26 Overview of Stata estimation commands

Contents

26.1 Introduction

Estimation commands fit models such as linear regression and probit. Stata has many such commands, so it is easy to overlook a few. Some of these commands differ greatly from each other, others are gentle variations on a theme, and still others are equivalent to each other.

Estimation commands share features that this chapter will not discuss; see **[U] 20 Estimation and postestimation commands**. Especially see **[U] 20.21 Obtaining robust variance estimates**, which discusses an alternative calculation for the estimated variance matrix (and hence standard errors) that many of Stata's estimation commands provide, and **[U] 20.12 Performing hypothesis tests on the coefficients**.

Here, however, this chapter will put aside all of that—and all issues of syntax—and deal solely with matching commands to their statistical concepts. This chapter will not cross-reference specific commands. To find the details on a particular command, look up its name in the index.

26.2 Means, proportions, and related statistics

This group of estimation commands computes summary statistics rather than fitting regression models. However, being estimation commands, they share the features discussed in **[U] 20 Estimation and postestimation commands**, such as allowing the use of postestimation commands.

`mean`, `proportion`, `ratio`, and `total` provide estimates of population means, proportions, ratios, and totals, respectively. Each of these commands allows for obtaining separate estimates within subpopulations, groups defined by a separate categorical variable. In addition, `mean`, `proportion`, and `ratio` can report statistics adjusted by direct standardization.

`pwmean` provides another option for computing means of one variable for each level of one or more categorical variables. In addition, `pwmean` computes all pairwise differences in these means along with the corresponding tests and confidence intervals, which can optionally be adjusted for to account for multiple comparisons.

26.3 Linear regression with simple error structures

Consider models of the form

$$y_j = \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j$$

for a continuous y variable. In this category, estimation is restricted to when σ^2_ϵ is constant across observations j. The model is called the linear regression model, and the estimator is often called the (ordinary) least-squares (OLS) estimator.

`regress` is Stata's linear regression command. (`regress` produces the robust estimate of variance as well as the conventional estimate, and `regress` has a collection of commands that can be run after it to explore the nature of the fit.)

Also, the following commands will do linear regressions, as does `regress`, but offer special features:

1. `ivregress` fits models in which some of the regressors are endogenous, using either instrumental variables or generalized method of moments (GMM) estimators.
2. `areg` fits models $y_j = \mathbf{x}_j\beta + \mathbf{d}_j\boldsymbol{\gamma} + \epsilon_j$, where $\mathbf{d}_j$ is a mutually exclusive and exhaustive dummy variable set. `areg` obtains estimates of $\boldsymbol{\beta}$ (and associated statistics) without ever forming $\mathbf{d}_j$, meaning that it also does not report the estimated $\boldsymbol{\gamma}$. If your interest is in fitting fixed-effects models, Stata has a better command—`xtreg`—discussed in **[U] 26.20.1 Linear regression with**

panel data. Most users who find `areg` appealing will probably want to use `xtreg` because it provides more useful summary and test statistics. `areg` duplicates the output that `regress` would produce if you were to generate all the dummy variables. This means, for instance, that the reported R^2 includes the effect of $\boldsymbol{\gamma}$.

3. `boxcox` obtains maximum likelihood estimates of the coefficients and the Box–Cox transform parameters in a model of the form

$$y_i^{(\theta)} = \beta_0 + \beta_1 x_{i1}^{(\lambda)} + \beta_2 x_{i2}^{(\lambda)} + \cdots + \beta_k x_{ik}^{(\lambda)} + \gamma_1 z_{i1} + \gamma_2 z_{i2} + \cdots + \gamma_l z_{il} + \epsilon_i$$

where $\epsilon \sim N(0, \sigma^2)$. Here the y is subject to a Box–Cox transform with parameter θ. Each of the $x_1, x_2, \ldots, x_k$ is transformed by a Box–Cox transform with parameter λ. The $z_1, z_2, \ldots, z_l$ are independent variables that are not transformed. In addition to the general form specified above, `boxcox` can fit three other versions of this model defined by the restrictions $\lambda = \theta$, $\lambda = 1$, and $\theta = 1$.

4. `tobit` allows estimation of linear regression models when y_i has been subject to left-censoring, right-censoring, or both. Say that y_i is not observed if $y_i < 1{,}000$, but for those observations, it is known that $y_i < 1{,}000$. `tobit` fits such models.

   `ivtobit` does the same but allows for endogenous covariates.

5. `intreg` (interval regression) is a generalization of `tobit`. In addition to allowing open-ended intervals, `intreg` allows closed intervals. Rather than observing y_j, `intreg` assumes that y_{0j} and y_{1j} are observed, where $y_{0j} \leq y_j \leq y_{1j}$. Survey data might report that a subject's monthly income was in the range \$1,500–\$2,500. `intreg` allows such data to be used to fit a regression model. `intreg` allows $y_{0j} = y_{1j}$ and so can reproduce results reported by `regress`. `intreg` allows y_{0j} to be $-\infty$ and y_{1j} to be $+\infty$ and so can reproduce results reported by `tobit`.

6. `truncreg` fits the regression model when the sample is drawn from a restricted part of the population and so is similar to `tobit`, except that here the independent variables are not observed. Under the normality assumption for the whole population, the error terms in the truncated regression model have a truncated-normal distribution.

7. `churdle` allows estimation of linear or exponential hurdle models when y_i is subject to a lower-boundary $\ell\ell$, an upper-boundary $u\ell$, or both. The dependent variable is a mixture of discrete observations at the boundary points and continuous observations over the interior. Both boundary and interior observations on y_i are actual realizations. In contrast, censored-data models such as `tobit` and `intreg` treat interior observations as actual realizations and treat boundary observations as indicating only that the actual realizations lie beyond the boundary. Hurdle models use one model to determine whether an observation is on the boundary or in the interior and another model for the values in the interior.

8. `cnsreg` allows you to place linear constraints on the coefficients.

9. `eivreg` adjusts estimates for errors in variables.

10. `nl` provides the nonlinear least-squares estimator of $y_j = f(\mathbf{x}_j, \boldsymbol{\beta}) + \epsilon_j$.

11. `rreg` fits robust regression models, which are not to be confused with regression with robust standard errors. Robust standard errors are discussed in **[U] 20.21 Obtaining robust variance estimates**. Robust regression concerns point estimates more than it does standard errors, and it implements a data-dependent method for downweighting outliers.

12. `qreg` produces quantile regression estimates, a variation that is not linear regression at all but is an estimator of $y_j = \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j$. In the basic form of this model, sometimes called median regression, $\mathbf{x}_j\boldsymbol{\beta}$ measures not the predicted mean of y_j conditional on $\mathbf{x}_j$ but its median. As such, `qreg` is of most interest when ϵ_j does not have constant variance. `qreg` allows you to

specify the quantile, so you can produce linear estimates for the predicted 1st, 2nd, ..., 99th percentile.

Another command, bsqreg, is identical to qreg but presents bootstrap standard errors.

The sqreg command estimates multiple quantiles simultaneously; standard errors are obtained via the bootstrap.

The iqreg command estimates the difference between two quantiles; standard errors are obtained via the bootstrap.

13. vwls (variance-weighted least squares) produces estimates of $y_j = \mathbf{x}_j\boldsymbol{\beta}+\epsilon_j$, where the variance of ϵ_j is calculated from group data or is known a priori. vwls is therefore of most interest to categorical-data analysts and physical scientists.

26.4 Structural equation modeling (SEM)

SEM stands for "structural equation modeling". The sem and gsem commands fit SEM.

sem fits standard linear SEMs. gsem fits what we call generalized SEMs, generalized to allow for generalized linear responses and multilevel modeling.

Generalized linear means, among other types of responses, binary responses such as probit and logit, count responses such as Poisson and negative binomial, categorical responses such as multinomial logit, ordered responses such as ordered probit and ordered logit, censored responses such as tobit, and survival responses such as exponential and Weibull. Generalized linear includes linear responses.

Multilevel modeling allows for nested effects such as patient within doctor and patients within doctor within hospital and crossed effects such as occupation and industry.

Let's start with sem. sem can fit models ranging from linear regression to measurement models to simultaneous equations, including confirmatory factor analysis (CFA) models, correlated uniqueness models, latent growth models, and multiple indicators and multiple causes (MIMIC) models. You can obtain standardized or unstandardized results, direct and indirect effects, goodness-of-fit statistics, modification indices, scores tests, Wald tests, linear and nonlinear tests of estimated parameters, and linear and nonlinear combinations of estimated parameters with confidence intervals. You can perform estimation across groups with easy model specification and easy-to-use tests for group invariance. All of this may be done using raw or summary statistics data. In addition, sem optionally can use full information maximum-likelihood (FIML) estimation to handle observations containing missing values.

gsem extends the types of models that can be fit. Responses may be continuous, ordinal, count, categorical, or survival time, and gsem allows for multilevel modeling. Latent variables can be included at any level. This allows for fitting models with random intercepts and random slopes. These random effects may be nested or crossed.

There is considerable overlap in the capabilities of sem and gsem. Whenever there is overlap, sem is faster and sometimes easier to use.

The generalized response variables allowed by gsem permit fitting measurement models with different types of responses, latent growth models with different types of responses, and so on.

gsem can also fit item response theory (IRT) models, multilevel CFA models, multilevel mixed-effects models, and multilevel structural equation models.

Where appropriate, results can be reported in exponentiated form to provide odds ratios, incidence-rate ratios, and relative-risk ratios. You can also obtain predictions, likelihood-ratio tests, Wald tests, predictive margins, contrasts, and pairwise comparisons.

Whether fitting a model with sem or gsem, you can specify your model by typing the command or by using the SEM Builder to draw path diagrams.

For those of you unfamiliar with SEM, it is worth your time to learn about it if you ever fit linear regressions, logistic regressions, ordered logit regressions, ordered probit regressions, Poisson regressions, seemingly unrelated regressions, multivariate regressions, simultaneous systems, measurement error models, selection models, endogenous treatment-effects models, tobit models, survival models, fractional response models, and multilevel mixed-effects models.

You may also want to learn about SEM if you are interested in GMM. sem and gsem fit many of the same models by maximum likelihood and quasimaximum likelihood that you can fit by GMM.

sem and gsem can be used to fit many models that can be fit by other Stata commands. The advantage of using sem and gsem is in the extensions that they can provide. They allow for introduction of latent variables to account for measurement error, simultaneous equations with different types of responses, multilevel versions of popular models such as selection models, and more.

See the [SEM] *Stata Structural Equation Modeling Reference Manual*; in particular, see [SEM] **intro 5**.

26.5 ANOVA, ANCOVA, MANOVA, and MANCOVA

ANOVA and ANCOVA are related to linear regression, but we classify them separately. The related Stata commands are anova, oneway, and loneway. The manova command provides MANOVA and MANCOVA (multivariate ANOVA and ANCOVA).

anova fits ANOVA and ANCOVA models, one-way and up—including two-way factorial, three-way factorial, etc.—and fits nested and mixed-design models and repeated-measures models.

oneway fits one-way ANOVA models. It is produces estimates more quickly than anova, although anova is so fast that this probably does not matter. The important difference is that oneway can report multiple-comparison tests.

loneway is an alternative to oneway. The results are numerically the same, but loneway can deal with more levels, limited only by dataset size (oneway is limited to 376 levels and anova to 798, but for anova to reach 798 requires a lot of memory). loneway also reports some additional statistics, such as the intraclass correlation.

manova fits MANOVA and MANCOVA models, one-way and up—including two-way factorial, three-way factorial, etc.—and it fits nested and mixed-design models.

26.6 Generalized linear models

The generalized linear model is

$$g\{E(y_j)\} = \mathbf{x}_j\boldsymbol{\beta}, \qquad y_j \sim F$$

where $g()$ is called the link function and F is a member of the exponential family, both of which you specify before estimation. glm fits this model.

The GLM framework encompasses a surprising array of models known by other names, including linear regression, Poisson regression, exponential regression, and others. Stata provides dedicated estimation commands for many of these. For instance, Stata has regress for linear regression, poisson for Poisson regression, and streg for exponential regression, and that is not all the overlap.

glm by default uses maximum likelihood estimation and alternatively estimates via iterated reweighted least squares (IRLS) when the irls option is specified. For each family, F, there is a corresponding link function, $g()$, called the canonical link, for which IRLS estimation produces results identical to maximum likelihood estimation. You can, however, match families and link functions as you wish, and when you match a family to a link function other than the canonical link, you obtain a different but valid estimator of the standard errors of the regression coefficients. The estimator you obtain is asymptotically equivalent to the maximum likelihood estimator, which, in small samples, produces slightly different results.

For example, the canonical link for the binomial family is logit. glm, irls with that combination produces results identical to the maximum-likelihood logit (and logistic) command. The binomial family with the probit link produces the probit model, but probit is not the canonical link here. Hence, glm, irls produces standard-error estimates that differ slightly from those produced by Stata's maximum-likelihood probit command.

Many researchers feel that the maximum-likelihood standard errors are preferable to IRLS estimates (when they are not identical), but they would have a difficult time justifying that feeling. Maximum likelihood probit is an estimator with (solely) asymptotic properties; glm, irls with the binomial family and probit link is an estimator with (solely) asymptotic properties, and in finite samples, the standard errors differ a little.

Still, we recommend that you use Stata's dedicated estimators whenever possible. IRLS—the theory—and glm, irls—the command—are all encompassing in their generality, meaning that they rarely use the right jargon or provide things in the way you wish they would. The narrower commands, such as logit, probit, and poisson, focus on the issue at hand and are invariably more convenient.

glm is useful when you want to match a family to a link function that is not provided elsewhere.

glm also offers several estimators of the variance–covariance matrix that are consistent, even when the errors are heteroskedastic or autocorrelated. Another advantage of a glm version of a model over a model-specific version is that many of these VCE estimators are available only for the glm implementation. You can also obtain the ML–based estimates of the VCE from glm.

26.7 Binary-outcome qualitative dependent-variable models

There are many ways to write these models, one of which is

$$\Pr(y_j \neq 0) = F(\mathbf{x}_j\boldsymbol{\beta})$$

where F is some cumulative distribution. Two popular choices for $F()$ are the normal and logistic, and the models are called the probit and logit (or logistic regression) models. A third is the complementary log–log function; maximum likelihood estimates are obtained by Stata's cloglog command.

The two parent commands for the maximum likelihood estimator of probit and logit are probit and logit, although logit has a sibling, logistic, that provides the same estimates but displays results in a slightly different way. There is also an exact logistic estimator; see **[U] 26.14 Exact estimators**.

Do not read anything into the names logit and logistic. Logit and logistic have two interchanged definitions in two scientific camps. In the medical sciences, logit means the minimum χ^2 estimator, and logistic means maximum likelihood. In the social sciences, it is the other way around. From our experience, it appears that neither reads the other's literature, because both assert that logit means one thing and logistic the other. Our solution is to provide both logit and logistic, which do the same thing, so that each camp can latch on to the maximum likelihood command under the name each expects.

There are two slight differences between `logit` and `logistic`. `logit` reports estimates in the coefficient metric, whereas `logistic` reports exponentiated coefficients—odds ratios. This is in accordance with the expectations of each camp and makes no substantial difference. The other difference is that `logistic` has a family of post`logistic` commands that you can run to explore the nature of the fit. Actually, that is not exactly true because all the commands for use after `logistic` can also be used after `logit`.

If you have not already selected `logit` or `logistic` as your favorite, we recommend that you try `logistic`. Logistic regression (logit) models are more easily interpreted in the odds-ratio metric.

`binreg` can be used to model either individual-level or grouped data in an application of the generalized linear model. The family is assumed to be binomial, and each link provides a distinct parameter interpretation. Also, `binreg` offers several options for setting the link function according to the desired biostatistical interpretation. The available links and interpretation options are the following:

Option	Implied link	Parameter
`or`	`logit`	Odds ratios = $\exp(\beta)$
`rr`	`log`	Risk ratios = $\exp(\beta)$
`hr`	`log complement`	Health ratios = $\exp(\beta)$
`rd`	`identity`	Risk differences = β

Related to logit, the skewed logit estimator scobit adds a power to the logit link function and is estimated by Stata's `scobit` command.

Turning to probit, you have two choices: `probit` and `ivprobit`. `probit` fits a maximum-likelihood probit model. `ivprobit` fits a probit model where one or more of the covariates are endogenously determined.

Continuing with probit: `hetprobit` fits heteroskedastic probit models. In these models, the variance of the error term is parameterized.

`heckprobit` fits probit models with sample selection.

Also, Stata's `biprobit` command fits bivariate probit models, meaning models with two correlated outcomes. `biprobit` also fits partial-observability models in which only the outcomes $(0, 0)$ and $(1, 1)$ are observed.

26.8 ROC analysis

ROC stands for "receiver operating characteristics". ROC deals with specificity and sensitivity, the number of false positives and undetected true positives of a diagnostic test. The term "ROC" dates back to the early days of radar when there was a knob on the radar receiver labeled "ROC". If you turned the knob one way, the receiver became more sensitive, which meant it was more likely to show airplanes that really were there and, simultaneously, more likely to show returns where there were no airplanes (false positives). If you turned the knob the other way, you suppressed many of the false positives, but unfortunately, you also suppressed the weak returns from real airplanes (undetected positives). These days, in the statistical applications we imagine, one does not turn a knob but instead chooses a value of the diagnostic test, above which is declared to be a positive and below which, a negative.

ROC analysis is applied to binary outcomes such as those appropriate for probit or logistic regression. After fitting a model, one can obtain predicted probabilities of a positive outcome. One chooses a value, above which the predicted probability is declared a positive and below which, a negative.

ROC analysis is about modeling the tradeoff of sensitivity and specificity as the threshold value is chosen.

Stata's suite for ROC analysis consists of six commands: `roctab`, `roccomp`, `rocfit`, `rocgold`, `rocreg`, and `rocregplot`.

`roctab` provides nonparametric estimation of the ROC curve and produces Bamber and Hanley confidence intervals for the area under the curve.

`roccomp` provides tests of equality of ROC areas. It can estimate nonparametric and parametric binormal ROC curves.

`rocfit` fits maximum likelihood models for a single classifier, an indicator of the latent binormal variable for the true status.

`rocgold` performs tests of equality of ROC area against a "gold standard" ROC curve and can adjust significance levels for multiple tests across classifiers via Šidák's method.

`rocreg` performs ROC regression; it can adjust both sensitivity and specificity for prognostic factors such as age and gender; it is by far the most general of all the ROC commands.

`rocregplot` graphs ROC curves as modeled by `rocreg`. ROC curves can be drawn across covariate values, across classifiers, and across both.

See [R] **roc**.

26.9 Conditional logistic regression

`clogit` is Stata's conditional logistic regression estimator. In this model, observations are assumed to be partitioned into groups, and a predetermined number of events occur in each group. The model measures the risk of the event according to the observation's covariates, $\mathbf{x}_j$. The model is used in matched case–control studies (`clogit` allows $1:1$, $1:k$, and $m:k$ matching) and is used in natural experiments whenever observations can be grouped into pools in which a fixed number of events occur. `clogit` is also used to fix logistic regression with fixed group effects.

26.10 Fractional-outcome dependent-variable models

Fractional response data occur when the outcome of interest is measured as a fraction, proportion, or rate. Two widely used methods for modeling these outcomes are beta regression and fractional regression.

`betareg` can be used to estimate the parameters of a beta-regression model for fractional responses that are strictly greater than zero and less than one.

`fracreg` can be used to estimate the parameters of a fractional logistic model, a fractional probit model, or a fractional heteroskedastic probit model for fractional responses that are greater than or equal to zero and less than or equal to one.

Both commands use quasimaximum likelihood estimation. When the dependent variable is between zero and one, `betareg` provides more flexibility than `fracreg` in the distribution of the conditional mean of the dependent variable.

26.11 Multiple-outcome qualitative dependent-variable models

For more than two outcomes, Stata provides ordered logit, ordered probit, rank-ordered logit, multinomial logistic regression, multinomial probit regression, McFadden's choice model (conditional fixed-effects logistic regression), and nested logistic regression.

`oprobit` and `ologit` provide maximum-likelihood ordered probit and logit. These are generalizations of probit and logit models known as the proportional odds model and are used when the outcomes have a natural ordering from low to high. The idea is that there is an unmeasured $z_j = \mathbf{x}_j\boldsymbol{\beta}$, and the probability that the kth outcome is observed is $\Pr(c_{k-1} < z_j < c_k)$, where $c_0 = -\infty$, $c_k = +\infty$, and $c_1, \ldots, c_{k-1}$ along with $\boldsymbol{\beta}$ are estimated from the data.

`heckoprobit` fits ordered probit models with sample selection.

`rologit` fits the rank-ordered logit model for rankings. This model is also known as the Plackett–Luce model, the exploded logit model, and choice-based conjoint analysis.

`asroprobit` fits the probit model for rankings, a more flexible estimator than `rologit` because `asroprobit` allows covariances among the rankings. `asroprobit` is also similar to `asmprobit` (below), which is used for outcomes that have no natural ordering. The `as` in the name signifies that `asroprobit` also allows alternative-specific regressors—variables that have different coefficients for each alternative.

`slogit` fits the stereotype logit model for data that are not truly ordered, as data are for `ologit`, but for which you are not sure that they are unordered, in which case `mlogit` would be appropriate.

`mlogit` fits maximum-likelihood multinomial logistic models, also known as polytomous logistic regression. It is intended for use when the outcomes have no natural ordering and you know only the characteristics of the outcome chosen (and, perhaps, the chooser).

`asclogit` fits McFadden's choice model, also known as conditional logistic regression. In the context denoted by the name McFadden's choice model, the model is used when the outcomes have no natural ordering, just as in multinomial logistic regression, but the characteristics of the outcomes chosen and not chosen are known (along with, perhaps, the characteristics of the chooser).

In the context denoted by the name conditional logistic regression—mentioned above—subjects are members of pools, and one or more are chosen, typically to be infected by some disease or to have some other unfortunate event befall them. Thus the characteristics of the chosen and not chosen are known, and the issue of the characteristics of the chooser never arises. Either way, it is the same model.

In their choice-model interpretations, `mlogit` and `clogit` assume that the odds ratios are independent of other alternatives, known as the independence of irrelevant alternatives (IIA) assumption. This assumption is often rejected by the data, and the nested logit model relaxes this assumption. `nlogit` is also popular for fitting the random utility choice model.

`asmprobit` is for use with outcomes that have no natural ordering and with regressors that are alternative specific. It is weakly related to `mlogit`. Unlike `mlogit`, `asmprobit` does not assume the IIA.

`mprobit` is also for use with outcomes that have no natural ordering but with models that do not have alternative-specific regressors.

26.12 Item response theory

Item response theory (IRT) is used in the design, analysis, scoring, and comparison of tests and similar instruments whose purpose is to measure a latent trait. Latent traits cannot be measured directly because they are unobservable, but they can be quantified with an instrument. An instrument is simply a collection of items designed to measure a person's level of the latent trait. For example, a researcher interested in measuring mathematical ability (latent trait) may design a test (instrument) consisting of 100 questions (items).

When designing the instrument or analyzing data from the instrument, the researcher is interested in how each individual item relates to the trait and how the group of items as a whole relates to this trait. IRT models allow us to study these relationships.

Stata provides a suite of IRT estimation commands to fit a variety of models for binary responses and categorical responses. Models can also be combined. The available commands are the following:

Command	Description	Response
`irt 1pl`	One-parameter logistic model	binary
`irt 2pl`	Two-parameter logistic model	binary
`irt 3pl`	Three-parameter logistic model	binary
`irt grm`	Graded response model	categorical
`irt nrm`	Nominal response model	categorical
`irt pcm`	Partial credit model	categorical
`irt rsm`	Rating scale model	categorical
`irt hybrid`	Hybrid IRT model	combination

A major concept in IRT is the item characteristic curve (ICC). The ICC maps the relationship between the latent trait and the probability that a person "succeeds" on a given item (individual test question). `irtgraph icc` can be used to plot the ICCs for items after any of the models above.

`irtgraph tcc` is used to plot the test characteristic curve (TCC), which shows the relationship between the expected score on the whole test and the latent trait. Plots of the item information and test information can be obtained with `irtgraph iif` and `irtgraph tif`.

See [IRT] **irt** for more information.

26.13 Count dependent-variable models

These models concern dependent variables that count the occurrences of an event. In this category, we include Poisson and negative binomial regression. For the Poisson model,

$$E(\text{count}) = E_j \exp(\mathbf{x}_j\boldsymbol{\beta})$$

where E_j is the exposure time. `poisson` fits this model; see [R] **poisson**. There is also an exact Poisson estimator; see [U] **26.14 Exact estimators**. `ivpoisson` fits a Poisson model where one or more of the covariates are endogenously determined. It can also be used for modeling nonnegative continuous outcomes instead of counts. See [R] **ivpoisson**.

Negative binomial regression refers to estimating with data that are a mixture of Poisson counts. One derivation of the negative binomial model is that individual units follow a Poisson regression model but that there is an omitted variable that follows a gamma distribution with parameter α. Negative binomial regression estimates $\boldsymbol{\beta}$ and α. `nbreg` fits such models. A variation on this, unique to Stata, allows you to model α. `gnbreg` fits those models. See [R] **nbreg**.

Sometimes, the value of the outcome variable is not observed when it falls outside a known range, and it is observed inside that range. This limitation comes in two flavors—censoring and truncation. It is called censoring when we have an observation for the outcome, but know only that the value of the outcome is outside the range. It is called truncation when we do not even have an observation when the value of the outcome is outside the range. The `cpoisson` command can be used to fit models for censored count data. Commands `tpoisson` and `tnbreg` can be used to fit models for truncated count data.

Zero inflation refers to count models in which the number of zero counts is more than would be expected in the regular model. The excess zeros are explained by a preliminary probit or logit process. If the preliminary process produces a positive outcome, the usual counting process occurs, and otherwise, the count is zero. Thus, whenever the preliminary process produces a negative outcome, excess zeros are produced. The `zip` and `zinb` commands fit such models; see [R] **zip** and [R] **zinb**.

26.14 Exact estimators

Exact estimators refer to models that, rather than being estimated by asymptotic formulas, are estimated by enumerating the conditional distribution of the sufficient statistics and then computing the maximum likelihood estimate using that distribution. Standard errors cannot be estimated, but confidence intervals can be and are obtained from the enumerations.

`exlogistic` fits logistic models of binary data in this way.

`expoisson` fits Poisson models of count data in this way.

In small samples, exact estimates have better coverage than the asymptotic estimates, and exact estimates are the only way to obtain estimates, tests, and confidence intervals of covariates that perfectly predict the observed outcome.

26.15 Linear regression with heteroskedastic errors

We now consider the model $y_j = \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j$, where the variance of ϵ_j is nonconstant.

`regress` can fit such models if you specify the `vce(robust)` option. What Stata calls robust is also known as the White correction for heteroskedasticity.

For scientists who have data where the variance of ϵ_j is known a priori, `vwls` is the command. `vwls` produces estimates for the model given each observation's variance, which is recorded in a variable in the data.

If you wish to model the heteroskedasticity on covariates, use the `het()` option of the `arch` command. Although `arch` is written primarily to analyze time-series data, it can be used with cross-sectional data. Before using `arch` with cross-sectional data, set the data as time series by typing `generate faketime = _n` and then typing `tsset faketime`.

`qreg` performs quantile regression, which in the presence of heteroskedasticity is most of interest. Median regression (one of `qreg`'s capabilities) is an estimator of $y_j = \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j$ when ϵ_j is heteroskedastic. Even more useful, you can fit models of other quantiles and so model the heteroskedasticity. Also see the `sqreg` and `iqreg` commands; `sqreg` estimates multiple quantiles simultaneously. `iqreg` estimates differences in quantiles.

26.16 Stochastic frontier models

`frontier` fits stochastic production or cost frontier models on cross-sectional data. The model can be expressed as

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + v_i - su_i$$

where

$$s = \begin{cases} 1 & \text{for production functions} \\ -1 & \text{for cost functions} \end{cases}$$

u_i is a nonnegative disturbance standing for technical inefficiency in the production function or cost inefficiency in the cost function. Although the idiosyncratic error term v_i is assumed to have a normal distribution, the inefficiency term is assumed to be one of the three distributions: half-normal, exponential, or truncated-normal. Also, when the nonnegative component of the disturbance is assumed to be either half-normal or exponential, `frontier` can fit models in which the error components are heteroskedastic conditional on a set of covariates. When the nonnegative component of the disturbance is assumed to be from a truncated-normal distribution, `frontier` can also fit a conditional mean model, where the mean of the truncated-normal distribution is modeled as a linear function of a set of covariates.

For panel-data stochastic frontier models, see **[U] 26.20.1 Linear regression with panel data**.

26.17 Regression with systems of equations

For systems of equations with endogenous covariates, use the three-stage least-squares (3SLS) estimator `reg3`, which can produce constrained and unconstrained estimates.

When we have correlated errors across equations but no endogenous right-hand-side variables,

$$\begin{aligned} y_{1j} &= \mathbf{x}_{1j}\boldsymbol{\beta} + \epsilon_{1j} \\ y_{2j} &= \mathbf{x}_{2j}\boldsymbol{\beta} + \epsilon_{2j} \\ &\vdots \\ y_{mj} &= \mathbf{x}_{mj}\boldsymbol{\beta} + \epsilon_{mj} \end{aligned}$$

where $\epsilon_{k\cdot}$ and $\epsilon_{l\cdot}$ are correlated with correlation ρ_{kl}, a quantity to be estimated from the data. This is called Zellner's seemingly unrelated regressions, and `sureg` fits such models. When $\mathbf{x}_{1j} = \mathbf{x}_{2j} = \cdots = \mathbf{x}_{mj}$, the model is known as multivariate regression, and the corresponding command is `mvreg`.

The equations need not be linear; if they are not linear, use `nlsur`.

26.18 Models with endogenous sample selection

What has become known as the Heckman model refers to linear regression in the presence of sample selection: $y_j = \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j$ is not observed unless some event occurs that itself has probability $p_j = F(\mathbf{z}_j\boldsymbol{\gamma} + \nu_j)$, where ϵ and ν might be correlated and $\mathbf{z}_j$ and $\mathbf{x}_j$ may contain variables in common.

`heckman` fits such models by maximum likelihood or Heckman's original two-step procedure.

This model has recently been generalized to replace the linear regression equation with another probit equation, and that model is fit by `heckprobit`. The command `heckoprobit` fits an ordered probit model in the presence of sample selection.

The endogenous treatment-effects model, which allows an endogenously determined binary treatment variable to affect an outcome, is also an example of an endogenous sample selection model. For more information, see **[U] 26.23 Treatment-effect models** and [TE] **eteffects**.

26.19 Models with time-series data

ARIMA refers to models with autoregressive integrated moving-average processes, and Stata's `arima` command fits models with ARIMA disturbances via the Kalman filter and maximum likelihood. These models may be fit with or without covariates. `arima` also fits ARMA models. See [TS] **arima**.

ARFIMA stands for "autoregressive fractionally integrated moving average" and handles long-memory processes. ARFIMA generalizes the ARMA and ARIMA models. ARMA models assume short memory; after a shock, the process reverts to its trend relatively quickly. ARIMA models assume shocks are permanent and memory never fades. ARFIMA provides a middle ground in the length of the process's memory. The `arfima` command fits ARFIMA models. In addition to one-step and dynamic forecasts, `arfima` can predict fractionally integrated series. See [TS] **arfima**.

UCM stands for "unobserved components model" and decomposes a time series into trend, seasonal, cyclic, and idiosyncratic components after controlling for optional exogenous variables. UCM provides a flexible and formal approach to smoothing and decomposition problems. The `ucm` command fits UCM models. See [TS] **ucm**.

The estimated parameters of ARIMA, ARFIMA, and UCM are sometimes more easily interpreted in terms of the implied spectral density. `psdensity` transforms results; see [TS] **psdensity**.

Band-pass and high-pass filters are also used to decompose a time series into trend and cyclic components, even though the `tsfilter` commands are not estimation commands; see [TS] **tsfilter**. Provided are Baxter–King, Butterworth, Christiano–Fitzgerald, and Hodrick–Prescott filters.

Stata's `prais` command performs regression with AR(1) disturbances using the Prais–Winsten or Cochrane–Orcutt transformation. Both two-step and iterative solutions are available, as well as a version of the Hildreth–Lu search procedure. See [TS] **prais**.

`newey` produces linear regression estimates with the Newey–West variance estimates that are robust to heteroskedasticity and autocorrelation of specified order. See [TS] **newey**.

Stata provides estimators for ARCH, GARCH, univariate, and multivariate models. These models are for time-varying volatility. ARCH models allow for conditional heteroskedasticity by including lagged variances. GARCH models also include lagged second moments of the innovations (errors). ARCH stands for "autoregressive conditional heteroskedasticity". GARCH stands for "generalized ARCH".

`arch` fits univariate ARCH and GARCH models, and the command provides many popular extensions, including multiplicative conditional heteroskedasticity. Errors may be normal or Student's t or may follow a generalized error distribution. Robust standard errors are optionally provided. See [TS] **arch**.

`mgarch` fits multivariate ARCH and GARCH models, including the diagonal vech model and the constant, dynamic, and varying conditional correlation models. Errors may be multivariate normal or multivariate Student's t. Robust standard errors are optionally provided. See [TS] **mgarch**.

Stata provides VAR, SVAR, and VEC estimators for modeling multivariate time series. VAR and SVAR deal with stationary series, and SVAR places additional constraints on the VAR model that identifies the impulse–response functions. VEC is for cointegrating VAR models. VAR stands for "vector autoregression"; SVAR, for "structural VAR"; and VEC, for "vector error-correction" model.

var fits VAR models, svar fits SVAR models, and vec fits VEC models. These commands share many of the same features for specification testing, forecasting, and parameter interpretation; see [TS] **var intro** for both var and svar, [TS] **vec intro** for vec, and [TS] **irf** for all three impulse–response functions and forecast-error variance decomposition. For lag-order selection, residual analysis, and Granger causality tests, see [TS] **var intro** (for var and svar) and [TS] **vec intro**.

sspace estimates the parameters of multivariate state-space models using the Kalman filter. The state-space representation of time-series models is extremely flexible and can be used to estimate the parameters of many different models, including vector autoregressive moving-average (VARMA) models, dynamic-factor (DF) models, and structural time-series (STS) models. It can also solve some stochastic dynamic-programming problems. See [TS] **sspace**.

dfactor estimates the parameters of dynamic-factor models. These flexible models for multivariate time series provide for a vector-autoregressive structure in both observed outcomes and unobserved factors. They also allow exogenous covariates for observed outcomes or unobserved factors. See [TS] **dfactor**.

Markov-switching models are used for series that transition over a finite set of unobserved states, allowing the process to evolve differently in each state. The transitions occur according to a Markov process. The time of transition from one state to another and the duration between changes in state are random.

Stata's mswitch command fits Markov-switching dynamic-regression (MSDR) and Markov-switching autoregression (MSAR) models. MSDR models can accommodate higher autoregressive lags than MSAR models because the state vector does not depend on the autoregressive lags in an MSDR model. See [TS] **mswitch**.

26.20 Panel-data models

26.20.1 Linear regression with panel data

This section could just as well be called "linear regression with complex error structures". Commands in this class begin with the letters xt.

xtreg fits models of the form

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

xtreg can produce the between-regression estimator, the within-regression (fixed-effects) estimator, or the GLS random-effects (matrix-weighted average of between and within results) estimator. It can also produce the maximum-likelihood random-effects estimator.

xtregar can produce the within estimator and a GLS random-effects estimator when the ϵ_{it} are assumed to follow an AR(1) process.

xtivreg contains the between-2SLS estimator, the within-2SLS estimator, the first-differenced-2SLS estimator, and two GLS random-effects-2SLS estimators to handle cases in which some of the covariates are endogenous.

xtabond is for use with dynamic panel-data models (models in which there are lagged dependent variables) and can produce the one-step, one-step robust, and two-step Arellano–Bond estimators. xtabond can handle predetermined covariates, and it reports both the Sargan and autocorrelation tests derived by Arellano and Bond.

xtdpdsys is an extension of xtabond and produces estimates with smaller bias when the coefficients of the AR process are large. xtpdsys is also more efficient than xtabond. Whereas xtabond uses moment conditions based on the differenced errors, xtpdsys uses moment conditions based on both the differenced errors and their levels.

xtdpd is an extension of xtdpdsys and can be used to estimate the parameters of a broader class of dynamic panel-data models. xtdpd can be used to fit models with serially correlated idiosyncratic errors, whereas xtdpdsys and xtabond assume no serial correlation. xtdpd can also be used with models where the structure of the predetermined variables is more complicated than that assumed by xtdpdsys or xtabond.

xtgls produces generalized least-squares estimates for models of the form

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \epsilon_{it}$$

where you may specify the variance structure of ϵ_{it}. If you specify that ϵ_{it} is independent for all i's and t's, xtgls produces the same results as regress up to a small-sample degrees-of-freedom correction applied by regress but not by xtgls.

You may choose among three variance structures concerning i and three concerning t, producing a total of nine different models. Assumptions concerning i deal with heteroskedasticity and cross-sectional correlation. Assumptions concerning t deal with autocorrelation and, more specifically, AR(1) serial correlation.

Alternative methods report the OLS coefficients and a version of the GLS variance–covariance estimator. xtpcse produces panel-corrected standard error (PCSE) estimates for linear cross-sectional time-series models, where the parameters are estimated by OLS or Prais–Winsten regression. When you are computing the standard errors and the variance–covariance estimates, the disturbances are, by default, assumed to be heteroskedastic and contemporaneously correlated across panels.

In the jargon of GLS, the random-effects model fit by xtreg has exchangeable correlation within i—xtgls does not model this particular correlation structure. xtgee, however, does.

xtgee fits population-averaged models, and it optionally provides robust estimates of variance. Moreover, xtgee allows other correlation structures. One that is of particular interest to those with many data goes by the name "unstructured". The within-panel correlations are simply estimated in an unconstrained way. [U] **26.20.3 Generalized linear models with panel data** will discuss this estimator further because it is not restricted to linear regression models.

xthtaylor uses instrumental-variables estimators to estimate the parameters of panel-data random-effects models of the form

$$y_{it} = \mathbf{X}_{1it}\boldsymbol{\beta}_1 + \mathbf{X}_{2it}\boldsymbol{\beta}_2 + \mathbf{Z}_{1i}\boldsymbol{\delta}_1 + \mathbf{Z}_{2i}\boldsymbol{\delta}_2 + u_i + e_{it}$$

The individual effects u_i are correlated with the explanatory variables $\mathbf{X}_{2it}$ and $\mathbf{Z}_{2i}$ but are uncorrelated with $\mathbf{X}_{1it}$ and $\mathbf{Z}_{1i}$, where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are constant within the panel.

xtfrontier fits stochastic production or cost frontier models for panel data. You may choose from a time-invariant model or a time-varying decay model. In both models, the nonnegative inefficiency term is assumed to have a truncated-normal distribution. In the time-invariant model, the inefficiency term is constant within panels. In the time-varying decay model, the inefficiency term is modeled as a truncated-normal random variable multiplied by a specific function of time. In both models, the idiosyncratic error term is assumed to have a normal distribution. The only panel-specific effect is the random inefficiency term.

See [U] **26.21 Multilevel mixed-effects models** for a generalization of xtreg that allows for multiple levels of panels, random coefficients, and variance-component estimation in general.

26.20.2 Censored linear regression with panel data

`xttobit` fits random-effects tobit models and generalizes that to observation-specific censoring.

`xtintreg` performs random-effects interval regression and generalizes that to observation-specific censoring. Interval regression, in addition to allowing open-ended intervals, allows closed intervals.

26.20.3 Generalized linear models with panel data

[U] 26.6 Generalized linear models discussed the model

$$g\{E(y_j)\} = \mathbf{x}_j\boldsymbol{\beta}, \qquad y_j \sim F \tag{1}$$

where $g()$ is the link function and F is a member of the exponential family, both of which you specify before estimation.

There are two ways to extend the generalized linear model to panel data. They are the generalized linear mixed model (GLMM) and generalized estimation equations (GEE).

GEE uses a working correlation structure to model within-panel correlation. GEEs may be fit with the `xtgee` command; see [XT] **xtgee**.

For generalized linear models with multilevel data, including panel data, see **[U] 26.21 Multilevel mixed-effects models**.

26.20.4 Qualitative dependent-variable models with panel data

`xtprobit` fits random-effects probit regression via maximum likelihood. It also fits population-averaged models via GEE. This last is nothing more than `xtgee` with the binomial family, probit link, and exchangeable error structure.

`xtlogit` fits random-effects logistic regression models via maximum likelihood. It also fits conditional fixed-effects models via maximum likelihood. Finally, as with `xtprobit`, it fits population-averaged models via GEE.

`xtcloglog` estimates random-effects complementary log-log regression via maximum likelihood. It also fits population-averaged models via GEE.

`xtologit` and `xtoprobit` are multiple-outcome models. `xtologit` fits a random-effects ordered logistic model, and `xtoprobit` fits a random-effects ordered probit model.

These models are generalizable to multilevel data; see **[U] 26.21 Multilevel mixed-effects models**.

26.20.5 Count dependent-variable models with panel data

`xtpoisson` fits two different random-effects Poisson regression models via maximum likelihood. The two distributions for the random effects are gamma and normal. `xtpoisson` also fits conditional fixed-effects models, and it fits population-averaged models via GEE. This last is nothing more than `xtgee` with the Poisson family, log link, and exchangeable error structure.

`xtnbreg` fits random-effects negative binomial regression models via maximum likelihood (the distribution of the random effects is assumed to be beta). `xtnbreg` also fits conditional fixed-effects models, and it fits population-averaged models via GEE.

These models are generalizable to multilevel data; see **[U] 26.21 Multilevel mixed-effects models**.

26.20.6 Survival models with panel data

xtstreg fits a random-effects parametric survival-time model by maximum likelihood. The conditional distribution of the response given the random effects is assumed to be exponential, Weibull, lognormal, loglogistic, or gamma. Depending on the selected distribution, xtstreg can fit models using a proportional hazards (PH) or accelerated failure-time (AFT) parameterization. Unlike the other panel-data commands, xtstreg requires that the data be xtset and stset.

These models are generalizable to multilevel data; see [U] **26.21 Multilevel mixed-effects models**.

26.20.7 Random-coefficients model with panel data

xtrc fits Swamy's random-coefficients linear regression model. In this model, rather than only the intercept varying across groups, all the coefficients are allowed to vary. xtrc is a special case of mixed.

Random-intercept and coefficient models can also be estimated for continuous, binary, count, ordered, and survival outcomes using multilevel modeling; see [U] **26.21 Multilevel mixed-effects models**.

26.21 Multilevel mixed-effects models

In multilevel data, observations—subjects, for want of a better word—can be divided into groups that have something in common. Perhaps the subjects are students, and the groups attended the same high school, or they are patients who were treated at the same hospital, or they are tractors that were manufactured at the same factory. Whatever they have in common, it may be reasonable to assume that the shared attribute affects the outcome being modeled.

With regard to students and high school, perhaps you are modeling later success in life. Some high schools are better (or worse) than others, so it would not be unreasonable to assume that the identity of the high school had an effect. With regard to patients and hospital, the argument is much the same if the outcome is subsequent health: some hospitals are better (or worse) than others, at least with respect to particular health problems. With regard to tractors and factory, it would hardly be surprising if tractors from some factories were more reliable than tractors from other factories.

Described above is two-level data. The first level is the student, patient, or tractor, and the second level is the high school, hospital, or factory. Observations are said to be nested within groups: students within a high school, patients within a hospital, or tractors within a factory.

Even though the effect on outcome is not directly observed, one can control for the effect if one is willing to assume that the effect is the same for all observations within a group and that, across groups, the effect is a random draw from a statistical distribution that is uncorrelated with the overall residual of the model and other group effects.

We have just described multilevel models.

A more complicated scenario might have three levels: students nested within teachers within a high school, patients nested within doctors within a hospital, or tractors nested within an assembly line within a factory.

An alternative to three-level hierarchical data is crossed data. We have workers and their occupation and the industry in which they work.

Stata provides a suite of multilevel estimation commands. The estimation commands are the following:

Command	Outcome variable	Equivalent to
mixed	continuous	linear regression
meprobit	binary	probit regression
melogit	binary	logistic regression
meqrlogit	binary	logistic regression [1]
mecloglog	binary	complementary log-log regression
meoprobit	ordered categorical	ordered probit regression
meologit	ordered categorical	ordered logistic regression
mepoisson	count	Poisson regression
meqrpoisson	count	Poisson regression [1]
menbreg	count	negative binomial regression
mestreg	survival-time	parametric survival-time regression
meglm	various	generalized linear models

[1] meqrlogit and meqrpoisson use QR decomposition to produce results. They exist for historical reasons but can be useful for estimation on the boundary of the parameter space. Results are numerically equivalent to melogit and mepoisson in other cases.

The above estimators provide random intercepts and random coefficients and allow constraints to be placed on coefficients and on variance components. (The QR decomposition estimators do not allow constraints.)

See the [ME] *Stata Multilevel Mixed-Effects Reference Manual*; in particular, see [ME] **me**.

26.22 Survival-time (failure-time) models

Commands are provided to fit Cox proportional hazards models, competing-risks regression, and several parametric survival models, including exponential, Weibull, Gompertz, lognormal, loglogistic, and generalized gamma; see [ST] **stcox**, [ST] **stcrreg**, and [ST] **streg**. The commands for Cox and parametric regressions, stcox and streg, respectively, are appropriate for single- or multiple-failure-per-subject data. The command for competing-risks regression, stcrreg, is appropriate only for single-failure data. Conventional, robust, bootstrap, and jackknife standard errors are available with all three commands, with the exception that for stcrreg, robust standard errors are the conventional standard errors.

Both the Cox model and the parametric models (as fit using Stata) allow for two additional generalizations. First, the models may be modified to allow for latent random effects, or frailties. Second, the models may be stratified in that the baseline hazard function may vary completely over a set of strata. The parametric models also allow for the modeling of ancillary parameters.

Competing-risks regression, as fit using Stata, is a useful alternative to Cox regression for datasets where more than one type of failure occurs, in other words, for data where failure events compete with one another. In such situations, competing-risks regression allows you to easily assess covariate effects on the incidence of the failure type of interest without having to make strong assumptions concerning the independence of failure types.

stcox, stcrreg, and streg require that the data be stset so that the proper response variables can be established. After you stset the data, the time/censoring response is taken as understood, and you need supply only the regressors (and other options) to stcox, stcrreg, and streg. With stcrreg, one required option deals with specifying which events compete with the failure event of interest that was previously stset.

Stata also provides commands to estimate average treatment effects and average treatment effects on the treated from observational survival-time data. See **[U] 26.23 Treatment-effect models**.

We discuss panel-data survival-time models in **[U] 26.20.6 Survival models with panel data**. These models generalize to multilevel data; see **[U] 26.21 Multilevel mixed-effects models**.

26.23 Treatment-effect models

teffects, stteffects, and eteffects estimate treatment effects from observational data.

A treatment effect is the change in an outcome caused by an individual getting one treatment instead of another. We can estimate average treatment effects, but not individual-level treatment effects, because we observe only each individual getting one or another treatment.

teffects, stteffects, and eteffects use methods that specify what the individual-level outcomes would be for each treatment level, even though only one of them can be realized. This approach is known as the potential-outcome framework. See [TE] **teffects intro** for a basic introduction to the key concepts associated with observational data analysis. See [TE] **teffects intro advanced** for a more advanced introduction that provides the intuition behind the potential-outcome framework. [TE] **stteffects intro** extends the concepts in the two earlier introductions to survival-time data.

Suppose we want to use observational data to learn about the effect of exercise on blood pressure. The potential-outcome framework provides the structure to estimate what would be the average effect of everyone exercising instead of everyone not exercising, an effect known as average treatment effect (ATE). Similarly, we can estimate the average effect, among those who exercise, of exercising instead of not exercising, which is known as the average treatment effect on the treated (ATET). Finally, we could estimate the average blood pressure that would be obtained if everyone exercised or if no one exercised, parameters known as potential-outcome means (POMs).

teffects can estimate the ATE, the ATET, and the POMs. The estimators implemented in teffects impose the structure of the potential-outcome framework on the data in different ways.

- Regression-adjustment estimators use models for the potential outcomes. See [TE] **teffects ra**.
- Inverse-probability-weighted estimators use models for treatment assignment. See [TE] **teffects ipw**.
- Augmented inverse-probability-weighted estimators and inverse-probability-weighted regression-adjustment estimators use models for the potential outcomes and for treatment assignment. These estimators have the double-robust property; they correctly estimate the treatment effect even if only one of the two models is correctly specified. See [TE] **teffects aipw** and [TE] **teffects ipwra**.
- Nearest-neighbor matching (NNM) and propensity-score matching (PSM) estimators compare the outcomes of individuals who are similar as possible except that one gets the treatment and the other does not. NNM uses a nonparametric similarity measure, while PSM uses estimated treatment probabilities to measure similarity. See [TE] **teffects nnmatch** and [TE] **teffects psmatch**.

stteffects can estimate the ATE, the ATET, and the POMs. The estimators implemented in stteffects impose the structure of the potential-outcome framework on the data in different ways.

- Regression-adjustment estimators use models for the potential outcomes, and censoring is adjusted for in the log-likelihood function. See [TE] **stteffects ra**.

- Inverse-probability-weighted estimators use models for treatment assignment and for the censoring time. See [TE] **stteffects ipw**.
- Inverse-probability-weighted regression-adjustment (IPWRA) estimators use models for the potential outcomes and for treatment assignment. IPWRA estimators can adjust for censoring in the outcome model or with a separate censoring model. These estimators have the double-robust property; they correctly estimate the treatment effect even if only the outcome model or the treatment-assignment model is correctly specified. If a censoring model is specified, both the treatment-assignment model and the censoring model must be correctly specified for the estimator to be double robust. See [TE] **stteffects ipwra**.
- Weighted regression-adjustment estimators model the outcome and the time to censoring. See [TE] **stteffects wra**.

`teffects` and `stteffects` can estimate treatment effects from multivalued treatments; see [TE] **teffects multivalued**.

It is not appropriate to use `teffects` or `stteffects` when a treatment is endogenously determined (the potential outcomes are not conditionally independent). When the treatment is endogenous, an endogenous treatment-effects model can be used to estimate the average treatment effect. These models consider the effect of an endogenously determined binary treatment variable on the outcome.

`eteffects` can estimate the ATE, the ATET, and the POMs. It fits endogenous treatment-effects models by using either a linear or a nonlinear (probit, fractional probit, or exponential) model for the outcome. `eteffects` implements control-function regression-adjustment estimators.

`etregress` and `etpoisson` also fit endogenous treatment-effects models and can be used to estimate the ATE and the ATET. See [TE] **etregress** and [TE] **etpoisson**. `etregress` fits an endogenous treatment-effects model by using a linear model for the outcome. `etpoisson` fits an endogenous treatment-effects model by using a nonlinear (exponential) model for the outcome.

26.24 Generalized method of moments (GMM)

`gmm` fits models using generalized method of moments (GMM). With the interactive version of the command, you enter your moment equations directly into the dialog box or command line using substitutable expressions just like with `nl` or `nlsur`. The moment-evaluator program version gives you greater flexibility in exchange for increased complexity; with this version, you write a program that calculates the moments based on a vector of parameters passed to it.

`gmm` can fit both single- and multiple-equation models, and you can combine moment conditions of the form $E\{\mathbf{z}_i u_i(\boldsymbol{\beta})\} = \mathbf{0}$, where $\mathbf{z}_i$ is a vector of instruments and $u_i(\boldsymbol{\beta})$ is often an additive regression error term, as well as more general moment conditions of the form $E\{\mathbf{h}_i(\mathbf{z}_i; \boldsymbol{\beta})\} = \mathbf{0}$. In the former case, you specify the expression for $u_i(\boldsymbol{\beta})$ and use the `instruments()` and `xtinstruments()` options to specify $\mathbf{z}_i$. In the latter case, you specify the expression for $\mathbf{h}_i(\mathbf{z}_i; \beta)$; because that expression incorporates your instruments, you do not use the `instruments()` or `xtinstruments()` option.

`gmm` supports cross-sectional, time-series, and panel data. You can request weight matrices and VCEs that are suitable for independent and identically distributed errors, that are suitable for heteroskedastic errors, that are appropriate for clustered observations, or that are heteroskedasticity- and autocorrelation-consistent (HAC). For HAC weight matrices and VCEs, `gmm` lets you specify the bandwidth or request an automatic bandwidth selection algorithm.

26.25 Estimation with correlated errors

By correlated errors, we mean that observations are grouped and that within a group, the observations might be correlated but that across groups, they are uncorrelated. `regress` with the `vce(cluster` *clustvar*`)` option can produce "correct" estimates, that is, inefficient estimates with correct standard errors and lots of robustness; see **[U] 20.21 Obtaining robust variance estimates**. Obviously, if you know the correlation structure (and are not mistaken), you can do better, so `xtreg` and `xtgls` are also of interest here; we discuss them in **[U] 26.20.1 Linear regression with panel data**.

Estimation in the presence of autocorrelated errors is discussed in **[U] 26.19 Models with time-series data**.

26.26 Survey data

Stata's `svy` command fits statistical models for complex survey data. `svy` is a prefix command, so to obtain linear regression, you type

```
. svy: regress ...
```

or to obtain probit regression, you type

```
. svy: probit ...
```

but you must first type a `svyset` command to define the survey design characteristics. Prefix `svy` works with many estimation commands, and everything is documented together in the *Stata Survey Data Reference Manual*.

`svy` supports the following variance-estimation methods:

- Taylor-series linearization
- Bootstrap
- Balanced repeated replication (BRR)
- Jackknife
- Successive difference replication (SDR)

See [SVY] **variance estimation** for details.

`svy` supports the following survey design characteristics:

- With- and without-replacement sampling
- Observation-level sampling weights
- Stage-level sampling weights
- Stratification
- Poststratification
- Clustering
- Multiple stages of clustering without replacement
- BRR and jackknife replication weights

See [SVY] **svyset** for details. For an application of the `svy` prefix with stage-level sampling weights, see example 6 in [ME] **meglm**.

Subpopulation estimation is available for all estimation commands.

Tabulations and summary statistics are also available, including means, proportions, ratios, and totals over multiple subpopulations and direct standardization of means, proportions, and ratios.

See [SVY] **survey**.

26.27 Multiple imputation

Multiple imputation (MI) is a statistical technique for estimation in the presence of missing data. If you estimate the parameters of y on x_1, x_2, and x_3 using any of the other Stata estimation commands, parameters are estimated on the data for which y, x_1, x_2, and x_3 contain no missing values. This process is known as listwise or casewise deletion because observations for which any of y, x_1, x_2, or x_3 contain missing values are ignored or, said differently, deleted from consideration. MI is a technique to recover the information in those ignored observations when the missing values are missing at random (MAR) or missing completely at random (MCAR). Data are MAR if the probability that a value is missing may depend on observed data but not on unobserved data. Data are MCAR if the probability of missingness is not even a function of the observed data.

MI is named for the imputations it produces to replace the missing values in the data. MI does not just form replacement values for the missing data; it produces multiple replacements. The purpose is not to create replacement values as close as possible to the true ones but to handle missing data in a way resulting in valid statistical inference.

There are three steps in an MI analysis. First, one forms M imputations for each missing value in the data. Second, one fits the model of interest separately on each of the M resulting datasets. Finally, one combines those M estimation results into the desired single result.

The `mi` command does this for you. It can be used with most of Stata's estimation commands, including with survey, survival, and panel and multilevel models. See [MI] **intro**.

26.28 Multivariate and cluster analysis

Most of Stata's multivariate capabilities can be found in the *Multivariate Statistics Reference Manual*, although there are some exceptions.

1. `mvreg` fits multivariate regressions.
2. `manova` fits MANOVA and MANCOVA models, one-way and up—including two-way factorial, three-way factorial, etc.—and it fits nested and mixed-design models. Also see **[U] 26.5 ANOVA, ANCOVA, MANOVA, and MANCOVA** above.
3. `canon` estimates canonical correlations and their corresponding loadings. Canonical correlation attempts to describe the relationship between two sets of variables.
4. `pca` extracts principal components and reports eigenvalues and loadings. Some people consider principal components a descriptive tool—in which case standard errors as well as coefficients are relevant—and others look at it as a dimension-reduction technique.
5. `factor` fits factor models and provides principal factors, principal-component factors, iterated principal-component factors, and maximum likelihood solutions. Factor analysis is concerned with finding few common factors $\widehat{\mathbf{z}}_k$, $k = 1, \ldots, q$, that linearly reconstruct the original variables $\mathbf{y}_i$, $i = 1, \ldots, L$.
6. `tetrachoric`, in conjunction with `pca` or `factor`, allows you to perform PCA or factor analysis on binary data.
7. `rotate` provides a wide variety of orthogonal and oblique rotations after `factor` and `pca`. Rotations are often used to produce more interpretable results.

8. procrustes performs Procrustes analysis, one of the standard methods of multidimensional scaling. It can perform orthogonal or oblique rotations as well as translation and dilation.
9. mds performs metric and nonmetric multidimensional scaling for dissimilarity between observations with respect to a set of variables. A wide variety of dissimilarity measures are available and, in fact, are the same as those for cluster.
10. ca performs correspondence analysis, an exploratory multivariate technique for analyzing cross-tabulations and the relationship between rows and columns.
11. mca performs multiple correspondence analysis (MCA) and joint correspondence analysis (JCA).
12. mvtest performs tests of multivariate normality along with tests of means, covariances, and correlations.
13. cluster provides cluster analysis; both hierarchical and partition clustering methods are available. Strictly speaking, cluster analysis does not fall into the category of statistical estimation. Rather, it is a set of techniques for exploratory data analysis. Stata's cluster environment has many different similarity and dissimilarity measures for continuous and binary data.
14. discrim and candisc perform discriminant analysis. candisc performs linear discriminant analysis (LDA). discrim also performs LDA, and it performs quadratic discriminant analysis (QDA), *k*th nearest neighbor (KNN), and logistic discriminant analysis. The two commands differ in default output. discrim shows the classification summary, candisc shows the canonical linear discriminant functions, and both will produce either.

26.29 Pharmacokinetic data

There are four estimation commands for analyzing pharmacokinetic data. See [R] **pk** for an overview of the pk system.

1. pkexamine calculates pharmacokinetic measures from time-and-concentration subject-level data. pkexamine computes and displays the maximum measured concentration, the time at the maximum measured concentration, the time of the last measurement, the elimination time, the half-life, and the area under the concentration-time curve (AUC).
2. pksumm obtains the first four moments from the empirical distribution of each pharmacokinetic measurement and tests the null hypothesis that the distribution of that measurement is normally distributed.
3. pkcross analyzes data from a crossover design experiment. When one is analyzing pharmaceutical trial data, if the treatment, carryover, and sequence variables are known, the omnibus test for separability of the treatment and carryover effects is calculated.
4. pkequiv performs bioequivalence testing for two treatments. By default, pkequiv calculates a standard confidence interval symmetric about the difference between the two treatment means. pkequiv also calculates confidence intervals symmetric about zero and intervals based on Fieller's theorem. Also, pkequiv can perform interval hypothesis tests for bioequivalence.

26.30 Specification search tools

There are three other commands that are not really estimation commands but are combined with estimation commands to assist in specification searches: stepwise, fp, and mfp.

stepwise, one of Stata's prefix commands, provides stepwise estimation. You can use the stepwise prefix with some, but not all, estimation commands. See [R] **stepwise** for a list of supported estimation commands.

fp and mfp are commands to assist you in performing fractional-polynomial functional specification searches.

26.31 Power and sample-size analysis

Power and sample-size (PSS) analysis is performed during the planning stage of a study. The main goal of PSS analysis is to provide an estimate of the sample size needed to successfully achieve the research objective of a study.

For example, suppose that we want to design a study to evaluate a new drug for lowering blood pressure. We want to test whether the mean blood pressure of the experimental group, which will receive the new drug, is the same as the mean blood pressure of the control group, which will receive the old drug. The post hoc analysis will use a two-sample t test to test the difference between the two means. How many subjects do we need to enroll in our study to detect a difference between means that is of clinical importance? PSS analysis can answer this question.

PSS analysis can also answer other questions that may arise during the planning stage of a study. For example, what is the power of a test given an available sample size, and how likely is it to detect an effect of interest given limited study resources? The answers to these questions may help reduce the cost of a study by preventing an overpowered study or may help avoid wasting resources on an underpowered study.

See [PSS] **intro** for more information about PSS analysis.

The power command performs PSS analysis. It provides PSS analysis for comparison of means, variances, proportions, and correlations, for comparison of contingency tables, and for comparison of survival-time data. One-sample, two-sample, and paired analyses of means, variances, proportions, and correlations are supported. Contingency table analyses may be for matched samples, $2 \times 2 \times K$ tables, or $2 \times J$ tables. For survival-time data, one-sample analysis is supported for Cox proportional hazards models; two-sample analysis is supported for parametric or nonparametric comparison of survivor functions.

power provides both tabular output and graphical output, or power curves; see [PSS] **power, table** and [PSS] **power, graph** for details.

See [PSS] **power** for a full list of supported methods and the description of the command.

You can work with power commands either interactively or via a convenient point-and-click interface; see [PSS] **GUI** for details.

26.32 Bayesian analysis

Bayesian analysis is a statistical analysis that answers research questions about unknown parameters of statistical models by using probability statements. Bayesian analysis rests on the assumption that all model parameters are random quantities and are subject to prior knowledge. This assumption is in sharp contrast with more traditional, frequentist analysis where all parameters are considered unknown but fixed quantities.

Bayesian analysis is based on modeling and summarizing the posterior distribution of parameters conditional on the observed data. The posterior distribution is composed of a likelihood distribution of the data and the prior distribution of the model parameters. Many posterior distributions do not have a closed form and must be approximated by using, for example, Markov chain Monte Carlo (MCMC) methods such as Metropolis–Hastings (MH) methods, the Gibbs method, or sometimes their combination. The convergence of MCMC must be verified before any inference can be made.

In Bayesian analysis, marginal posterior distributions of parameters are used for inference. They are summarized using point estimators, such as posterior mean and median, and using interval estimators, such as equal-tailed credible intervals and highest-posterior density intervals.

Stata provides a suite of commands for conducting Bayesian analysis. The `bayesmh` command fits a variety of Bayesian regression models by using an MH MCMC method. You can choose from a variety of supported Bayesian models, or you can program your own Bayesian models; see [BAYES] **bayesmh** and [BAYES] **bayesmh evaluators**.

Convergence of MCMC can be accessed visually by using `bayesgraph`. Marginal summaries can be obtained by using `bayesstats summary`, and hypothesis testing can be performed by using `bayestest`; see [BAYES] **bayesmh postestimation**.

See [BAYES] **bayes** for more information about commands and for a quick *Overview example*.

26.33 Obtaining new estimation commands

This chapter has discussed all the official estimation commands included in Stata 14. Users may have written their own estimation commands that they are willing to share. Type `search estimation`, `ssc new`, and `ssc hot` to discover more estimation commands; see [R] **ssc**.

And, of course, you can always write your own commands; see [R] **ml**.

26.34 References

Gould, W. W. 2000. sg124: Interpreting logistic regression in all its forms. *Stata Technical Bulletin* 53: 19–29. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 257–270. College Station, TX: Stata Press.

——. 2011. Use poisson rather than regress; tell a friend. The Stata Blog: Not Elsewhere Classified. http://blog.stata.com/2011/08/22/use-poisson-rather-than-regress-tell-a-friend/.