tlevel(#) is optionally used with pomean, te, or tet. Its use is illustrated above.

outlevel(#) is optionally used with pomean, te, or tet with models fit by eoprobit. See *Predicting treatment effects after eoprobit* in *Remarks and examples* below.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Predicting treatment effects after eregress and eintreg

eregress and eintreg concern models with a continuous outcome variable. In eregress models, $\mathtt{y}_i$ is observed. In eintreg models, y_i is not observed directly, but it is known that $\mathtt{yl}_i \leq y_i \leq \mathtt{yu}_i$.

Thus, the treatment statistics are expressed in the units of y. If y is blood pressure, the units are presumably mmHG. POMs are in mmHG. TEs and TETs are differences in blood pressure expressed in mmHG.

Predicting treatment effects after eprobit

eprobit concerns models with binary outcomes, and predictions are in terms of the probability of a positive outcome. Thus, POMs are probabilities. TEs and TETs are differences in probabilities.

Predicting treatment effects after eoprobit

`eprobit` concerns models with ordinal outcome variables, and predictions are in terms of the probabilities—the probability of each outcome.

Treatment statistics are calculated on the basis of probabilities of outcomes. Thus, POMs are probabilities. TEs and TETs are differences in probabilities.

We want probabilities and differences in probabilities, but you need to specify which probability. The probability for the first outcome? The second?

If you do not specify which and simply type

```
. predict pom1 pom2 pom3, pomean
```

then the POMs are calculated for the first outcome, what `eoprobit` calls `outlevel(#1)`. If you wanted to obtain the POMs for `outlevel(#2)`, you would type

```
. predict pom1 pom2 pom3, pomean outlevel(#2)
```

If you wanted them for `outlevel(#3)`, you would type

```
. predict pom1 pom2 pom3, pomean outlevel(#3)
```

The same logic applies to calculating TE and TET with the `te` and `tet` options. `outlevel(#1)` is used unless you specify otherwise.

Methods and formulas

See *Methods and formulas* in [ERM] **eintreg**, [ERM] **eoprobit**, [ERM] **eprobit**, and [ERM] **eregress**.

Also see

[ERM] **eintreg postestimation** — Postestimation tools for eintreg

[ERM] **eintreg predict** — predict after eintreg

[ERM] **eoprobit postestimation** — Postestimation tools for eoprobit

[ERM] **eoprobit predict** — predict after eoprobit

[ERM] **eprobit postestimation** — Postestimation tools for eprobit

[ERM] **eprobit predict** — predict after eprobit

[ERM] **eregress postestimation** — Postestimation tools for eregress

[ERM] **eregress predict** — predict after eregress