The Stata Journal (2022)
22, Number 4, pp. 734–771 DOI: 10.1177/1536867X221140943

Endogenous models of binary choice outcomes: Copula-based maximum-likelihood estimation and treatment effects

Takuya Hasebe
Sophia University
Tokyo, Japan
thasebe@sophia.ac.jp

Abstract. In this article, I describe the commands that implement the estimation of three endogenous models of binary choice outcome. The command `esbinary` fits the endogenously switching model, where a potential outcome differs across two treatment states. The command `edbinary` fits the endogenous dummy model, which includes a dummy variable indicating the treatment state as one of the explanatory variables. After one estimates the parameters of these models, various treatment effects can be estimated as postestimation statistics. The command `ssbinary` fits the sample-selection model, where an outcome is observed in only one of the states. The commands fit these models using copula-based maximum-likelihood estimation.

Keywords: st0691, esbinary, edbinary, ssbinary, endogeneity, treatment effects, binary outcome, copula-based maximum-likelihood estimation, endogenous switching model, sample selection

1 Introduction

The issue of endogeneity is a common problem in empirical studies. Researchers also frequently encounter binary choice outcomes. There are various kinds of endogeneity, and different econometric models of binary choice outcome have been proposed to deal with different kinds of endogeneity. The endogeneity issue fundamentally stems from the dependence of unobservable terms in a model. To estimate model parameters consistently, one must model the dependence of unobservable terms. A traditional and commonly used assumption is joint normality. However, it is restrictive, and the violation of the distributional assumption results in inconsistency. To relax the distributional assumption, I consider a copula-based approach that generates a joint distribution by combining two separate marginal distributions. The copula-based approach includes the joint normality assumption as its special case, but it also allows for various nonnormal dependence structures of unobservable terms.

In this article, I discuss three econometric models of binary choice outcomes with potential endogeneity that are closely related to one another, and I introduce the commands to fit these models using the copula-based maximum-likelihood estimation method. The first model is an endogenous switching model, where potential outcomes differ across two alternative treatment states and the treatment state is determined

endogenously. Hasebe (2021) discusses the copula-based approach of this model. The command `esbinary` implements estimation. This command is the copula-based extension of the command `switch_probit` by Lokshin and Sajaia (2011), which fits the same model under the assumption of joint normality.

In the second model, a potential endogenous dummy variable enters as one of the variables to explain a binary choice outcome. This model can be seen as a special case of the first model with some restrictions, such as the equality of coefficients across two treatment states except for constant terms. Winkelmann (2012) and Hasebe (2013a) discuss the copula-based approach of this endogenous dummy model. The Stata command `biprobit` fits this model under the joint normality assumption. The command `edbinary` implements copula-based maximum-likelihood estimation.

The third model is also another special case of the first model, where a binary choice outcome is observed in one of two possible states but is missing in the other state. This is a well-known form of the sample-selection model. Dancer, Rammohan, and Smith (2008) discuss the copula-based approach of this model. The command `ssbinary` implements the copula-based maximum-likelihood estimation of this model, which relaxes the joint normality assumption that the Stata command `heckprobit` makes.

The literature has also discussed the copula-based estimation of the endogenous models of continuous outcome. See, for example, Lee (1983), Smith (2003, 2005), and Trivedi and Zimmer (2007). Hasebe (2013b) introduces the commands to implement the copula-based estimation of the endogenous switching and sample-selection models of continuous outcome. The commands introduced in this article are the binary choice outcome counterparts of the commands by Hasebe (2013b).

The endogenous switching model and its special case, the endogenous dummy model, are useful model specifications to estimate treatment effects. Different treatment effects are defined to allow for heterogeneity in treatment effects among different populations (Heckman and Vytlacil 2007). The endogenous switching model enables us to derive various treatment effects straightforwardly. Building on the endogenous switching model, Hasebe (2021) expresses the treatment effects for binary choice outcomes in terms of copulas. After one executes `esbinary` and `edbinary`, the postestimation command `estat teffects` is available to estimate some treatment effects. Specifically, the command options allow the estimation of the average treatment effect (ATE), average treatment effect on the treated (ATT), average treatment effect on the untreated (ATU), local average treatment effect (LATE), and marginal treatment effect (MTE).

Note that the estimation of the treatment effects by these commands relies on the parametric distributional assumptions because the commands allow for only the options of parametric copulas and marginal distributions. Such parametric assumptions may be considered as restrictive in the recent literature of program evaluation because the literature has moved toward less stringent assumptions in modeling and estimating treatment effects. Indeed, several semiparametric estimators for the treatment effects have been proposed. For example, the package `mtefe` by Andresen (2018) implements several estimation methods for MTE. However, these estimators are not developed specif-

ically for the binary choice outcome.[1] These commands provide a benchmark estimation result under the distributional assumption that is parametric but weaker than the joint normality.

The structure of this article is as follows. The next section outlines the three models of the binary choice outcome. Section 3 briefly explains the copula method and the copula-based maximum-likelihood estimation of the models. Section 4 discusses various treatment effects proposed in the literature and shows the expressions of these treatment effects based on the endogenous switching model. Section 3 describes the commands `esbinary`, `edbinary`, and `ssbinary`, followed by data applications in section 6. Section 5 concludes the article.

2 Models

This section describes the three models of binary choice outcome, which are closely related to one another. In the first model, a potential outcome differs across two alternative states (or regimes), and the state is endogenously determined. We call this an endogenous switching model. The second model, which can be considered as a special case of the first model, has a dummy variable indicating an endogenously chosen treatment state as one of explanatory variables. We refer to this as an endogenous dummy model. In the third model, which can also be seen as another special case of the first model, an outcome is missing in one of two possible states. Like the two models above, which state is chosen is endogenously determined. We call this model a sample-selection model.

The endogenous switching model consists of three equations: a selection equation and two outcome equations. For an individual i, $i = 1, \ldots, N$, the selection equation is

$$d_i = 1(\mathbf{z}_i{}'\boldsymbol{\gamma} + \nu_i \geq 0) \tag{1}$$

where $1(\cdot)$ is an indicator function, $\mathbf{z}_i$ is a vector of observable characteristics, and ν_i is an unobservable term. There are two potential binary outcomes, y_{0i} and y_{1i}, which are determined as follows:

$$y_{0i} = 1(\mathbf{x}_{0i}{}'\boldsymbol{\beta}_0 + \varepsilon_{0i} \geq 0)$$
$$y_{1i} = 1(\mathbf{x}_{1i}{}'\boldsymbol{\beta}_1 + \varepsilon_{1i} \geq 0)$$

$\mathbf{x}_{0i}$ and $\mathbf{x}_{1i}$ are vectors of explanatory variables including constant terms, and ε_{0i} and ε_{1i} are unobservable terms. Although $\mathbf{x}_{0i}$ and $\mathbf{x}_{1i}$ can be different in theory,[2] it is usual

1. Han and Vytlacil (2017) discuss an identification of the endogenous dummy model with unknown marginal distributions, and Han and Lee (2019) propose a sieve maximum-likelihood estimation of the endogenous dummy model with parametric copula and nonparametric marginal distributions. Chiburis (2010) presents semiparametric bounds of ATE on a binary choice outcome. A factor structure model is an alternative approach to model the dependence of unobservable terms. Aakvik, Heckman, and Vytlacil (2005) define the treatment effects for binary choice outcome in a factor structure model.
2. The command `esbinary` allows sets of explanatory variables to be different across the two states.

to have the same set of explanatory variables in practice. Without loss of generality, we assume that $\mathbf{x}_{0i}$ and $\mathbf{x}_{1i}$ are the same and denote them by $\mathbf{x}_i$. Depending on the value of d, we observe either y_0 or y_1, but both outcomes cannot be observed simultaneously for the same individual: $y_i = d_i y_{1i} + (1 - d_i) y_{0i}$. The endogeneity issue arises when ε_j, $j = 0, 1$, is not independent of ν.

Note that, although it is standard that an unobservable term enters with a positive sign in a binary choice model, the policy evaluation literature often includes an unobservable term with a negative sign in the selection equation. That is, $d_i = 1(\mathbf{z}_i{}'\boldsymbol{\gamma} - \nu_i \geq 0)$ instead of $d_i = 1(\mathbf{z}_i{}'\boldsymbol{\gamma} + \nu_i \geq 0)$. This modification does not alter the interpretation of the coefficients, although it changes the expression of joint probabilities. In the copula-based approach, changing the sign of an unobservable term is of practical use. We will return to this point later.

The endogenous dummy model can be seen as a special case of the first model. In this model, constant terms differ by the treatment state, but coefficients on observables are the same across the two states. The unobservable terms are also assumed to be identical across the states. In other words, we can obtain the endogenous dummy model by restricting $\varepsilon_0 = \varepsilon_1 = \varepsilon$ and $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_1$ except for constant terms. Only one outcome equation exists, and it is written as follows:

$$y_i = 1(\alpha d_i + \mathbf{x}_i{}'\boldsymbol{\beta} + \varepsilon_i \geq 0)$$

In the potential outcome framework, the outcomes in this model can be interpreted as $y_{1i} = 1(\alpha + \mathbf{x}_i{}'\boldsymbol{\beta} + \varepsilon_i \geq 0)$ and $y_{0i} = 1(\mathbf{x}_i{}'\boldsymbol{\beta} + \varepsilon_i \geq 0)$. The equation for a dummy variable d_i is formalized as the selection (1). The dependence between ε and ν makes d endogenous.

The sample-selection model is also a special case of the first model. An outcome for an individual i is observed only when $d_i = 1$, but it is missing when $d_i = 0$. Like the endogenous dummy model, there is only one outcome equation in this model:

$$y_i = 1(\mathbf{x}_i{}'\boldsymbol{\beta} + \varepsilon_i \geq 0) \qquad \text{if } d_i = 1$$

The selection mechanism is governed by the selection (1). The sample-selection issue is attributed to the dependence between ε and ν.

To estimate the parameters in those models, one must model the dependence between the unobservable term in the selection equation, ν, and that in the outcome equation(s), ε ($\varepsilon_0/\varepsilon_1$). To this end, we use a copula approach.

3 Maximum likelihood estimation

I fit the models described above using maximum likelihood estimation. I first illustrate the maximum likelihood estimation of the endogenous switching model. I explain the copula-based approach to model the dependence between the unobservable terms and discuss some related issues. I leave out detailed explanations of the copula method. See, for example, Nelsen (2006) for a general introduction of copulas and Trivedi and

Zimmer (2007) for the use of copulas in microeconometric studies. At the end of this section, I briefly discuss the log-likelihood functions for the other two models.

The log-likelihood function of the endogenous switching model has the following general form:

$$\begin{aligned}\ln L = \sum_{i=1}^{N} [& d_i \{ y_i \ln \Pr(y_{1i}=1, d_i=1) + (1-y_i) \ln \Pr(y_{1i}=0, d_i=1) \} \\ & + (1-d_i) \{ y_i \ln \Pr(y_{0i}=1, d_i=0) + (1-y_i) \ln \Pr(y_{0i}=0, d_i=0) \}]\end{aligned}$$

$\Pr(y_{ji}, d_i)$ for $j = 0, 1$ is the joint probability of d and y_j conditional on the observable characteristics x and z. To specify the joint probabilities of d and y_j, we specify the joint distributions of ε_j and ν. Let $F_j(\varepsilon_j)$ be a marginal cumulative distribution function (c.d.f.) of ε_j, and let $F_d(\nu)$ be a marginal c.d.f. of ν. Let $F_{jd}(\varepsilon_j, \nu)$ be a joint c.d.f. of ε_j and ν. Then, given the observable characteristics $\mathbf{x}_i$ and $\mathbf{z}_i$, for example, the probability of observing $y_i = 1$ and $d_i = 1$ for an individual i, $\Pr(y_{1i} = 1, d_i = 1)$, can be written as $1 - F_1(-\mathbf{x}_i{}'\boldsymbol{\beta}_1) - F_d(-\mathbf{z}_i{}'\boldsymbol{\gamma}) + F_{1d}(-\mathbf{z}_i{}'\boldsymbol{\gamma}, -\mathbf{x}_i{}'\boldsymbol{\beta}_1)$. A joint normal distribution is the most conventional assumption. Under joint normality, $\Pr(y_{1i} = 1, d_i = 1) = 1 - \Phi(-\mathbf{x}_i{}'\boldsymbol{\beta}_1) - \Phi(-\mathbf{z}_i{}'\boldsymbol{\gamma}) + \Phi_2(-\mathbf{x}_i{}'\boldsymbol{\beta}_1, -\mathbf{z}_i{}'\boldsymbol{\gamma}; \rho_1)$, where $\Phi(\cdot)$ is the c.d.f. of univariate standard normal distribution, $\Phi_2(\cdot)$ is the c.d.f. of bivariate normal distribution, and ρ_1 is the coefficient of correlation between ε_1 and ν. Using the fact that the bivariate normal is symmetric, we can simplify this probability as $\Phi_2(\mathbf{z}_i{}'\boldsymbol{\gamma}, \mathbf{x}_i{}'\boldsymbol{\beta}_1; \rho_1)$.

A copula-based approach nests the joint normality but also allows nonnormal dependence. In short, a copula is a function that generates a joint distribution by binding marginal distributions. Using a copula function $C_j(\cdot)$, we can write the joint distribution of $F_{jd}(\varepsilon_j, \nu)$ as $C_j\{F_j(\varepsilon_j), F_d(\nu); \theta_j\}$, where θ_j is a (vector of) dependence parameters. We will discuss θ_j below. Note that it is not necessary to use the same copula function for the joint distributions of F_{0d} and F_{1d}. The subscript j for C_j indicates that a copula is for the joint distribution between ε_j and ν. In terms of copula, $\Pr(y_{1i} = 1, d_i = 1) = 1 - F_1(-\mathbf{x}_i{}'\boldsymbol{\beta}_1) - F_d(-\mathbf{z}_i{}'\boldsymbol{\gamma}) + C_1\{F_1(-\mathbf{x}_i{}'\boldsymbol{\beta}_1), -F_d(-\mathbf{z}_i{}'\boldsymbol{\gamma}); \theta_j\}$. To form the log-likelihood function, we need to specify other joint probabilities. See table 1 for all the relevant probabilities in terms of copula. It also lists the expression of the probabilities when each unobservable term enters with either a positive or a negative sign.

Table 1. Expressions of probabilities using a copula

$d = 1(\mathbf{z}'\boldsymbol{\gamma} + \nu > 0)$ and $y_j = 1(\mathbf{x}'\boldsymbol{\beta} + \varepsilon_j > 0)$	
$\Pr(y_j = 1)$	$1 - F_j(-\mathbf{x}'\boldsymbol{\beta}_j)$
$\Pr(d = 1)$	$1 - F_d(-\mathbf{z}'\boldsymbol{\gamma})$
$\Pr(y_j = 1, d = 1)$	$1 - F_d(-\mathbf{z}'\boldsymbol{\gamma}) - F_j(-\mathbf{x}'\boldsymbol{\beta}_j) + C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 0, d = 1)$	$F_j(-\mathbf{x}'\boldsymbol{\beta}_j) - C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1, d = 0)$	$F_d(-\mathbf{z}'\boldsymbol{\gamma}) - C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), u_d; \theta_j\}$
$\Pr(y_j = 0, d = 0)$	$C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1 \mid \nu = \widetilde{\nu})$	$1 - \dfrac{\partial C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(\nu); \theta_j\}}{\partial F_d}\Big\vert_{\nu=\widetilde{\nu}}$
$d = 1(\mathbf{z}'\boldsymbol{\gamma} + \nu > 0)$ and $y_j = 1(\mathbf{x}'\boldsymbol{\beta} - \varepsilon_j > 0)$	
$\Pr(y_j = 1)$	$F_j(\mathbf{x}'\boldsymbol{\beta}_j)$
$\Pr(d = 1)$	$1 - F_d(-\mathbf{z}'\boldsymbol{\gamma})$
$\Pr(y_j = 1, d = 1)$	$F_j(\mathbf{x}'\boldsymbol{\beta}_j) - C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 0, d = 1)$	$1 - F_d(-\mathbf{z}'\boldsymbol{\gamma}) - F_j(\mathbf{x}'\boldsymbol{\beta}_j) + C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1, d = 0)$	$C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 0, d = 0)$	$F_d(-\mathbf{z}'\boldsymbol{\gamma}) - C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(-\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1 \mid \nu = \widetilde{\nu})$	$\dfrac{\partial C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(\nu); \theta_j\}}{\partial F_d}\Big\vert_{\nu=\widetilde{\nu}}$
$d = 1(\mathbf{z}'\boldsymbol{\gamma} - \nu > 0)$ and $y_j = 1(\mathbf{x}'\boldsymbol{\beta} + \varepsilon_j > 0)$	
$\Pr(y_j = 1)$	$1 - F_j(-\mathbf{x}'\boldsymbol{\beta}_j)$
$\Pr(d = 1)$	$F_d(\mathbf{z}'\boldsymbol{\gamma})$
$\Pr(y_j = 1, d = 1)$	$F_d(\mathbf{z}'\boldsymbol{\gamma}) - C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma})\}$
$\Pr(y_j = 0, d = 1)$	$C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1, d = 0)$	$1 - F_d(\mathbf{z}'\boldsymbol{\gamma}) - F_j(-\mathbf{x}'\boldsymbol{\beta}_j) + C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 0, d = 0)$	$F_j(-\mathbf{x}'\boldsymbol{\beta}_j) - C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1 \mid \nu = \widetilde{\nu})$	$1 - \dfrac{\partial C_j\{F_j(-\mathbf{x}'\boldsymbol{\beta}_j), F_d(\nu); \theta_j\}}{\partial F_d}\Big\vert_{\nu=\widetilde{\nu}}$
$d = 1(\mathbf{z}'\boldsymbol{\gamma} - \nu > 0)$ and $y_j = 1(\mathbf{x}'\boldsymbol{\beta} - \varepsilon_j > 0)$	
$\Pr(y_j = 1)$	$F_j(\mathbf{x}'\boldsymbol{\beta}_j)$
$\Pr(d = 1)$	$F_d(\mathbf{z}'\boldsymbol{\gamma})$
$\Pr(y_j = 1, d = 1)$	$C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 0, d = 1)$	$F_d(\mathbf{z}'\boldsymbol{\gamma}) - C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1, d = 0)$	$F_j(\mathbf{x}'\boldsymbol{\beta}_j) - C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 0, d = 0)$	$1 - F_d(\mathbf{z}'\boldsymbol{\gamma}) - F_j(\mathbf{x}'\boldsymbol{\beta}_j) + C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(\mathbf{z}'\boldsymbol{\gamma}); \theta_j\}$
$\Pr(y_j = 1 \mid \nu = \widetilde{\nu})$	$\dfrac{\partial C_j\{F_j(\mathbf{x}'\boldsymbol{\beta}_j), F_d(\nu); \theta_j\}}{\partial F_d}\Big\vert_{\nu=\widetilde{\nu}}$

Although one of the main advantages of the copula approach is its ability to separate the specification of the marginal distributions of the unobservables from the specification of the dependence structure when modeling a joint distribution, we consider only common marginal distributions. Standard normal distribution (which corresponds to

probit) and logistic distribution (which corresponds to logit) are common choices for binary choice models. My commands provide an option of probit or logit for each marginal distribution. A future task could be to expand the commands by adding more flexible marginal distributions. Still, various dependence structures by copulas allow us to model a rich pattern of joint distributions.

Many different copula functions are available. Table 2 shows the list of copula functions available in the commands discussed in this article. See Nelsen (2006) and Trivedi and Zimmer (2007) for other copula functions. Among the copulas in table 2, Gaussian, Frank, Clayton, Gumbel, and Joe are popular parametric copulas in empirical applications. These copulas, other than Gaussian, belong to the Archimedean family, of which copulas have useful properties in empirical modeling (Smith 2003). Different copulas exhibit different dependence structures. For example, a Gaussian copula is symmetric in that the degrees of dependence are the same in both the lower and the upper tails of a joint distribution. A Gaussian copula with univariate standard normal margins corresponds to the standard assumption of joint normality. A Frank copula is also symmetric but has weaker tail dependence than a Gaussian copula. Clayton, Gumbel, and Joe are symmetric. A Clayton copula has strong lower-tail but weak upper-tail dependence, while Gumbel and Joe copulas have weaker lower-tail but strong upper-tail dependence. See Nelsen (2006) and Trivedi and Zimmer (2007) for graphical representations of dependence structures of different copulas.

Table 2. Copula functions

Copula name	$C(u_1, u_2; \theta)$
Product	$u_1 u_2$
Gaussian	$\Phi_2\{\Phi^{-1}(u_1), \Phi^{-1}(u_2); \theta\}$
Farlie–Gumbel–Morgenstern (FGM)	$u_1 u_2\{1 + \theta(1 - u_1)(1 - u_2)\}$
Plackett	$\dfrac{r - \sqrt{r^2 - 4u_1 u_2 \theta(\theta - 1)}}{2(\theta - 1)}$
Ali–Mikhail–Haq (AMH)	$u_1 u_2 \left\{1 - \theta(1 - u_1)(1 - u_2)\right\}^{-1}$
Clayton	$\left(u_1^{-\theta} + u_2^{-\theta} - 1\right)^{-1/\theta}$
Frank	$-\theta^{-1} \log\left\{1 + \dfrac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{(e^{-\theta} - 1)}\right\}$
Gumbel	$\exp\left[-\left\{(-\log u_1)^\theta + (-\log u_2)^\theta\right\}^{1/\theta}\right]$
Joe	$1 - \left\{(\widetilde{u}_1)^\theta + (\widetilde{u}_2)^\theta - (\widetilde{u}_1 \widetilde{u}_2)^\theta\right\}^{1/\theta}$

NOTES: Let $u_i = F_i(\omega_i)$ be a marginal c.d.f. of a random variable ω_i for i = 1, 2. For Plackett, $r = 1 + (\theta - 1)(u_1 + u_2)$. For Joe, $\widetilde{u}_j = 1 - u_j$.

Besides the dependence pattern, the degree of dependence is important to measure. Even though a dependence parameter θ governs a degree of dependence, it is not di-

rectly comparable across copulas. Along with the estimated θ, it is common to report Kendall's τ as a comparable measure of the degree of dependence. Kendall's τ can be computed as a function of θ, although some copulas have no closed form. In principle, it takes the range of $[-1, 1]$, with the upper (lower) bound corresponding to perfect positive (negative) dependence. A copula that covers this entire range is called comprehensive. Not all copulas are comprehensive, and so their ranges of possible values of τ are narrower than $[-1, 1]$. See table 3 for a feasible range of τ of each copula. Clayton, Gumbel, and Joe copulas allow only positive dependence. Although it is restrictive, changing the sign of the unobservable term enables us to evade this restriction. For example, by letting $\varepsilon_j = -\varepsilon_j^*$ and defining the copula with respect to ε_j^* and ν, we can allow for negative dependence for these copulas.

Table 3. Copula dependence parameter and Kendall's τ

Copula name	Range of θ	θ_{ind}	Kendall's $\tau(\theta)$	Range of τ
Product	N.A.	N.A.	N.A.	N.A.
Gaussian	$-1 \le \theta \le 1$	0	$\frac{2}{\pi}\sin^{-1}(\theta)$	$-1 \le \tau \le 1$
FGM	$-1 \le \theta \le 1$	0	$\frac{2}{9}\theta$	$-\frac{2}{9} \le \tau \le \frac{2}{9}$
Plackett	$0 < \theta < \infty$	1	·	$-1 \le \tau \le 1$
AMH	$-1 \le \theta \le 1$	0	$\left(\frac{3\theta-2}{\theta}\right) -\frac{2}{3}\left(1-\frac{1}{\theta}\right)^2 \ln(1-\theta)$	$-0.1817 \le \tau < \frac{1}{3}$
Clayton	$0 \le \theta < \infty$	0	$\frac{\theta}{\theta+2}$	$0 \le \tau < 1$
Frank	$-\infty < \theta < \infty$	0	$1 - \frac{4}{\theta}\{1 - D_1(\theta)\}$	$-1 < \tau < 1$
Gumbel	$1 \le \theta < \infty$	1	$\frac{\theta-1}{\theta}$	$0 \le \tau < 1$
Joe	$1 \le \theta < \infty$	1	·	$0 \le \tau < 1$

NOTES: θ_{ind} is the value of θ if independent. For Frank, $D_1(\theta)$ is a Debye function: $D_1(\theta) = (1/\theta)\int_0^\theta \{t/(e^t-1)\}dt$. For Plackett and Joe, no closed form is available.

The important value of Kendall's τ is 0. It indicates the independence between the underlying unobservable terms. When independent, the endogeneity issue does not arise in our models, and separate estimation of outcome equations is consistent and efficient. To test the independence, we make use of the fact that the specific value of θ corresponds to $\tau = 0$ (see table 3). A usual hypothesis test can be conducted, such as a likelihood-ratio or Wald test on $\theta = \theta_{\text{ind}}$, and the test statistic is asymptotically distributed as χ^2

under the null. Caution is necessary when a copula is Clayton, Gumbel, or Joe because the independence occurs at the boundary of the parameter space of θ. In this case, the test statistic asymptotically follows a mixture of χ^2 distribution under the null. Our commands report the p-value of the independence test, which takes this fact into account.

To implement maximum likelihood estimation, we need to choose the copula functions C_0 and C_1. If we have ideas on the shapes of the joint distributions of the unobservable terms, it may not be hard to choose an appropriate copula. However, in practice it is unusual to have such information a priori. We usually choose copulas that fit data best. A straightforward way to select copulas is based on information criteria such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC).[3] To allow for possible misspecification of a copula, Trivedi and Zimmer (2007) recommend the "sandwich" variance estimator under the theory of quasilikelihood.

The copula-based maximum-likelihood estimation of the other two models can be conducted basically in the same way. The log-likelihood function of the endogenous dummy model is almost identical to that of the endogenous switching model. However, it has no subscript for an outcome y:

$$\begin{aligned}\ln L = \sum_{i=1}^{N} [&d_i\{y_i \ln \Pr(y_i=1, d_i=1) + (1-y_i) \ln \Pr(y_i=0, d_i=1)\} \\ &+ (1-d_i)\{y_i \ln \Pr(y_i=1, d_i=0) + (1-y_i) \ln \Pr(y_i=0, d_i=0)\}]\end{aligned}$$

Because there are only two unobservable terms, ε and ν, only one copula is considered. The expressions of the joint probabilities in table 1 are also applicable here, although the subscript j does not matter and the vector of explanatory variable x includes a dummy variable d in this model. The log-likelihood function of the sample-selection model is

$$\begin{aligned}\ln L = \sum_{i=1}^{N} [&d_i\{y_i \ln \Pr(y_i=1, d_i=1) + (1-y_i) \ln \Pr(y_i=0, d_i=1)\} \\ &+ (1-d_i) \ln \Pr(d_i=0)]\end{aligned}$$

As with the endogenous dummy variable, one copula function is considered. The expressions of joint probabilities in table 1 are directly applicable by ignoring the subscript j.

4 Treatment effects

Different treatment effects are defined to allow for heterogeneous treatment effects among different populations. The structure of the endogenous switching model enables us to derive various treatment effects straightforwardly. For example, Heckman, Tobias, and Vytlacil (2003) derive various treatment-effect estimators based on the

3. There have been more formal selection procedures proposed in the literature. For example, see Cai and Wang (2014). However, we do not consider these procedures to avoid additional complexity.

switching model with continuous outcomes under the joint normality assumption. When an outcome is binary, the treatment effects can be expressed in terms of marginal or joint probabilities in table 1. As mentioned earlier, the endogenous dummy model can be considered a restricted case of the endogenous switching model. By observing $y_{1i} = 1(\alpha + \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i \geq 0)$ and $y_{0i} = 1(\mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i \geq 0)$, we see the derivation of the treatment effects below also applies to the endogenous dummy model. Note that the treatment effects discussed below are conditional on $\mathbf{x}$ and $\mathbf{z}$.

The ATE is the average effect among the whole population:

$$E(y_1 - y_0) = \Pr(y_1 = 1) - \Pr(y_0 = 1)$$

The ATT is the average effect for those who are actually treated:

$$\begin{aligned} E(y_1 - y_0 | d = 1) &= \Pr(y_1 = 1 | d = 1) - \Pr(y_0 = 1 | d = 1) \\ &= \frac{\Pr(y_1 = 1, d = 1) - \Pr(y_0 = 1, d = 1)}{\Pr(d = 1)} \end{aligned}$$

The ATU is for those who are not treated:

$$\begin{aligned} E(y_1 - y_0 | d = 0) &= \Pr(y_1 = 1 | d = 0) - \Pr(y_0 = 1 | d = 0) \\ &= \frac{\Pr(y_1 = 1, d = 0) - \Pr(y_0 = 1, d = 0)}{\Pr(d = 0)} \end{aligned}$$

The LATE is the treatment effect for those who change the treatment status in response to a change in an instrument variable. To define LATE, suppose that a value of the kth variable of $\mathbf{z}$, which is excluded from $\mathbf{x}$, z_k changes from the lower value $z_{k,L}$ to the upper value $z_{k,U}$, and let $d(z_k)$ be the treatment indicator with the value of z_k. Then LATE is

$$\begin{aligned} &E\{y_1 - y_0 | d(z_{k,U}) = 1, d(z_{k,L}) = 0\} \\ &\quad = \Pr\{y_1 = 1 | d(z_{k,U}) = 1, d(z_{k,L}) = 0\} - \Pr\{y_0 = 1 | d(z_{k,U}) = 1, d(z_{k,L}) = 0\} \\ &\quad = \frac{\Pr\{y_1 = 1, d(z_{k,U}) = 1\} - \Pr\{y_1 = 1, d(z_{k,L}) = 1\}}{\Pr\{d(z_{k,U}) = 1\} - \Pr\{d(z_{k,L}) = 1\}} \\ &\qquad - \frac{\Pr\{y_0 = 1, d(z_{k,U}) = 1\} - \Pr\{y_0 = 1, d(z_{k,L}) = 1\}}{\Pr\{d(z_{k,U}) = 1\} - \Pr\{d(z_{k,L}) = 1\}} \end{aligned}$$

Heckman and Vytlacil (2005) show that these treatment effects are weighted averages of the MTE. See Heckman and Vytlacil (2005, 2007) for further discussion. MTE is the treatment effect for those with a particular value of ν, which is

$$E\left(y_1 - y_0 | \nu = \widetilde{\nu}\right) = \Pr\left(y_1 = 1 | \nu = \widetilde{\nu}\right) - \Pr\left(y_0 = 1 | \nu = \widetilde{\nu}\right)$$

Note that when ν enters in the selection with a positive (negative) sign, the larger (smaller) $\widetilde{\nu}$ is, and the more likely an individual is to receive the treatment. ν is often normalized as $\nu^* = F_d(\nu)$ so that $0 < \nu^* < 1$. A larger (smaller) value of ν^* indicates

a higher propensity to receive the treatment when the sign of ν is positive (negative) in the selection equation. $\Pr(y_j = 1|\nu = \widetilde{\nu})$ can be expressed as the derivative of a copula function. That is, the shapes of MTE differ by the dependence structure of the unobservable terms. The simulation study by Hasebe (2021) shows the misspecification of copula results in the biased estimation of MTE and other treatment effects.

Although it is implicit, the treatment effects discussed above are conditional on $\mathbf{x}$ and $\mathbf{z}$. The unconditional treatment effects are estimated by evaluating the treatment effects for each observation and then by averaging relevant samples. ATE, LATE, and MTE are estimated by averaging over the whole sample. ATT and ATU are the averages of the subsamples of those with $d_i = 1$ and those with $d_i = 0$, respectively.

The asymptotic variance of the conditional version of the treatment effects can be obtained by applying the delta method. When estimating the asymptotic variance of the unconditional version of the treatment-effect estimators, we additionally consider the sampling variability from the randomness of covariates $\mathbf{x}_i$ and $\mathbf{z}_i$. Our commands for the estimation of the unconditional treatment effects report the standard errors taking this variability into consideration. The derivation of the asymptotic variance is based on Newey and McFadden (1994). See also Terza (2016) for the correct asymptotic variance of sample means of nonlinear transformations such as the treatment-effect estimators discussed here.

5 The commands

This section describes the commands `esbinary`, `edbinary`, and `ssbinary`, which implement copula-based maximum-likelihood estimation of the models with a binary choice outcome. These commands utilize the Stata command `ml`. This section also describes the postestimation command `estat teffects`, which estimates the treatment effects after the execution of `esbinary` and `edbinary`.

5.1 esbinary

5.1.1 Syntax

The syntax for the command is as follows:

```
esbinary (depvar0 [=] varlist0) [(depvar1 [=] varlist1)] [if] [in] [weight],
    select(depvars = varlists[, noconstant offset(varnameo)])
    [copula0(copula) copula1(copula) margsel(margin) margin0(margin)
    margin1(margin) negatives negative0 negative1 consel noconstant
    offset(varnameo) constraints(constraints) vce(vcetype) maximize_options]
```

aweights, fweights, iweights, and pweights can be used depending on the methods chosen; see [U] **11.1.6 weight**.

When dependent variables and sets of explanatory variables are the same across regimes, you need to specify only one equation. If not, you need to specify two equations separately, with each equation enclosed by parentheses. In this case, the first equation is for regime 0 and the second equation is for regime 1.

5.1.2 Options

`select(`*depvar*$_s$ `=` *varlist*$_s$[`, noconstant offset(`*varname*$_o$`)`]`)` specifies the variables and options for the selection equation. *depvar*$_s$ should be coded as `0` or `1`, with `0` indicating regime 0 and `1` indicating regime 1. `select()` is required.

`copula0(`*copula*`)` specifies a copula function for the dependence between the error terms in the selection equation and regime 0, ν and ε_0, which must be one of the list in table 2. Available copulas are `product`, `gaussian`, `fgm`, `plackett`, `amh`, `clayton`, `frank`, `gumbel`, and `joe`. The default is `copula0(gaussian)`. Note that the name of *copula* is case sensitive, so all letters in the name should be typed in lowercase, for example, `copula0(gaussian)`. Except for `copula0(product)`, the results table displays estimates of an auxiliary dependence parameter (`atheta0`) and a dependence parameter (`theta0`). For copulas for which Kendall's τ can be calculated analytically, as in table 3, the results table reports the estimate of τ_0 `tau0`.

`copula1(`*copula*`)` specifies a copula function for the dependence between the error terms in the selection equation and regime 1, ν and ε_1, which must be one of the list in table 2. Available copulas are `product`, `gaussian`, `fgm`, `plackett`, `amh`, `clayton`, `frank`, `gumbel`, and `joe`. The default is `copula1(gaussian)`. Note that the name of *copula* is case sensitive, so all letters in the name should be typed in lowercase, for example, `copula1(gaussian)`. Except for `copula1(product)`, the results table displays estimates of an auxiliary dependence parameter (`atheta1`) and a dependence parameter (`theta1`). For copulas for which Kendall's τ can be calculated analytically, as in table 3, the results table reports the estimate of τ_1 `tau1`.

`margsel(`*margin*`)` specifies the marginal distribution of the error term in the selection equation, $F_d(\nu)$. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margsel(probit)`.

`margin0(`*margin*`)` specifies the marginal distribution of the error term in regime 0, $F_0(\varepsilon_0)$. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margin0(probit)`.

`margin1(`*margin*`)` specifies the marginal distribution of the error term in regime 1, $F_1(\varepsilon_1)$. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margin1(probit)`.

`negatives` makes the error term of the selection equation negative.

`negative0` makes the error term of regime 0 negative.

`negative1` makes the error term of regime 1 negative.

`consel` allows contributions to the likelihood of the selection equation by observations whose selection decision is observed but whose outcome variables or some of the covariates in the outcome equations are not observed.

`noconstant` suppresses the constant term of the outcome equation.

`offset(`*varname*$_o$`)` includes *varname*$_o$ in the model with the coefficient constrained to 1.

`constraints(`*constraints*`)`; see [R] **Estimation options**.

`vce(`*vcetype*`)` specifies the type of the standard errors reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that are derived from asymptotic theory (`oim`, `opg`); see [R] ***vce_option***.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **Maximize**. These options are seldom used.

5.1.3 Stored results

Results that are stored by `esbinary`, `edbinary`, and `ssbinary` in common are listed at the end of this section. The results stored in `e()` specific to `esbinary` are as follows:

Scalars

`e(negatives)`	`1` if the option `negatives` is specified, `0` otherwise
`e(negative0)`	`1` if the option `negative0` is specified, `0` otherwise
`e(negative1)`	`1` if the option `negative1` is specified, `0` otherwise

Macros

`e(cmd)`	`esbinary`
`e(estat_cmd)`	program used to implement `estat`
`e(copula0)`	specified `copula0()`
`e(copula1)`	specified `copula1()`
`e(margsel)`	specified `margsel()`
`e(margin0)`	specified `margin0()`
`e(margin1)`	specified `margin1()`

5.1.4 Postestimation statistics

After an execution of `esbinary`, the `estat teffects` command is available to estimate the treatment effects as postestimation statistics. `estat teffects` has the following syntax:

`estat teffects [, ate att atu late(`*varname lower upper*`) mte(`*nu*`)]`

This command estimates the unconditional treatment effects and their standard errors using the estimated parameters of the endogenous switching model.

The options for `estat teffects` are as follows:

`ate` estimates ATE. This is the default.

`att` estimates ATT.

`atu` estimates ATU.

`late(`*varname lower upper*`)` estimates LATE when the value of *varname* (z_k) changes from *lower* ($z_{k,L}$) to *upper* ($z_{k,U}$). *lower* and *upper* should be numerical.

`mte(`*nu*`)` estimates MTE evaluated at *nu* (ν). *nu* should be numerical.

`estat teffects` stores the following in `r()`:

Matrix
`r(table)` table of result

The matrix `r(table)` contains the estimated treatment effect in the first column, its standard error in the second column, the z-value in the third column, and the p-value in the fourth column. When only one option is specified, this matrix is a row vector. When more than one option is specified, the rows of this matrix correspond to the treatment effects that are specified as options.

5.1.5 Prediction

After an execution of `esbinary`, the `predict` command is available to compute several predictions for each observation. `predict` has the following syntax:

`predict` [*type*] *newvar* [*if*] [*in*] [`, psel xb0 xb1 xbsel prob0 prob1 prob00 prob11 prob01 prob10 cll te tt tu late(`*varname lower upper*`) mte(`*nu*`)`]

The options for `predict` are as follows:

`psel` calculates the probability that the treatment status of the selection variable is 1: $\Pr(d_i = 1)$. This is the default.

`xb0` calculates the linear prediction of the outcome variable of regime 0: $\mathbf{x}_i'\boldsymbol{\beta}_0$.

`xb1` calculates the linear prediction of the outcome variable of regime 1: $\mathbf{x}_i'\boldsymbol{\beta}_1$.

`xbsel` calculates the linear prediction of the selection variable: $\mathbf{z}_i'\boldsymbol{\gamma}$.

`prob0` calculates the probability that the outcome variable of regime 0 is equal to 1: $\Pr(y_{0i} = 1)$.

`prob1` calculates the probability that the outcome variable of regime 1 is equal to 1: $\Pr(y_{1i} = 1)$.

`prob00` calculates the joint probability that the outcome variable of regime 0 is equal to 1 when the treatment status is 0: $\Pr(y_{0i} = 1, d_i = 0)$.

`prob11` calculates the joint probability that the outcome variable of regime 1 is equal to 1 when the treatment status is 1: $\Pr(y_{1i} = 1, d_i = 1)$.

`prob01` calculates the hypothetical joint probability that the outcome variable of regime 0 is equal to 1 when the treatment status is 1: $\Pr(y_{0i} = 1, d_i = 1)$.

`prob10` calculates the hypothetical joint probability that the outcome variable of regime 1 is equal to 1 when the treatment status is 0: $\Pr(y_{1i} = 1, d_i = 0)$.

`cll` computes the contribution to the log-likelihood function by each observation.

`te` computes the observation-level treatment effect: $E(y_{1i} - y_{0i}|\mathbf{x}_i)$.

`tt` computes the observation-level treatment effect on the treated: $E(y_{1i} - y_{0i}|d_i = 1, \mathbf{x}_i, \mathbf{z}_i)$.

`tu` computes the observation-level treatment effect on the untreated: $E(y_{1i} - y_{0i}|d_i = 0, \mathbf{x}_i, \mathbf{z}_i)$.

`late(`*varname lower upper*`)` computes the observation-level LATE when the value of *varname* (z_k) changes from *lower* ($z_{k,L}$) to *upper* ($z_{k,U}$): $E\{y_{1i} - y_{0i}|d(z_{k,U}) = 1, d(z_{k,L}) = 0, \mathbf{x}_i, \mathbf{z}_i\}$.

`mte(`*nu*`)` computes the observational-level MTE evaluated at *nu* (ν): $E(y_{1i} - y_{0i}|\nu = \nu, \mathbf{x}_i)$.

5.2 edbinary

5.2.1 Syntax

The syntax for `edbinary` is as follows:

`edbinary` *depvar* [`=`] *indepvars* [*if*] [*in*] [*weight*]`,`
`select(`*depvar*$_s$ `=` *varlist*$_s$[`, noconstant offset(`*varname*$_o$`)`]`)`
[`copula(`*copula*`) margsel(`*margin*`) margout(`*margin*`) negatives negativeo`
`consel noconstant offset(`*varname*$_o$`) constraints(`*constraints*`) vce(`*vcetype*`)`
maximize_options]

`aweight`s, `fweight`s, `iweight`s, and `pweight`s can be used depending on the methods chosen; see [U] **11.1.6 weight**.

5.2.2 Options

`select(`*depvar*$_s$ `=` *varlist*$_s$[`, noconstant offset(`*varname*$_o$`)`]`)` specifies the variables and options for the selection equation. *depvar*$_s$ should be coded as `0` or `1`, with `0` indicating regime 0 and `1` indicating regime 1. `select()` is required.

`copula(`*copula*`)` specifies a copula function for the dependence between the error terms in the selection equation and outcome equations, which must be one of the list in table 2. Available copulas are `product`, `gaussian`, `fgm`, `plackett`, `amh`, `clayton`, `frank`, `gumbel`, and `joe`. The default is `copula(gaussian)`. Note that the name of *copula* is case sensitive, so all letters in the name should be typed in lowercase, for example, `copula(gaussian)`. Except for `copula(product)`, the results table displays estimates of an auxiliary dependence parameter (`atheta`) and a dependence parameter (`theta`). For copulas for which Kendall's τ can be calculated analytically, as in table 3, the results table reports the estimate of τ.

`margsel(`*margin*`)` specifies the marginal distribution of the error term in the selection equation. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margsel(probit)`.

`margout(`*margin*`)` specifies the marginal distribution of the error term in the outcome equation. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margout(probit)`.

`negatives` makes the error term of the selection equation negative.

`negativeo` makes the error term of the outcome equation negative.

`consel` allows contributions to the likelihood of selection equation by observations whose selection decision is observed but whose outcome variables or some of the covariates in the outcome equations are not observed.

`noconstant` suppresses the constant term of the outcome equation.

`offset(`$varname_o$`)` includes $varname_o$ in the model with the coefficient constrained to 1.

`constraints(`*constraints*`)`; see [R] **Estimation options**.

`vce(`*vcetype*`)` specifies the type of the standard errors reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that are derived from asymptotic theory (`oim`, `opg`); see [R] ***vce_option***.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **Maximize**. These options are seldom used.

5.2.3 Stored results

The results stored in `e()` specific to `edbinary` are as follows:

Scalars

`e(negatives)`	`1` if the option `negatives` is specified, `0` otherwise
`e(negativeo)`	`1` if the option `negativeo` is specified, `0` otherwise

Macros

`e(cmd)`	`edbinary`
`e(estat_cmd)`	program used to implement `estat`
`e(copula)`	specified `copula()`
`e(margsel)`	specified `margsel()`
`e(margout)`	specified `margout()`

5.3 Postestimation statistics

After an execution of `edbinary`, the `estat teffects` command is available to estimate the treatment effects as postestimation statistics. `estat teffects` has the following syntax:

`estat teffects` [`, ate att atu late(`*varname lower upper*`) mte(`*nu*`)`]

This command estimates the unconditional treatment effects and their standard errors using the estimated parameters of the endogenous dummy model.

5.3.1 Options

The options for `estat teffects` are as follows:

`ate` estimates ATE. This is the default.

`att` estimates ATT.

`atu` estimates ATU.

`late(`*varname lower upper*`)` estimates LATE when the value of *varname* (z_k) changes from *lower* ($z_{k,L}$) to *upper* ($z_{k,U}$). *lower* and *upper* should be numerical.

`mte(`*nu*`)` estimates MTE evaluated at *nu* (ν). *nu* should be numerical.

5.3.2 Stored results

`estat teffects` stores the following in `r()`:

Matrix

`r(table)`	table of result

The matrix `r(table)` contains the estimated treatment effect in the first column, its standard error in the second column, the z-value in the third column, and the p-value in the fourth column. When only one option is specified, this matrix is a row vector. When more than one option is specified, the rows of this matrix correspond to the treatment effects that are specified as options.

5.3.3 Prediction

After an execution of `edbinary`, the `predict` command is available to compute several predictions for each observation. `predict` has the following syntax:

`predict` [*type*] *newvar* [*if*] [*in*] [`, psel pout xbo xbsel prob0 prob1 prob00 prob11 prob01 prob10 cll te tt tu late(`*varname lower upper*`) mte(`*nu*`)`]

The options for `predict` are as follows:

`psel` calculates the probability that the treatment status of the selection variable is 1: $\Pr(d_i = 1)$. This is the default.

`pout` calculates the probability that the treatment status of the outcome variable is 1: $\Pr(y_i = 1)$.

`xbo` calculates the linear prediction of the outcome variable: $\alpha d_i + \mathbf{x}_i'\boldsymbol{\beta}$.

`xbsel` calculates the linear prediction of the selection variable: $\mathbf{z}_i'\boldsymbol{\gamma}$.

`prob0` calculates the probability that the outcome variable is equal to 1 for the treatment status $d = 0$: $\Pr(y_{0i} = 1)$.

`prob1` calculates the probability that the outcome variable is equal to 1 for the treatment status $d = 1$: $\Pr(y_{1i} = 1)$.

`prob00` calculates the joint probability that the outcome variable is equal to 1 and the treatment status is 0 when the treatment dummy variable takes the value 0 in the outcome equation: $\Pr(y_{0i} = 1, d_i = 0)$.

`prob11` calculates the joint probability that the outcome variable is equal to 1 and the treatment status is 1 when the treatment dummy variable takes the value 1 in the outcome equation: $\Pr(y_{1i} = 1, d_i = 1)$.

`prob01` calculates the hypothetical joint probability that the outcome variable is equal to 1 and the treatment status is 1 when the treatment dummy variable takes the value 0 in the outcome equation: $\Pr(y_{0i} = 1, d_i = 1)$.

`prob10` calculates the hypothetical joint probability that the outcome variable is equal to 1 and the treatment status is 0 when the treatment dummy variable takes the value 1 in the outcome equation: $\Pr(y_{1i} = 1, d_i = 0)$.

`cll` computes the contribution to the log-likelihood function by each observation.

`te` computes the observation-level treatment effect: $E(y_{1i} - y_{0i}|\mathbf{x}_i)$.

`tt` computes the observation-level treatment effect on the treated: $E(y_{1i} - y_{0i}|d_i = 1, \mathbf{x}_i, \mathbf{z}_i)$.

`tu` computes the observation-level treatment effect on the untreated: $E(y_{1i} - y_{0i}|d_i = 0, \mathbf{x}_i, \mathbf{z}_i)$.

`late(`*varname lower upper*`)` computes the observation-level LATE when the value of *varname* (z_k) changes from *lower* ($z_{k,L}$) to *upper* ($z_{k,U}$): $E\{y_{1i} - y_{0i}|d(z_{k,U}) = 1, d(z_{k,L}) = 0, \mathbf{x}_i, \mathbf{z}_i\}$.

`mte(`*nu*`)` computes the observational-level MTE evaluated at *nu* (ν): $E(y_{1i} - y_{0i}|\nu = \nu, \mathbf{x}_i)$.

5.4 ssbinary

5.4.1 Syntax

The syntax for the command is as follows:

```
ssbinary depvar [=] indepvars [if] [in] [weight],
    select(depvar_s = varlist_s[, noconstant offset(varname_o)])
    [copula(copula) margsel(margin) margout(margin) negatives negativeo
    noconstant offset(varname_o) constraints(constraints) vce(vcetype)
    maximize_options]
```

`aweights`, `fweights`, `iweights`, and `pweights` can be used depending on the methods chosen; see [U] **11.1.6 weight**.

5.4.2 Options

`select(`*depvar*$_s$ `=` *varlist*$_s$`[, noconstant offset(`*varname*$_o$`)])` specifies the variables and options for the selection equation. *depvar*$_s$ should be coded as `0` or `1`, with `0` indicating regime 0 and `1` indicating regime 1. `select()` is required.

`copula(`*copula*`)` specifies a copula function governing the dependence between the errors in the outcome equation and selection equation, which must be one of the list in table 2. Available copulas are `product`, `gaussian`, `fgm`, `plackett`, `amh`, `clayton`, `frank`, `gumbel`, and `joe`. The default is `copula(gaussian)`. Note that the name of *copula* is case sensitive, so all letters in the name should be typed in lowercase, for example, `copula(gaussian)`. The results table reports the estimate of the dependence parameter θ `theta` (and an ancillary parameter `atheta`). For copulas for which Kendall's τ can be calculated analytically, as in table 3, the results table reports the estimate of τ.

`margsel(`*margin*`)` specifies the marginal distribution of the error term in the selection equation. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margsel(probit)`.

`margout(`*margin*`)` specifies the marginal distribution of the error term in the outcome equation. *margin* may be `probit` (or `normal`) or `logit` (or `logistic`). The default is `margout(probit)`.

`negatives` makes the error term of the selection equation negative.

`negativeo` makes the error term of the outcome equation negative.

`noconstant` suppresses the constant term of the outcome equation.

`offset(`*varname*$_o$`)` includes *varname*$_o$ in the model with the coefficient constrained to 1.

`constraints(`*constraints*`)`; see [R] **Estimation options**.

`vce(`*vcetype*`)` specifies the type of the standard errors reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that are derived from asymptotic theory (`oim`, `opg`); see [R] ***vce_option***.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(`#`)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(`#`)`, `ltolerance(`#`)`, `nrtolerance(`#`)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **Maximize**. These options are seldom used.

5.4.3 Stored results

The results stored in `e()` specific to `ssbinary` are as follows:

Scalars

`e(negatives)`	`1` if the option `negatives` is specified, `0` otherwise
`e(negativeo)`	`1` if the option `negativeo` is specified, `0` otherwise

Macros

`e(cmd)`	`ssbinary`
`e(copula)`	specified `copula()`
`e(margsel)`	specified `margsel()`
`e(margout)`	specified `margout()`

5.4.4 Prediction

After an execution of `ssbinary`, the `predict` command is available to compute several predictions for each observation. `predict` has the following syntax:

`predict` `[`*type*`]` *newvar* `[`*if*`]` `[`*in*`]` `[, pmargin p11 p10 p01 p00 psel pcond xb xbsel cll]`

The options for `predict` are as follows:

`pmargin` calculates the probability that the outcome is 1: $\Pr(y_i = 1)$. This is the default.

`p11` calculates $\Pr(y_i = 1, d_i = 1)$.

`p10` calculates $\Pr(y_i = 1, d_i = 0)$.

`p01` calculates $\Pr(y_i = 0, d_i = 1)$.

`p00` calculates $\Pr(y_i = 0, d_i = 0)$.

`psel` calculates the probability that the outcome is observed: $\Pr(d_i = 1)$.

`pcond` calculates $\Pr(y_i = 1 | d_i = 1)$.

`xb` calculates the linear prediction of the outcome variable: $\mathbf{x}_i'\boldsymbol{\beta}$.

`xbsel` calculates the linear prediction of the selection variable: $\boldsymbol{z}_i'\boldsymbol{\gamma}$.

`cll` computes the contribution to the log-likelihood function by each observation.

5.5 Notes

5.5.1 Stored results

Results that are stored after the execution of `esbinary`, `edbinary`, and `ssbinary` in common are as follows:

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_aux)`	number of auxiliary parameters
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(chi2)`	χ^2
`e(p)`	significance of model test
`e(rank)`	rank of `e(V)`
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged; `0` otherwise
`e(ll0)`	log likelihood, independent model
`e(AIC)`	AIC
`e(BIC)`	BIC

Macros

`e(depvar)`	names of dependent variables
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset1)`	offset for outcome equation
`e(offset2)`	offset for selection equation
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`

Matrices

`e(b)`	coefficient vector
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

5.5.2 Ancillary parameter

In the maximum likelihood estimation, θ is not directly estimated. Instead, an ancillary dependence parameter θ^*, which takes any real value, is directly estimated, and it is transformed to the dependence parameter so that it takes a feasible value, as in table 3. The way to transform differs by copula:

$$\theta = \begin{cases} (e^{\theta^*} - e^{-\theta^*})/(e^{\theta^*} + e^{-\theta^*}) & \text{(Gaussian, FGM, AMH)} \\ e^{\theta^*} & \text{(Plackett, Clayton)} \\ 1 + e^{\theta^*} & \text{(Gumbel, Joe)} \\ \theta^* & \text{(Frank)} \end{cases}$$

The ancillary dependence parameter and transformed dependence parameter are reported as `atheta` and `theta`, respectively.

6 Examples

To illustrate the use of the commands, we show two sets of examples.

6.1 Example 1

The first set of examples uses a dataset from the Stata website. This dataset is the one used for the example of `heckprobit` and `biprobit`. Assuming joint normality, our commands with the default options yield the same results that the Stata commands do.

First, we fit the sample-selection binary model. The Stata command `heckprobit` fits this model under the joint normality assumption.

```
. webuse school
. heckprobit private years logptax, select(vote = years loginc logptax)
  (output omitted)
Iteration 5:   log likelihood = -74.244973
Probit model with sample selection              Number of obs     =         95
                                                      Selected    =         59
                                                      Nonselected =         36
                                                Wald chi2(2)      =       1.04
Log likelihood = -74.24497                      Prob > chi2       =     0.5935
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
private						
years	-.1142596	.1461715	-0.78	0.434	-.4007505	.1722313
logptax	.3516101	1.016483	0.35	0.729	-1.64066	2.34388
_cons	-2.780667	6.905827	-0.40	0.687	-16.31584	10.75451
vote						
years	-.0167511	.0147735	-1.13	0.257	-.0457067	.0122045
loginc	.9923023	.4430008	2.24	0.025	.1240368	1.860568
logptax	-1.278783	.5717545	-2.24	0.025	-2.399401	-.1581646
_cons	-.5458205	4.070417	-0.13	0.893	-8.523692	7.432051
/athrho	-.8663164	1.450017	-0.60	0.550	-3.708298	1.975665
rho	-.6994978	.7405281			-.9987983	.9622674

```
LR test of indep. eqns. (rho = 0): chi2(1) = 0.27         Prob > chi2 = 0.6020
```

The command **ssbinary** with the default specifications yields the same result, although the results are not exactly identical because of a nonlinear optimization process.

```
. ssbinary private years logptax, select(vote = years loginc logptax)
  (output omitted)
Iteration 5:   log likelihood = -74.244973
Sample Selection Binary Outcome Model: gaussian-probit-probit
                                                  Number of obs =     95
                                                  Wald chi2(2)  =   1.04
Log likelihood = -74.244973                       Prob > chi2   = 0.5935
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
private						
years	-.1142596	.1461715	-0.78	0.434	-.4007505	.1722313
logptax	.3516101	1.016483	0.35	0.729	-1.64066	2.34388
_cons	-2.780667	6.905827	-0.40	0.687	-16.31584	10.75451
select						
years	-.0167511	.0147735	-1.13	0.257	-.0457067	.0122045
loginc	.9923023	.4430008	2.24	0.025	.1240368	1.860568
logptax	-1.278783	.5717545	-2.24	0.025	-2.399401	-.1581646
_cons	-.5458205	4.070417	-0.13	0.893	-8.523692	7.432051
/atheta	-.8663165	1.450017	-0.60	0.550	-3.708298	1.975665
theta	-.6994978	.7405279			-.9987983	.9622674
tau	-.4931859	.6596862			-.9687874	.8245599

LR test of independence : Test statistic 0.272 with p-value 0.6020

Next, we fit the endogenous dummy model. The Stata command `biprobit` fits this model under the joint normality assumption, and the command `edbinary` yields the same result with the default specifications.

```
. biprobit (private = vote logptax loginc years) (vote = logptax loginc years)
  (output omitted)
. * Our command edbinary yields the same result.
. edbinary  private = logptax loginc years, select(vote = logptax loginc years)
  (output omitted)
Iteration 7:   log likelihood = -89.098442
Bivariate Binary Outcome Model: gaussian-probit-probit
                                                  Number of obs =     95
                                                  Wald chi2(4)  =   4.14
Log likelihood = -89.098442                       Prob > chi2   = 0.3870
------------------------------------------------------------------------------
             |  Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
private      |
        vote |   1.047535   .6753432     1.55   0.121    -.2761137    2.371183
     logptax |   .4241346   .7265603     0.58   0.559    -.9998975    1.848167
      loginc |  -.0962434   .5746769    -0.17   0.867    -1.222589    1.030103
       years |  -.0034985   .0240795    -0.15   0.884    -.0506935    .0436965
       _cons |   -3.56953   4.262826    -0.84   0.402    -11.92452    4.785456
-------------+----------------------------------------------------------------
vote         |
     logptax |  -1.327977   .6061065    -2.19   0.028    -2.515923   -.1400298
      loginc |   .9801528   .4390441     2.23   0.026     .1196422    1.840663
       years |   -.019161    .015062    -1.27   0.203    -.0486819    .0103599
       _cons |  -.0562336   4.336532    -0.01   0.990     -8.55568    8.443213
-------------+----------------------------------------------------------------
     /atheta |  -1.293455   1.248206    -1.04   0.300    -3.739895    1.152984
-------------+----------------------------------------------------------------
       theta |  -.8600287   .3249712                     -.9988719    .8187404
         tau |   -.659109   .4054579                     -.9697578    .6106544
------------------------------------------------------------------------------
LR test of independence :        Test statistic    0.399 with p-value  0.5277
```

As described in the previous section, some treatment effects are estimated using the postestimation command `estat teffects` after `edbinary`. For example, we estimate ATE, ATT, and ATU using the options `ate`, `att`, and `atu`, respectively.

```
. estat teffects, ate att atu
Treatment Effects:
     Estimate        SE         z     P>|z|
ATE    .23215    .20633    1.1251    .26053
ATT    .08399     .0393    2.1371    .03259
ATU    .48042    .52679    .91197    .36178
```

The `predict` command has the options to compute the observation-level treatment effects. Taking the average of the predicted values over the relevant samples, we obtain the unconditional treatment effects that are the same as the results from `estat teffects`.

```
. predict te, te
. predict tt, tt
. predict tu, tu
. * ATE
. summarize te
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
          te |         95    .2321475    .0299214   .1400827    .301635
. * ATT
. summarize tt if vote == 1
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
          tt |         59    .0839859    .0377103   .0008582   .1777933
. * ATU
. summarize tu if vote == 0
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
          tu |         36    .4804196    .0740163   .3137575   .5859616
```

The last example is the estimation of the endogenous switching model. Although there is no Stata command to fit this model, the command `switch_probit`, written by Lokshin and Sajaia (2011), fits it under the joint normality assumption.

```
. * The community-contributed command (install it if not installed).
. switch_probit private logptax loginc years, select(vote = logptax loginc years)
```

(*output omitted*)

```
. * Our command esbinary yields the same result.
. esbinary (private logptax loginc years), select(vote = logptax loginc years)
> difficult
```

(*output omitted*)

```
Iteration 14:  log likelihood = -87.605063

Switching Binary Outcome Model: Copula: gaussian-gaussian &  Margin:
> probit-probit-probit
                                                  Number of obs =     95
                                                  Wald chi2(3)  =   0.30
Log likelihood = -87.605063                       Prob > chi2   = 0.9592
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
regime0						
logptax	-.1247044	.9198479	-0.14	0.892	-1.927573	1.678164
loginc	.261568	.6560623	0.40	0.690	-1.02429	1.547427
years	.010009	.0273661	0.37	0.715	-.0436276	.0636456
_cons	-3.449124	5.164728	-0.67	0.504	-13.57181	6.673557
regime1						
logptax	.9175379	.7580993	1.21	0.226	-.5683094	2.403385
loginc	-.7603868	.5244297	-1.45	0.147	-1.78825	.2674765
years	-.0491842	.1278883	-0.38	0.701	-.2998406	.2014722
_cons	1.153637	6.37084	0.18	0.856	-11.33298	13.64025
selection						
logptax	-1.404702	.6504772	-2.16	0.031	-2.679614	-.1297902
loginc	.9819269	.4351344	2.26	0.024	.1290792	1.834775
years	-.0201954	.0152615	-1.32	0.186	-.0501074	.0097166
_cons	.463035	4.682142	0.10	0.921	-8.713795	9.639865
/atheta0	-2.689369	491.0011	-0.01	0.996	-965.0339	959.6552
/atheta1	-5.519322	133.8555	-0.04	0.967	-267.8713	256.8327
theta0	-.9908151	8.978176			-1	1
theta1	-.9999679	.0086032			-1	1
tau0	-.9136494	42.26836			-1	1
tau1	-.9948962	.6831694			-1	1

LR test of independence : Test statistic 1.196 with p-value 0.5499

Note that the maximization option `difficult` often helps the program converge. There are some occasions when the log-likelihood function does not converge without this option.

As with the command `edbinary`, using `estat teffects` after `esbinary` estimates the treatment effects.

```
. estat teffects, ate att atu
Treatment Effects:
     Estimate        SE         z     P>|z|
ATE    .28671    .13993     2.049    .04047
ATT    .08233    .13299    .61912    .53584
ATU    .61656    .47186    1.3067    .19133
```

Because the model specifications are different, the treatment-effect estimates of `esbinary` are different from those of `edbinary` reported above.

6.2 Example 2

This application is from Winkelmann (2012), which is originally based on Deb, Munkin, and Trivedi (2006).[4] It examines if the effect of an enrollment in a health maintenance organization (`HMO`) on the outcome of interest is whether an individual has any positive ambulatory heath expenditure (`PAMBEXP`). Explanatory variables include socioeconomic characteristics of each individual. Two variables are excluded from the outcome equations but included in the selection equation. These instrumental variables are the age of spouse (`SPAGE`) and whether the spouse enrolled in a health maintenance organization in the previous year (`LAGSPHMO`).

```
. clear
. infile AMBEXP LNAMBEXP DAMBEXP HOSPEXP LNHOSPEXP DHOSPEXP PPO HMO FFS FAMSIZE
> AGE EDUC INCOME FEMALE BLACK HISPANIC MARRIED NOREAST MIDWEST SOUTH MSA AGE2
> AGEXFEM VEGOOD GOOD FAIRPOOR PHYSLIM TOTCHR INJURY SPAGE LAGSPPPO LAGSPHMO
> YEAR98 YEAR99 YEAR00 YEAR01 using "meps96_01.txt"
(20,460 observations read)
. keep if FFS != 1                // drop all individuals enrolled in a
> fee-for-service plan
(8,078 observations deleted)
. * outcome variable
. generate PAMBEXP = (AMBEXP > 0) // 1 if ambulatory expediture is positive
. * setting locals
. local Y PAMBEXP
. local S HMO
. local Zlist FAMSIZE EDUC INCOME FEMALE BLACK HISPANIC MARRIED NOREAST MIDWEST
> SOUTH MSA AGE AGE2 AGEXFEM VEGOOD GOOD FAIRPOOR PHYSLIM TOTCHR INJURY YEAR98
> YEAR99 YEAR00 YEAR01 SPAGE LAGSPHMO
. local Xlist FAMSIZE EDUC INCOME FEMALE BLACK HISPANIC MARRIED NOREAST MIDWEST
> SOUTH MSA AGE AGE2 AGEXFEM VEGOOD GOOD FAIRPOOR PHYSLIM TOTCHR INJURY YEAR98
> YEAR99 YEAR00 YEAR01
```

First, we fit the endogenous switching model by the command `esbinary`. In this example, we do not have a particular idea about the dependence structure of the unobservable terms. We choose the best-fitting combination of copulas that attains the smallest value of the AIC or BIC. We allow for different copulas for C_0 and C_1. The copulas to be considered are Gaussian, Frank, Clayton, Gumbel, and Joe. For the last three copulas, we consider the negative version as well. We fix the standard normal as marginal distributions. The number of parameters to estimate is the same across all copula models. Therefore, choosing the minimum AIC or BIC is equivalent to choosing the largest log likelihood. We find the best-fitting combination of copulas by using loops.

4. The data are obtained from http://qed.econ.queensu.ca/jae/datasets/deb003/.

```
. * list of copulas
. local copula_list gaussian frank clayton gumbel joe nclayton ngumbel njoe
. local llmax = -999999999
. foreach copula0 of local copula_list {
  2.     foreach copula1 of local copula_list {
  3.        foreach j in 0 1 {
  4.           if "`copula`j''"=="nclayton" | "`copula`j''"=="ngumbel" |
> "`copula`j''"=="njoe" {
  5.               local negative`j' negative`j'
  6.               if "`copula`j''"=="nclayton" local cname`j' clayton
  7.               if "`copula`j''"=="ngumbel" local cname`j' gumbel
  8.               if "`copula`j''"=="njoe" local cname`j' joe
  9.           }
 10.           else {
 11.               local negative`j'
 12.               local cname`j' `copula`j''
 13.            }
 14.        }
 15.     quietly esbinary (`Y' `Xlist'), select(`S' = `Zlist') difficult
> copula0(`cname0') copula1(`cname1') `negative0' `negative1' iterate(50)
> vce(robust)
 16.     if `e(ll)' > `llmax' {
 17.         local llmax = `e(ll)'
 18.         estimates store best_ll_model
 19.     }
 20.     }
 21. }
```

We store the estimation result of the largest log-likelihood combination of copulas under the name of `best_ll_model`. After the loop, we display the stored result. To save space, we show the estimated coefficients for only selected explanatory variables.

```
. estimates restore best_ll_model
(results best_ll_model are active now)
```

```
. estimates replay
```

Model best_ll_model

```
Switching Binary Outcome Model: Copula: joe-n-clayton &  Margin:
> probit-probit-probit
                                                  Number of obs = 12,382
                                                  Wald chi2(24) = 115.78
Log pseudolikelihood = -8328.6889                 Prob > chi2   = 0.0000
```

	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
regime0						
FAMSIZE	-.0795145	.0268144	-2.97	0.003	-.1320697	-.0269594
EDUC	.0345774	.0167232	2.07	0.039	.0018005	.0673542
INCOME	.0029204	.0018453	1.58	0.114	-.0006963	.0065372
FEMALE	.6910343	.3274524	2.11	0.035	.0492393	1.332829
BLACK	-.1651596	.1091418	-1.51	0.130	-.3790736	.0487544
HISPANIC	-.3803383	.1104741	-3.44	0.001	-.5968635	-.1638131
(*output omitted*)						
_cons	.4990339	.7443555	0.67	0.503	-.9598761	1.957944
regime1						
FAMSIZE	-.0808057	.012126	-6.66	0.000	-.1045721	-.0570392
EDUC	.0644769	.0070161	9.19	0.000	.0507255	.0782283
INCOME	.0046353	.0008534	5.43	0.000	.0029626	.006308
FEMALE	.9118476	.1288768	7.08	0.000	.6592537	1.164441
BLACK	-.2738078	.0478586	-5.72	0.000	-.3676088	-.1800067
HISPANIC	-.0968655	.0453532	-2.14	0.033	-.1857561	-.0079749
(*output omitted*)						
_cons	-.611228	.2526928	-2.42	0.016	-1.106497	-.1159592
selection						
FAMSIZE	-.0489857	.0120616	-4.06	0.000	-.072626	-.0253454
EDUC	-.0166882	.0070091	-2.38	0.017	-.0304258	-.0029505
INCOME	-.0010322	.0006916	-1.49	0.136	-.0023878	.0003234
FEMALE	.0089783	.1200184	0.07	0.940	-.2262534	.2442101
BLACK	.1720656	.0479293	3.59	0.000	.0781259	.2660054
HISPANIC	.1696283	.0480129	3.53	0.000	.0755248	.2637318
(*output omitted*)						
SPAGE	-.1104754	.0196521	-5.62	0.000	-.1489928	-.071958
LAGSPHMO	1.708451	.0442205	38.63	0.000	1.621781	1.795122
_cons	2.096168	.2477623	8.46	0.000	1.610563	2.581774
/atheta0	-.4474592	1.266758	-0.35	0.724	-2.930259	2.035341
/atheta1	-.8915252	.5012244	-1.78	0.075	-1.873907	.0908566
theta0	1.63925	.8097753			1.053383	8.654859
theta1	.4100299	.205517			.1535227	1.095112
tau1	-.1701348	.0707673			-.3538198	-.0712891

```
        tau0    .26288519
Wald test of independence :      Test statistic    4.581 with p-value  0.0668
```

The best-fitting combination of copulas is Joe for C_0 and the negative version of Clayton for C_1. The estimated τ's indicate that the dependence between ε_0 and ν is positive and the dependence between ε_1 and ν is negative. Both of the dependences are relatively weak, and we cannot reject the joint independence at the 5% level of significance, but we can at the 10% level.

Based on this estimated endogenous switching model, we estimate the treatment effects. For illustration, we also report the three estimates of LATE as well as ATE, ATT, and ATU. One is the LATE when the value of `LAGSPHMO` switches from 0 to 1. The other two are the LATE when the spouse gets older by one year from 34 years old (the value of `LAGSPHMO` increases from 3.4 to 3.5) and from 58 years old.

```
. * treatment effects
. * ATE, ATT, ATU
. estat teffects, ate att atu
Treatment Effects:

     Estimate        SE         z     P>|z|
ATE   -.04881    .03246   -1.5035    .13272
ATT   -.06514    .03784   -1.7212    .08522
ATU    .04346    .02279    1.9071    .05651

. * LATE
. estat teffects, late(LAGSPHMO 0 1)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .03536     .0226    1.5646    .11768

. estat teffects, late(SPAGE 3.4 3.5)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .03657    .02102    1.7394    .08196

. estat teffects, late(SPAGE 5.8 5.9)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .01841    .02005    .91838    .35842
```

As for the comparison, we show the estimates of the treatment effects under the assumption of the joint normality.

```
. * the joint normal model
. esbinary (`Y' `Xlist'), select(`S' = `Zlist') difficult vce(robust)
  (output omitted)
. * treatment effects
. * ATE, ATT, ATU
. estat teffects, ate att atu
Treatment Effects:

     Estimate        SE         z     P>|z|
ATE   -.00939    .02665   -.35255    .72442
ATT   -.01533    .03069   -.49941    .61749
ATU    .02431    .02467    .98541    .32442
```

```
. * LATE
. estat teffects, late(LAGSPHMO 0 1)
Treatment Effects:
      Estimate        SE         z     P>|z|
LATE    .02141    .02385    .89771    .36934
. estat teffects, late(SPAGE 3.4 3.5)
Treatment Effects:
      Estimate        SE         z     P>|z|
LATE     .0222    .02347    .94617    .34406
. estat teffects, late(SPAGE 5.8 5.9)
Treatment Effects:
      Estimate        SE         z     P>|z|
LATE    .01573    .02142    .73449    .46265
```

The estimates of the treatment effects are considerably different between the best-fitting copula model and the joint normal model, although the signs are the same. Under the joint normality assumption, none of the treatment effects are statistically significant. On the other hand, the ATT, the ATU, and one of the LATEs from the best-fitting copulas model are statistically significant, at least at the 10% level.

The heterogeneity in the treatment effects is more apparent when we see MTE. We estimate and plot MTEs over normalized values of ν with 95% confidence intervals. We normalize ν as $\Phi(\nu)$ so that it takes values between 0 and 1. This normalized value represents the propensity for treatment. Pointwise MTEs and standard errors are estimated using a loop.

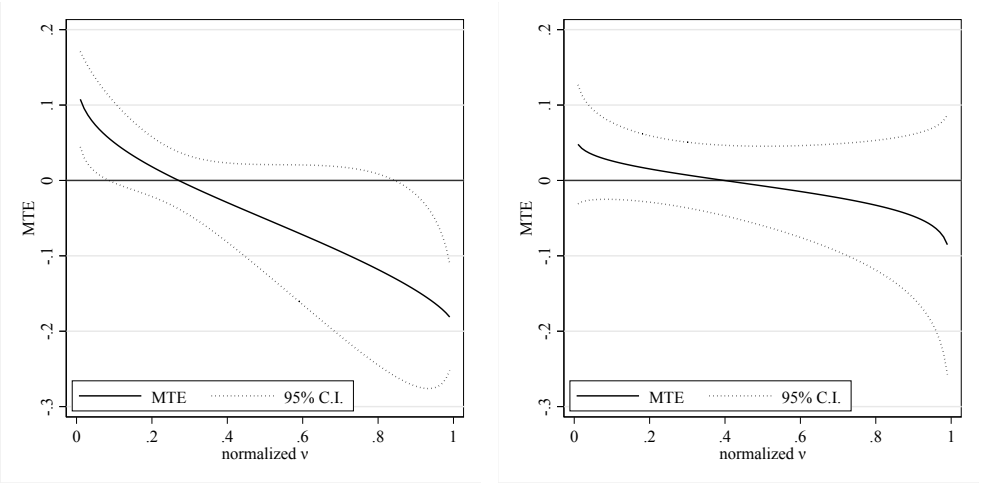
```
. * MTE
. generate P = .
(12,382 missing values generated)
. quietly generate MTEjn = .
. quietly generate MTEjn_se =.
. forvalue j = 1/99 {
  2.         display "`j'", _continue
  3.         local  p = `j'/100
  4.         local v = invnormal(`p')
  5.         quietly estat teffects, mte(`v')
  6.         matrix b = r(table)
  7.         quietly replace MTEjn = b["MTE","Estimate"] if _n == `j'
  8.         quietly replace MTEjn_se = b["MTE","SE"] if _n == `j'
  9. }
```

(*output omitted*)

Then, we plot the estimated MTEs over the propensity of the treatment with the 95% confidence intervals.

```
. generate lowjn = MTEjn - 1.96*MTEjn_se
(12,283 missing values generated)
. generate highjn = MTEjn + 1.96*MTEjn_se
(12,283 missing values generated)
```

```
. twoway (rline lowjn highjn P, sort bcolor(black) lpattern(dot))
>         (line MTEjn P, lcolor(black) lwidth(medium) lpattern(solid)),
>         xtitle("normalized {&nu}")
>         ytitle("MTE")
>         legend(order(2 "MTE" 1 "95% C.I.") ring(0) pos(7))
>         graphregion(color(white)) plotregion(lcolor(black))
>         yline(0)
>         name(mtejn, replace)
>         yscale(range(-0.3 0.2)) ytick(-0.3(0.1)0.2) ylabel(-0.3(0.1)0.2)
>         ysize(5) xsize(5)
```



(a) Best-fitting copulas (b) Joint normal

Figure 1. MTE from the endogenous switching model

We plot the estimated MTEs from the model with the best-fitting copulas in figure 1(a). Figure 1(b) shows the estimated MTEs from the joint normality for comparison. Both figures show the downward sloping MTE curves, indicating that the more likely an individual is to be treated, the smaller the treatment effect is. The result that the ATT is smaller than the ATU reflects this downward slope. While the MTE is not significant at any value of the propensity under the joint normality assumption, the MTEs from the best-fitting copulas are significantly different from 0 at lower and higher values of the treatment propensity. This pattern makes the ATT and the ATU (marginally) significant. The former puts more weight on MTE at higher values of the treatment propensity, but the latter weights MTEs at lower values.

Similarly, we can fit the endogenous dummy model and the treatment effects. At this time, there is only one copula to be specified, and a Frank copula is chosen as the best-fitting copula. As before, we also fit the model under the joint normality assumption for comparison. To save space, we suppress the results of the model estimation, and we report the estimation results of the treatment effects.

```
. edbinary `Y' `Xlist', select(`S' = `Zlist') difficult copula(frank) vce(robust)
```

(*output omitted*)

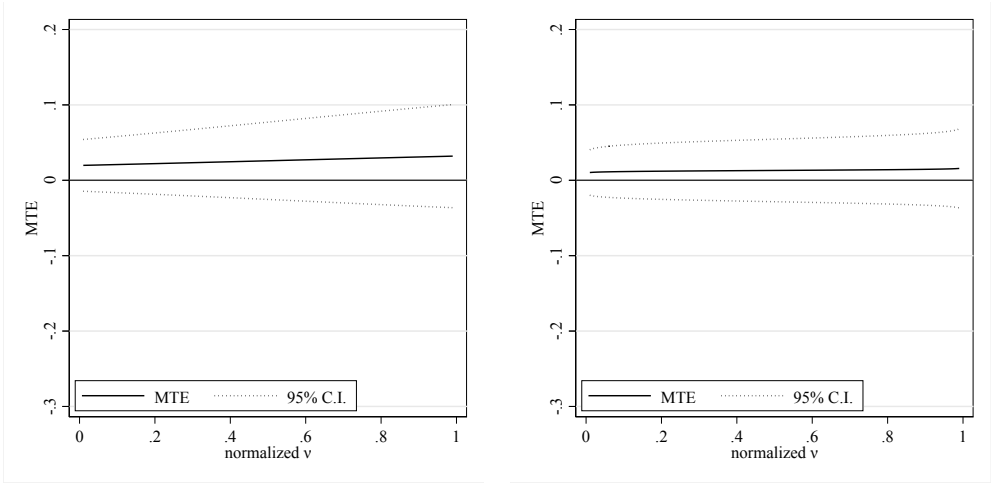
```
. * treatment effects
. * ATE, ATT, ATU
. estat teffects, ate att atu
Treatment Effects:

     Estimate        SE         z     P>|z|
ATE    .02589    .02853    .90731    .36424
ATT    .02665    .02973    .89657    .36995
ATU    .02153    .02174    .99054    .32191

. * LATE
. estat teffects, late(LAGSPHMO 0 1)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .02182    .02216    .98451    .32487

. estat teffects, late(SPAGE 3.4 3.5)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .02184    .02222    .98292    .32565

. estat teffects, late(SPAGE 5.8 5.9)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE     .0226     .0234    .96558    .33425

. * the joint normal model
. edbinary `Y' `Xlist', select(`S' = `Zlist') difficult vce(robust)
```

(*output omitted*)

```
. * treatment effects
. * ATE, ATT, ATU
. estat teffects, ate att atu
Treatment Effects:

     Estimate        SE         z     P>|z|
ATE    .01305    .02114    .61743    .53695
ATT    .01331    .02167    .61392    .53927
ATU    .01159    .01808    .64127    .52135

. * LATE
. estat teffects, late(LAGSPHMO 0 1)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE     .0118     .0185    .63776    .52363

. estat teffects, late(SPAGE 3.4 3.5)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .01163    .01816    .64061    .52178

. estat teffects, late(SPAGE 5.8 5.9)
Treatment Effects:

      Estimate        SE         z     P>|z|
LATE    .01195    .01882    .63484    .52553
```

Interestingly, all the treatment effects from the same model are close to one another. In other words, there is no heterogeneity in the treatment effects. This finding is clearer from the plots of MTEs (figure 2). The figures show almost horizontal lines. The MTEs from the best-fitting copula model are slightly larger than those from the joint normal

model. This explains why the ATE, ATT, ATU, and LATEs are higher in the copula model than the joint normal model.



(a) Best-fitting copula

(b) Joint normal

Figure 2. MTE from the endogenous dummy model

7 Conclusions

The issue of endogeneity is an empirical challenge that applied researchers often face. There are various kinds of endogeneity, and different econometric methods have been proposed in the literature. Apart from the endogeneity issue, researchers also frequently encounter binary choice outcomes. In this article, I discussed the three models of binary choice outcomes that are designed to address the endogeneity issue, and I introduced the commands to fit each of these models. Specifically, the command `esbinary` fits the endogenous switching model, `edbinary` fits the endogenous dummy model, and `ssbinary` fits the sample-selection model, implementing maximum likelihood estimation with a copula method. The copula method allows for various dependence structures of the unobservable terms. Moreover, one can estimate various treatment effects based on the estimated endogenous switching model or the endogenous dummy model by using the postestimation command `estat teffects`. The option to fit more flexible marginal distributions and the option to select a suitable copula function more formally are left as future tasks.

8 Acknowledgments

The author thanks an anonymous referee and the editor for helpful suggestions. This study was partially funded by Grant-in-Aid for Young Scientists from the Japan Society for the Promotion of Science: JSPS KAKENHI grant number JP18K12801.

9 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 22-4
. net install st0691     (to install program files, if available)
. net get st0691         (to install ancillary files, if available)
```

10 References

Aakvik, A., J. J. Heckman, and E. J. Vytlacil. 2005. Estimating treatment effects for discrete outcomes when responses to treatment vary: An application to Norwegian vocational rehabilitation programs. *Journal of Econometrics* 125: 15–51. https://doi.org/10.1016/j.jeconom.2004.04.002.

Andresen, M. E. 2018. Exploring marginal treatment effects: Flexible estimation using Stata. *Stata Journal* 18: 118–158. https://doi.org/10.1177/1536867X1801800108.

Cai, Z., and X. Wang. 2014. Selection of mixed copula model via penalized likelihood. *Journal of the American Statistical Association* 109: 788–801. https://doi.org/10.1080/01621459.2013.873366.

Chiburis, R. C. 2010. Semiparametric bounds on treatment effects. *Journal of Econometrics* 159: 267–275. https://doi.org/10.1016/j.jeconom.2010.07.006.

Dancer, D., A. Rammohan, and M. D. Smith. 2008. Infant mortality and child nutrition in Bangladesh. *Health Economics* 17: 1015–1035. https://doi.org/10.1002/hec.1379.

Deb, P., M. K. Munkin, and P. K. Trivedi. 2006. Bayesian analysis of the two-part model with endogeneity: Application to health care expenditure. *Journal of Applied Econometrics* 21: 1081–1099. https://doi.org/10.1002/jae.891.

Han, S., and S. Lee. 2019. Estimation in a generalization of bivariate probit models with dummy endogenous regressors. *Journal of Applied Econometrics* 34: 994–1015. https://doi.org/10.1002/jae.2727.

Han, S., and E. J. Vytlacil. 2017. Identification in a generalization of bivariate probit models with dummy endogenous regressors. *Journal of Econometrics* 199: 63–73. https://doi.org/10.1016/j.jeconom.2017.04.001.

Hasebe, T. 2013a. Marginal effects of a bivariate binary choice model. *Economic Letters* 121: 298–301. https://doi.org/10.1016/j.econlet.2013.08.028.

———. 2013b. Copula-based maximum-likelihood estimation of sample-selection models. *Stata Journal* 13: 547–573. https://doi.org/10.1177/1536867X1301300307.

———. 2021. On the treatment effects of a binary choice outcome model. *Economics Letters* 200: 109768. https://doi.org/10.1016/j.econlet.2021.109768.

Heckman, J. J., J. L. Tobias, and E. Vytlacil. 2003. Simple estimators for treatment parameters in a latent-variable framework. *Review of Economics and Statistics* 85: 748–755. https://doi.org/10.1162/003465303322369867.

Heckman, J. J., and E. J. Vytlacil. 2005. Structural equations, treatment effects, and econometric policy evaluation. *Econometrica* 73: 669–738. https://doi.org/10.1111/j.1468-0262.2005.00594.x.

———. 2007. Econometric evaluation of social programs, part II: Using the marginal treatment effect to organize alternative econometric estimators to evaluate social programs, and to forecast their effects in new environments. In Vol. 6B of *Handbook of Econometrics*, ed. J. J. Heckman and E. E. Leamer, 4875–5143. Amsterdam: Elsevier. https://doi.org/10.1016/S1573-4412(07)06071-0.

Lee, L.-F. 1983. Generalized econometric models with selectivity. *Econometrica* 51: 507–512. https://doi.org/10.2307/1912003.

Lokshin, M., and Z. Sajaia. 2011. Impact of interventions on discrete outcomes: Maximum likelihood estimation of the binary choice models with binary endogenous regressors. *Stata Journal* 11: 368–385. https://doi.org/10.1177/1536867X1101100303.

Nelsen, R. B. 2006. *An Introduction to Copulas*. 2nd ed. New York: Springer.

Newey, W. K., and D. McFadden. 1994. Large sample estimation and hypothesis testing. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier. https://doi.org/10.1016/S1573-4412(05)80005-4.

Smith, M. D. 2003. Modelling sample selection using Archimedean copulas. *Econometrics Journal* 6: 99–123. https://doi.org/10.1111/1368-423X.00101.

———. 2005. Using copulas to model switching regimes with an application to child labour. *Economic Record* 81: S47–S57. https://doi.org/10.1111/j.1475-4932.2005.00246.x.

Terza, J. V. 2016. Inference using sample means of parametric nonlinear data transformations. *Health Services Research* 51: 1109–1113. https://doi.org/10.1111/1475-6773.12494.

Trivedi, P. K., and D. M. Zimmer. 2007. Copula modeling: An introduction for practitioners. *Foundations and Trends in Econometrics* 1: 1–111. http://doi.org/10.1561/0800000005.

Winkelmann, R. 2012. Copula bivariate probit models: With an application to medical expenditures. *Health Economics* 21: 1444–1455. https://doi.org/10.1002/hec.1801.

About the author

Takuya Hasebe is an associate professor at the Faculty of Liberal Arts, Sophia University, Tokyo, Japan.