The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–21

Robit regression in Stata

Roger B. Newson
King's College London
London, United Kingdom
roger.newson@kcl.ac.uk

Milena Falcaro
King's College London
London, United Kingdom
milena.falcaro@kcl.ac.uk

Abstract.

Logistic and probit models are the most popular regression model for binary outcomes. A simple robust alternative is the robit model, which replaces the underlying Normal distribution in the probit model with a Student t–distribution. The heavier tails of the t–distribution (compared with the Normal distribution) means that model outliers are less influential. Robit regression can be fit as a generalized linear model with the link function defined as the inverse cumulative t–distribution function with a specified number of degrees of freedom (df), and it has been advocated as being particularly suitable for estimating inverse–probability weights and propensity scoring more generally. Here we describe a new package called `robit` that implements robit regression in Stata.

Keywords: st0001, `robit`, `xlink`, robit regression, binary regression, generalized linear models, inverse probability weights

1 Introduction

Robit regression models are similar to probit models, but the underlying Normal distribution in the latter is replaced by a central Student t–distribution with a zero median and ν degrees of freedom (df). More formally, they can be defined as generalized linear models with a binomial family (usually Bernoulli) variance function and a robit link function with ν df.

The Student t–distribution resembles a Normal distribution in that it is symmetrical and bell–shaped, but it has heavier tails. For this reason, it has been advocated as an alternative to the Normal distribution in defining regression models for continuous outcomes, without giving too much influence to outlying values. For example, this was done by Zellner (1976) and by Lange et al. (1989). These ideas were extended to regression models with binary outcomes (see e.g. Liu (2004)).

Compared to the better–known probit and logit link functions, the robit link gives less influence to observations that are highly unlikely given the values of the predictors. This property is discussed in Mudholkar and George (1982), Albert and Chib (1998), Liu (2004), and Kang and Schafer (2007), and is thought to make it particularly suited for use in estimating probability weights.

Seaman and White (2011) recommended the use of robit models for computing inverse probability weights to handle missing at random values and included this method in a list of useful techniques that are "not routinely available in most statistical soft-

ware". In the case of robit regression and Stata software, this is no longer true. We here present a package called `robit` that enables robit regression in Stata.

2 Methods and formulas

Robit models are a special class of generalized linear models (GLMs). GLMs were introduced by McCullagh and Nelder (1989), and are implemented in Stata using the `glm` command. Specifically, robit regression corresponds to a GLM with a binomial (usually Bernoulli) variance function and a robit link function.

In general, a link function $\eta(\mu)$ is an invertible monotonic transformation of the conditional mean μ, equal to a conditional probability in the case of a Bernoulli model. For instance, the logit link is defined as

$$\eta(\mu) \;=\; \ln\left[\mu/(1-\mu)\right], \tag{1}$$

and the probit link is defined as

$$\eta(\mu) \;=\; \Phi^{-1}(\mu), \tag{2}$$

where $\Phi(\cdot)$ is the cumulative standard Normal distribution function and $\Phi^{-1}(\cdot)$ is its inverse. And both of these link functions have twice–differentiable inverses. To fit a GLM with a specified link function, we need to be able to generate variables containing the link function $\eta(\mu)$ from the conditional mean μ, the inverse link function $\mu(\eta)$ from the link function, and also the first 2 derivatives of μ with respect to η.

A robit link function (also known as a t–link function) with ν df is defined by substituting an inverse cumulative t–distribution function for the inverse cumulative standard Normal distribution function in the probit link function, as

$$\eta(\mu) \;=\; F^{-1}_{t(\nu)}(\mu), \tag{3}$$

where $F_{t(\nu)}(\cdot)$ is the cumulative Student t–distribution function with ν df, and $F^{-1}_{t(\nu)}(\cdot)$ is its inverse. This link function also has a twice–differential inverse, with a first derivative given by

$$\frac{d\mu}{d\eta} \;=\; f_{t(\nu)}(\eta) \;=\; \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{\eta^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{4}$$

where $f_{t(\nu)}(\cdot)$ is the density function for the t–distribution with ν degrees of freedom. Therefore, differentiating (4) with respect to η, defining $u = 1+\eta^2/\nu$ and using the chain rule, we have the second derivative of μ with respect to η as

$$\frac{d^2\mu}{d\eta^2} \;=\; \frac{d}{du}\left(\frac{d\mu}{d\eta}\right)\frac{du}{d\eta} \;=\; -\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\,\frac{2\eta}{\nu}\,\frac{\nu+1}{2}\left(1+\frac{\eta^2}{\nu}\right)^{-\frac{\nu+3}{2}}. \tag{5}$$

The formulas (3), (4), and (5) define the variables that we need to generate, in order for `glm` to fit a robit model. As the official `glm` command does not allow the specification of

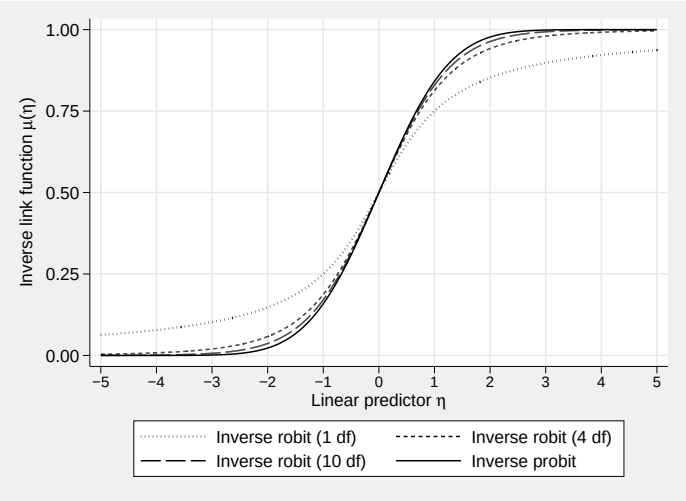

Figure 1: Inverse robit and probit link functions $\mu(\eta)$.

robit models, we wrote user–defined robit link functions to do this. See [R] **glm**, under **User–defined functions**, for technical details of how this is done.

Figure 1 shows the inverse robit link functions (also known as t–distribution functions) with df 1, 4 and 10, together with the inverse probit link function (also known as the Normal distribution function or as a t–distribution function with infinite df). Note that the fewer the df are, the further $\mu(\eta)$ is from 0 (in the case of negative η), or from 1 (in the case of positive η).

The choice of df for robit models still seems to be an open question. Kang and Schafer (2007) recommended 4 df, and, commenting on this article, Ridgeway and McCaffrey (2007) discuss and demonstrate the possibility of 1 df. Liu (2004) described 7 df as being an excellent approximation to the logit link function, but less influenced by model outliers. Albert and Chib (1998) discussed the case of 8 df. Robit with 9 df was mentioned by Mudholkar and George (1982) as having a similar kurtosis to the logit link function. In general, robit link functions with fewer df are influenced less by outliers than those with more df. In the limit, as ν tends to infinity, the robit model with ν df becomes the probit model. The df of a robit model can either be pre–specified by the user (for computational simplicity, as implemented in our `robit` package), or be estimated together with the other parameters of the model, possibly using an EM–type algorithm, as discussed in Liu (2006). Gelman et al. (2020), in their Chapter 15, express the view that an estimate of the df from the data "might be noisy".

Note that the t–distributions used by our packages are all standard t–distributions, specified uniquely by their df. Chapter 15 of Gelman et al. (2020) discusses the possibility of defining robit link functions using generalized t–distributions (with added scale parameters), to modify the units in which the parameters are expressed. Generalizations

of the t–distribution are reviewed for example in Li and Nadarajah (2020).

3 The package `robit`

`robit` ***depvar*** [*indepvars*] [*if*] [*in*] [*weight*] , dfreedom(#) [noconstant offset(*varname*) constraints(*constraints*) asis vce(*vcetype*) level(#) noheader notable collinear coeflegend difficult from(*init_specs*)]

where *depvar* is a dependent variable which must be binary.

`fweights`, `iweights`, `aweights`, and `pweights` are allowed; see help for `weight`.

`robit` has all the features available after estimation for `glm`, such as the `predict` and `margins` commands. See [R] **glm postestimation**.

3.1 Description

`robit` fits a robit regression model, with a number of degrees of freedom specified by the user. It requires the SSC package `xlink` in order to work.

3.2 Options

`dfreedom(#)` specifies the df for the robit model to be fitted. It must be specified, as an integer between 1 and 10.

`noconstant` suppresses the constant term (intercept) in the model.

`offset(`*varname*`)` specifies that *varname* be included in the model, with the coefficient constrained to be 1.

`constraints(`*constraints*`)` specifies the linear constraints to be applied during estimation. The default is to perform unconstrained estimation. See [R] **Estimation options**.

`asis` forces retention of perfect predictor variables and their associated, perfectly predicted observations. This may produce instabilities in maximization; see [R] **probit**.

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

`level(#)` specifies the confidence level, set to 95 if absent. See [R] **Estimation options**.

`noheader` suppresses the header information from the output. The coefficient table is

still displayed.

`notable` suppresses the table of coefficients from the output. The header information is still displayed.

`collinear` specifies that the estimation command not omit collinear variables. This option is seldom used because collinear variables make a model unidentified. However, you can add constraints to a model that will identify it even with collinear variables. See [R] **Estimation options** for details.

`coeflegend` instructs Stata not to show the coefficient results but to display instead the legend of the coefficients and how they should be specified in an expression.

`difficult` specifies that the likelihood function is likely to be difficult to maximize because of nonconcave regions. There is no guarantee that `difficult` will work better than the default; sometimes it is better and sometimes it is worse. You should use the `difficult` option only when the default stepper declares convergence and the last iteration is "not concave" or when the default stepper is repeatedly issuing "not concave" messages and producing only tiny improvements in the log likelihood. See [R] **Maximize**.

`from()` specifies initial values for the regression coefficients. See [R] **Maximize**.

3.3 Remarks

`robit` works by calling `glm` with a user–defined robit link function and a Bernoulli distribution family. More in detail, the link function is specified as `robit` followed by an integer between 1 and 10 representing the df; for example, `link(robit7)` corresponds to a robit link function with 7 df. We collected these user–written robit link functions into an SSC package called `xlink`, which must be installed in order for `robit` to work.

`robit` is designed to be user–friendly, and not to require advanced Stata or statistical skills. Users who want to fit robit models with the full power of `glm` can use `glm` directly, with a robit function from `xlink`. For example, `robit y x1 x2, df(4)` is equivalent to `glm y x1 x2, family(binomial) link(robit4)`. The use of `glm` in place of `robit` may be advantageous when, for instance, the specification of nonstandard maximization (see [R] **Maximize**) or display (see [R] **Estimation options**) options is needed.

3.4 Saved results

`robit` saves in `e()` all results saved by `glm` with a robit link and a Bernoulli variance family, and also the following:

Scalars

`e(depvarsum)`	sum of dependent variable in estimation sample

4 Examples

4.1 Creating an outlier in a simulated dataset

We illustrate the use of our `robit` command using a two–scenario simulation, similar in spirit to the one in Chapter 15 of Gelman et al. (2020). We generated data for 200 subjects, aiming to estimate the effect of a predictor `x` on a binary outcome `d` (1 if a subject has a disease, 0 otherwise). We assumed the predictor to be Normally distributed (as might be the case with the log of a biological assay result), with mean 0 and standard deviation 5. In the first scenario (the base scenario), we simulated a binary outcome `d`, using a logistic model with an intercept (log odds for zero `x`) of -3 and a log odds ratio of 1 per unit of `x`. This was done using the code

```
gen x=rnormal(0,5)
gen y=invlogit(-3+1*x)
gen d=runiform()<y
```

(See Buis (2007) for more about simulating binary and other discrete models.) In the second scenario (the outlier scenario), we introduced an outlier by switching the outcome of an extreme `x`–value from 0 to 1. Specifically, we created a new binary variable `d2`, which was as `d` in the first scenario, except that the subject with the smallest `x`–value (and therefore with the lowest probability of disease in the base scenario) was diagnosed (or misdiagnosed) as having the disease. Note that outliers are usually thought of as extreme observations, but, in the context of binary outcomes, are usually observations that are highly unlikely given the values of the predictors.

Of the 200 subjects, 52 had the disease in the base scenario, increasing to 53 in the outlier scenario (as the outcome of one observation was switched from 0 to 1). We fitted 3 binary regression models:

1. a logit model for the base scenario, regressing `d` with respect to `x`.
2. a logit model for the outlier scenario, regressing `d2` with respect to `x`.
3. a robit model with 4 degrees of freedom for the outlier scenario, regressing `d2` with respect to `x`.

We used Huber (or "robust") variances for consistency throughout, as not all the models were correctly specified, although we knew that the first one was, having carried out the simulation under it. For each of the 3 models, we estimated the probability of having the disease as a function of `x`. Note that using Huber variances does not affect the point estimates or the predicted probabilities.

The logit model for the base scenario gave the following results:

```
. logit d x, vce(robust)
Iteration 0:   log pseudolikelihood = -114.61138
Iteration 1:   log pseudolikelihood = -59.457924
Iteration 2:   log pseudolikelihood = -44.370079
Iteration 3:   log pseudolikelihood = -43.221978
Iteration 4:   log pseudolikelihood = -43.207413
```

```
Iteration 5:   log pseudolikelihood = -43.207411
Logistic regression                                     Number of obs =    200
                                                        Wald chi2(1)  =  22.36
                                                        Prob > chi2   = 0.0000
Log pseudolikelihood = -43.207411                       Pseudo R2     = 0.6230
```

d	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
x	1.001279	.2117557	4.73	0.000	.5862459	1.416313
_cons	-2.784094	.4921013	-5.66	0.000	-3.748595	-1.819593

```
. predict p_logit
(option pr assumed; Pr(d))
```

We see that the estimated log odds ratio is 1.001 per unit of `x` (95% CI, 0.586 to 1.416).

The logit regression in the outlier scenario produced the following output:

```
. logit d2 x, vce(robust)
Iteration 0:   log pseudolikelihood = -115.64441
Iteration 1:   log pseudolikelihood = -67.070275
Iteration 2:   log pseudolikelihood = -57.513152
Iteration 3:   log pseudolikelihood = -56.998017
Iteration 4:   log pseudolikelihood = -56.996734
Iteration 5:   log pseudolikelihood = -56.996734
Logistic regression                                     Number of obs =    200
                                                        Wald chi2(1)  =  10.75
                                                        Prob > chi2   = 0.0010
Log pseudolikelihood = -56.996734                       Pseudo R2     = 0.5071
```

d2	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
x	.7103551	.2166666	3.28	0.001	.2856964	1.135014
_cons	-2.041539	.5319693	-3.84	0.000	-3.084179	-.9988981

```
. predict p_logit_o
(option pr assumed; Pr(d2))
```

This time, the log odds ratio per `x`–unit is estimated as 0.710 (95% CI, 0.286 to 1.135). Therefore, creating the outlier has reduced the estimated log odds ratio (non–significantly).

The robit model under the outlier scenario produced output as follows:

```
. robit d2 x, dfr(4) vce(robust)
Iteration 0:   log pseudolikelihood =  -69.26115
Iteration 1:   log pseudolikelihood = -51.513345
Iteration 2:   log pseudolikelihood = -51.379071
Iteration 3:   log pseudolikelihood = -51.378786
Iteration 4:   log pseudolikelihood = -51.378786
Model: Robit with 4 d.f.
Number of obs: 200
Wald chi2(1): 20.013863
Prob > chi2: 7.688e-06
```

```
Log pseudolikelihood: -51.378786
```

d2	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
d2						
x	.6739678	.1506516	4.47	0.000	.3786962	.9692395
_cons	-1.84183	.3691371	-4.99	0.000	-2.565325	-1.118334

```
. predict p_robit_o
(option mu assumed; predicted mean d2)
```

This time, the regression coefficient of `d2` with respect to `x` is expressed in different units, namely units of the t–distribution with 4 df. The value is estimated as 0.674 (95% CI, 0.379 to 0.969). These units are not always easy to understand, but the predicted probabilities are. Figure 2 gives the predicted probabilities from each of the 3 models, together with the actual data points in the outlier scenario. We see that the predicted probability curve estimated with the logit model in the outlier scenario is less steep than that obtained from the logit model fitted to the base scenario. This is because the outlier (visible in the top left corner of the graph) is very atypical for its outcome group, making it the kind of outlier that has a large impact on the regression. However, the robit model fitted to the contaminated data (the outlier scenario) leads to predicted probabilities much more similar to those obtained from the logit model fitted to the base scenario data.

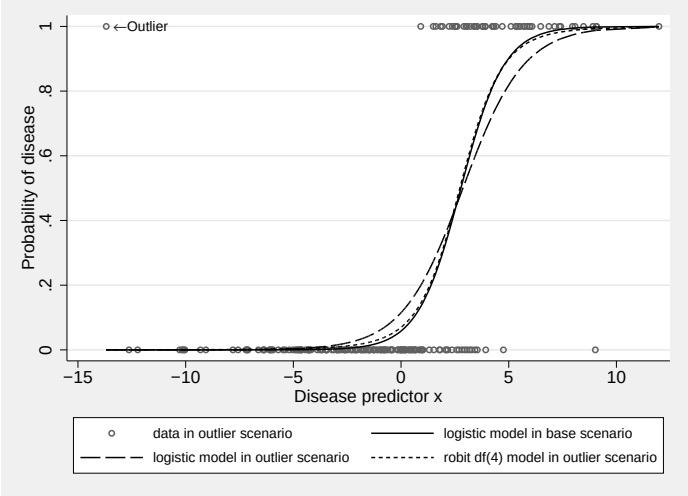

Figure 2: Predicted disease probabilities from the 3 models plotted against `x`.

4.2 Creating an outlier in propensity score analysis

In the real world, robit models are sometimes recommended for generating treatment–propensity scores (Ridgeway and McCaffrey (2007)) or completeness–propensity scores (Seaman and White (2011)). In both settings, the aim is to prevent outlying propensity weights. These may be encountered in a treatment–propensity setting if a treated subject has a very high predicted probability of being untreated or *vice versa*, or in a completeness–propensity setting if a subject with complete data has a very high predicted probability of having missing values. Hereafter we will describe an example of how robit regression can be used in the context of Rubin's causal model.

The Rubin method of confounder adjustment, in its 21st–century version described by Rubin (2008), is a 2–phase method for estimating the causal effect of a proposed intervention, using observational data. In Phase 1 ("design"), we fit a regression model to the sample data, predicting the exposure (which we propose to intervene to change) from confounders (expected to be unaffected). This model is used to define a propensity score, predicting exposure probability as a function of the confounders. In Phase 2 ("analysis"), we add in the outcome data, and use the propensity score in a second regression model to estimate a propensity–adjusted exposure effect on the outcome. This adjusted effect is interpreted as a difference between mean outcomes in two scenario populations, with the same propensity distribution, but different exposure levels. This is frequently done using inverse–propensity weighting.

As an example, we use the dataset of Cattaneo (2010) (see [TE] **teffects ipw**), which has 1 observation for each of 4642 pregnancies and data on self–reported maternal smoking status, child birth weight, and a list of candidate confounders, which predict maternal smoking and which might predict child birth weight. This dataset can be downloaded from within Stata and described as follows:

```
. use https://www.stata-press.com/data/r17/cattaneo2.dta
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138154)
. desc, fu
Contains data from https://www.stata-press.com/data/r17/cattaneo2.dta
 Observations:         4,642                  Excerpt from Cattaneo (2010)
                                                Journal of Econometrics 155:
                                                138154
    Variables:            23                  14 Jan 2020 09:49
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
bweight         int     %9.0g                 Infant birthweight (grams)
mmarried        byte    %11.0g     mmarried   1 if mother married
mhisp           byte    %9.0g                 1 if mother hispanic
fhisp           byte    %9.0g                 1 if father hispanic
foreign         byte    %9.0g                 1 if mother born abroad
alcohol         byte    %9.0g                 1 if alcohol consumed during
                                                pregnancy
deadkids        byte    %9.0g                 Previous births where newborn
                                                died
mage            byte    %9.0g                 Mother´s age
medu            byte    %9.0g                 Mother´s education attainment
fage            byte    %9.0g                 Father´s age
```

```
fedu            byte    %9.0g                 Father´s education attainment
nprenatal       byte    %9.0g                 Number of prenatal care visits
monthslb        int     %9.0g                 Months since last birth
order           byte    %9.0g                 Order of birth of the infant
msmoke          byte    %27.0g      smoke2    Cigarettes smoked during
                                                pregnancy
mbsmoke         byte    %9.0g       mbsmoke   1 if mother smoked
mrace           byte    %9.0g                 1 if mother is white
frace           byte    %9.0g                 1 if father is white
prenatal        byte    %9.0g                 Trimester of first prenatal care
                                                visit
birthmonth      byte    %9.0g                 Month of birth
lbweight        byte    %9.0g                 1 if low birthweight baby
fbaby           byte    %9.0g       YesNo     1 if first baby
prenatal1       byte    %9.0g       YesNo     1 if first prenatal visit in 1
                                                trimester
-------------------------------------------------------------------------------
Sorted by:
```

We will concentrate on the binary maternal smoking status (`mbsmoke`) as a predictor of the child's quantitative birth weight in grams (`bweight`). Of the 4642 pregnancies, 864 (18.61 percent) involved mothers who admitted to smoking during pregnancy, and there were no missing values for birth weight. The other covariates will be used in a propensity model to predict maternal smoking during pregnancy.

In the Rubin causal model, we are allowed to find a propensity model by trial and error in the exposure and confounder data, as long as we apply it to the outcome data afterwards, and write it up for publication unconditionally on whether it gives the answer we wanted to hear. We want the propensity model to predict the exposure, and at the same time to generate propensity weights that remove (or at least reduce) any imbalance in confounder values between the 2 exposure groups (self–reported smoking and nonsmoking mothers). We would also like this to be done in a way that does not lose too much power to detect a contrast in outcome between the exposure groups. And it is also important to define the kind of contrast that we aim to measure between the 2 exposure groups.

We will summarize our trial and error process by running the Rubin causal sequence for 4 candidate designs, based on the covariates of the `cattaneo2` dataset. These designs, corresponding to 4 combinations of 2 design matrices and 2 propensity models, are as follows:

1. Original dataset (without outliers), logit model.
2. Original dataset, robit model with 2 df.
3. Outlier dataset (with 1 observation altered to produce an outlier), logit model.
4. Outlier dataset, robit model with 2 df.

(The 2 df robit was itself chosen by trial and error, which we are allowed to do in the context of a Rubin causal design phase.) We will start by demonstrating the Rubin causal design phase in detail with the first design (original dataset, logit model), and

then proceed to presenting the other designs in less detail. The methods used will be similar to those presented in Newson (2016).

We start by fitting the logit propensity model in the original dataset as follows:

```
. logit mbsmoke mmarried mhisp fhisp foreign alcohol deadkids ///
>   mage medu fage fedu nprenatal monthslb order mrace frace ///
>   prenatal fbaby, ///
>   vce(robust)
Iteration 0:   log pseudolikelihood = -2230.7484
Iteration 1:   log pseudolikelihood = -1977.6794
Iteration 2:   log pseudolikelihood = -1956.3216
Iteration 3:   log pseudolikelihood = -1956.1193
Iteration 4:   log pseudolikelihood = -1956.1191
Logistic regression                                     Number of obs =  4,642
                                                        Wald chi2(17) = 469.16
                                                        Prob > chi2   = 0.0000
Log pseudolikelihood = -1956.1191                       Pseudo R2     = 0.1231
```

mbsmoke	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
mmarried	-1.023616	.1151624	-8.89	0.000	-1.24933	-.7979016
mhisp	-.9928626	.3883324	-2.56	0.011	-1.75398	-.231745
fhisp	-.2900995	.3653637	-0.79	0.427	-1.006199	.4260002
foreign	-.6518797	.2450522	-2.66	0.008	-1.132173	-.1715863
alcohol	1.596766	.1936559	8.25	0.000	1.217207	1.976325
deadkids	.3995759	.0909573	4.39	0.000	.2213028	.577849
mage	-.0277733	.0113963	-2.44	0.015	-.0501096	-.005437
medu	-.1092305	.0212416	-5.14	0.000	-.1508632	-.0675978
fage	.0030495	.0059549	0.51	0.609	-.0086218	.0147208
fedu	-.0540416	.0144029	-3.75	0.000	-.0822709	-.0258124
nprenatal	-.0295687	.011578	-2.55	0.011	-.0522611	-.0068763
monthslb	.0060745	.0015118	4.02	0.000	.0031115	.0090375
order	-.0141329	.0512791	-0.28	0.783	-.114638	.0863722
mrace	.5774233	.2231685	2.59	0.010	.1400211	1.014826
frace	.2472658	.2182185	1.13	0.257	-.1804346	.6749662
prenatal	.1004235	.0765748	1.31	0.190	-.0496603	.2505073
fbaby	-.3276605	.1316204	-2.49	0.013	-.5856319	-.0696892
_cons	1.175976	.3416339	3.44	0.001	.5063856	1.845566

We then compute the propensity score, equal, for each subject, to the estimated probability of smoking for that subject:

```
. cap drop propscor
. predict propscor
(option pr assumed; Pr(mbsmoke))
. lab var propscor "Propensity score"
. summ propscor, detail
                    Propensity score
      Percentiles      Smallest
 1%     .0216587       .0067189
 5%     .0459129       .0069247
10%     .0589656       .0077027       Obs               4,642
25%     .0888048       .0094055       Sum of wgt.       4,642
```

```
50%     .1421662                    Mean           .1861267
                        Largest     Std. dev.      .1405168
75%     .2434112        .8819793
90%     .3807587        .8905958    Variance        .019745
95%     .4689913        .8947034    Skewness       1.682102
99%     .6907851        .9081622    Kurtosis        6.30969
```

We see that subjects in the dataset have fitted probabilities of smoking ranging from .0067 to .9082. We would like to estimate average treatment effect (ATE) weights, sometimes known simply as inverse probability of treatment weights (IPTW). These can be used to estimate the difference in mean birthweight between 2 fantasy scenarios, defined as alternative versions of the dataset, one where all mothers admit to smoking during pregnancy and one where no mothers admit to smoking during pregnancy, both with other covariate values the same as in the original dataset. These weights are computed as follows:

```
. cap drop propwt
. gene propwt=cond(mbsmoke,1/propscor,1/(1-propscor))
. lab var propwt "Propensity ATE weight"
. summ propwt, detail
                    Propensity ATE weight
-------------------------------------------------------------
      Percentiles      Smallest
 1%     1.022138       1.006764
 5%     1.048729       1.006973
10%     1.063403       1.007763       Obs               4,642
25%     1.099818       1.009495       Sum of wgt.       4,642
50%     1.182591                      Mean           1.968402
                        Largest       Std. dev.      2.326476
75%     1.524615       28.72898
90%      3.91102       31.86331       Variance       5.412493
95%     6.362721       34.80801       Skewness       5.487046
99%     11.37603       38.67091       Kurtosis       50.21069
```

We see that the propensity ATE weights vary from 1.007 to 38.671.

To check whether these weights balance out the association of smoking with the propensity score and its component covariates, we will use Somers' D statistics for these associations, unweighted and weighted by the propensity ATE weights. Somers' D is discussed in Newson (2006) and Newson (2002) as an asymmetric measure of association, on a scale from -1 to 1, and related to Harrell's c–index $c(V|X)$ (also known as the ROC area of V with respect to X) by the formula $D(V|X) = 2c(V|X) - 1$, where X is the binary exposure variable and V can be either an outcome variable, a propensity score, or a confounder. In a propensity balance–checking context, it has advantages over the more commonly used standardized exposed–unexposed differences, used by official Stata's `teffects` command (see [TE] **teffects ipw**) and by Mark Lunt's `pbalchk` package (found by typing `findit pbalchk` in Stata). In particular, under a wide variety of regression models, $D(V|X)$ can be transformed to give a predictive treatment effect of X on V. For instance, if X and V are both binary, then $D(V|X)$ is exactly the difference between $\Pr(V = 1|X = 1)$ and $\Pr(V = 1|X = 0)$. And, if X is binary and V is conditionally equal–variance Normal, with different conditional means for each value of X and a common standard deviation (SD), and $D(V|X)$ is between -0.5 and $+0.5$, then $2D(V|X)$ is approximately the difference between the conditional means given

$X = 1$ and $X = 0$, expressed in units of the common SD. And, as $D(V|X)$ is invariant under any monotone–increasing Normalizing and variance–stabilizing transform on V, $2D(V|X)$ will be approximately the standardized difference between the corresponding conditional means of the transformed V. So, either way, for a confounder or propensity score W, a small propensity–weighted Somers' $D(W|X)$ can be used to give an upper bound to the spurious treatment effect on an outcome Y attributable to W, because a larger $D(Y|X)$ cannot be secondary to a smaller $D(W|X)$ with the same sign. And, a large propensity–weighted $D(W|X)$ indicates a problem of non–overlap, which our weighting has not balanced.

In our case, we measure the unweighted Somers' D values of the propensity score, and its component covariates, with respect to the exposure using the Stata command

```
. somersd mbsmoke propscor mmarried mhisp fhisp foreign alcohol deadkids ///
>   mage medu fage fedu nprenatal monthslb order mrace frace prenatal fbaby ///
>   , tdist
```

and the corresponding propensity–weighted Somers' D values using the Stata command

```
. somersd mbsmoke propscor mmarried mhisp fhisp foreign alcohol deadkids ///
>   mage medu fage fedu nprenatal monthslb order mrace frace prenatal fbaby ///
>   [pwei=propwt], tdist
```

And, instead of trying to digest the printed `somersd` output, we will look at Figure 3, which plots the unweighted and ATE–weighted indices against the propensity score and its component covariates. We see, from the unweighted indices, that the propensity score predicts smoking positively, and that its component covariates predict smoking positively or negatively. We also observe, from the ATE–weighted indices, that the ATE weights balance out most (but not quite all) of the predictive power, implying a limit to the potential spurious ATE attributable to residual confounding. Note that we have not included confidence intervals and P–values, as we are not really worrying about whether these associations arose by chance. We are worrying about whether these associations could be primary to whatever exposure–outcome associations may be discovered, once the outcome data are included.

Has the balancing power been won at the cost of inflating the confidence intervals for the outcome effects? We can answer this question using the SSC package `haif`, which measures homoskedastic adjustment inflation factors (often known as variance inflation factors). We can measure variance inflation caused either by including confounders in an outcome model or by using the confounders to compute propensity weights, under the pessimistic assumption that the confounder adjustment is not really necessary, because the "confounders" predict only the exposure, not the outcome. General principles of variance inflation can be found in Seber and Lee (2003). In our case, we imagine that we will fit a regression model for birthweight with 2 parameters, namely an intercept measuring average birthweights for babies with nonsmoking mothers, and a smoking effect (the ATE) measuring the difference in average birthweights between smoking and nonsmoking mothers, with ATE–weighted Huber variances. The output produced is as follows:

```
. haif mbsmoke, pwei(propwt)
Number of observations: 4642
Homoskedastic adjustment inflation factors
for variances and standard errors:
               Variance         SE
```

```
          mbsmoke   1.499336   1.224474
            _cons   1.072645   1.035686
```

The 2 columns of the listed output matrix contain inflation factors for the variances and standard errors, respectively. And the 2 rows correspond to the 2 parameters estimated, namely the smoking effect (the ATE) and the intercept estimating mean outcome for babies with nonsmoking mothers. We see that, if the confounders predict only smoking and not birthweight, then, for the ATE, variances (and therefore sample numbers required for a specified power) will be inflated by a factor of 1.499, and standard errors (and therefore confidence interval widths) will be inflated by a factor of 1.224.

We might decide, in the light of this design phase, to proceed to the analysis phase, and to measure the effect of smoking on birthweight, adjusted for the confounders. If we do this, then we fit a regression model of the outcome `bweight` with respect to the exposure `mbsmoke`, using the ATE weights as probability weights, as follows:

```
. regress bweight ibn.mbsmoke [pweight=propwt],noconst vce(robust) nohead
(sum of wgt is 9,137.32200610638)
```

bweight	Coefficient	Robust std. err.	t	P>|t|	[95% conf.	interval]
mbsmoke						
Nonsmoker	3404.982	9.749963	349.23	0.000	3385.867	3424.096
Smoker	3169.984	25.17523	125.92	0.000	3120.628	3219.339

The parameters here are the counterfactual scenario means for the dream scenario (where no mothers smoke) and the nightmare scenario (where all mothers smoke). We see that the mean birthweight is 3404.982 grams in the dream scenario and 3169.984 grams in the nightmare scenario. The nightmare–dream scenario difference is the ATE, and can be estimated using the SSC package `lincomest`, a version of `lincom` that saves its results as estimation results. (This enables us to tabulate the estimates, using the SSC packages `parmest` and `listtab`.)

```
. lincomest 1.mbsmoke-0.mbsmoke
Confidence interval for formula:
1.mbsmoke-0.mbsmoke
```

bweight	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
(1)	-234.998	26.9973	-8.70	0.000	-287.9256	-182.0705

We see that the ATE is -234.998 grams (95% CI, -287.926 to -182.071 grams). Note that the regression model is the same as the one assumed when we used `haif`, but with a different initial parameterization (two scenario means). The most interesting parameter (the ATE) is the one estimated using `lincomest`.

Alternatively, we might not proceed immediately to the analysis phase, but instead try out other designs. For the second design (original data, robit model), instead of using `logit`, we use `robit` with 2 df:

```
. robit mbsmoke mmarried mhisp fhisp foreign alcohol deadkids ///
>   mage medu fage fedu nprenatal monthslb order mrace frace ///
>   prenatal fbaby, ///
```

```
>   dfreedom(2) vce(robust)
Iteration 0:   log pseudolikelihood = -1994.5641
Iteration 1:   log pseudolikelihood = -1967.0614
Iteration 2:   log pseudolikelihood = -1966.3347
Iteration 3:   log pseudolikelihood = -1966.3318
Iteration 4:   log pseudolikelihood = -1966.3318
Model: Robit with 2 d.f.
Number of obs: 4642
Wald chi2(17): 390.92481
Prob > chi2: 1.460e-72
Log pseudolikelihood: -1966.3318
```

mbsmoke	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
mbsmoke						
mmarried	-.9152632	.1062371	-8.62	0.000	-1.123484	-.7070422
mhisp	-.7790951	.3695303	-2.11	0.035	-1.503361	-.0548291
fhisp	-.3426513	.3532382	-0.97	0.332	-1.034985	.3496828
foreign	-.539971	.2424757	-2.23	0.026	-1.015215	-.0647274
alcohol	1.295738	.1629519	7.95	0.000	.9763577	1.615118
deadkids	.3706682	.0816798	4.54	0.000	.2105788	.5307576
mage	-.0240536	.0112105	-2.15	0.032	-.0460258	-.0020814
medu	-.0888997	.0200479	-4.43	0.000	-.128193	-.0496065
fage	.000597	.0054267	0.11	0.912	-.0100391	.0112331
fedu	-.0386988	.0126222	-3.07	0.002	-.0634378	-.0139597
nprenatal	-.0246084	.0101487	-2.42	0.015	-.0444996	-.0047172
monthslb	.0051614	.0013556	3.81	0.000	.0025045	.0078183
order	-.0061435	.0460109	-0.13	0.894	-.0963233	.0840362
mrace	.5372472	.1972264	2.72	0.006	.1506905	.9238038
frace	.2083295	.1902149	1.10	0.273	-.1644848	.5811439
prenatal	.0820835	.0651084	1.26	0.207	-.0455266	.2096936
fbaby	-.3523408	.1188287	-2.97	0.003	-.5852407	-.1194409
_cons	.9871871	.3059164	3.23	0.001	.387602	1.586772

This time, the parameters are even less easy to understand, as they are expressed in robit units with 2 degrees of freedom. However, we can still compute propensity scores and ATE weights and do balance checks and variance inflation checks as before.

For the outlier designs, we identify a candidate outlier in the original dataset by choosing the subject with the lowest smoking propensity score under the logit model:

```
. summ propscor
```

Variable	Obs	Mean	Std. dev.	Min	Max
propscor	4,642	.1861267	.1405168	.0067189	.9081622

```
. gene byte candout=propscor==r(min)
. lab var candout "1 if candidate outlier"
. tab candout, m
```

1 if candidate outlier	Freq.	Percent	Cum.
0	4,641	99.98	99.98
1	1	0.02	100.00

```
      Total |      4,642      100.00
```

We see that the candidate outlier (identified by the indicator variable `candout`) is unique. To make the outlier dataset, we replace the values of a few variables in the outlier only, as follows:

```
. replace mage=40 if candout
(1 real change made)
. replace fage=40 if candout
(1 real change made)
. replace medu=30 if candout
(1 real change made)
. replace fedu=30 if candout
(1 real change made)
. replace nprenatal=40 if candout
(1 real change made)
. replace mbsmoke=1 if candout
(1 real change made)
. replace msmoke=1 if candout
(1 real change made)
```

We have revised this pregnancy (which already had a low smoking propensity) so that the mother and father are both 40 years old, both have 30 years of full–time education (being perpetual students), and bother their doctor sufficiently to have 40 prenatal visits (the maximum observed in the original data). All these features will probably predict a high social/educational rank and a low smoking propensity, as people with such features do not often smoke. However, we then make them smokers. As very atypical smokers, they will probably have a high propensity ATE weight. (These fantasy parents are possibly living off trust funds and smoking ganja weed.) Having created our pregnancy record with exceptional but credible parents, we can re–run our logit and robit propensity models, doing the balance and variance inflation checks as before.

The balance checks for the 4 designs are done by plotting the unweighted and ATE–weighted Somers' D statistics as reported in Figures 3, 4, 5, and 6, respectively. We see that the robit model on the original dataset balances the propensity score, and the covariates, similarly to the logit model on the original dataset. However, the logit model on the outlier dataset is a disaster, as a lot of weighted Somers' D indices are large in either direction, and the weighted Somers' D for the propensity score is actually negative. The robit model on the outlier dataset, by contrast, balances the covariates, and its propensity score, similarly to the logit and robit models on the original dataset.

The smoking propensity score percentiles for the 4 designs are given in Table 1. The corresponding smoking ATE weight percentiles are given in Table 2. We see that there is an enormous maximum ATE weight of 1808.188 for the logit model in the outlier dataset, which belongs to our generated outlier. This is probably important in preventing these weights from balancing. By contrast, the maximum ATE weight for the robit model in the outlier dataset is "only" 85.208, which does not seem to compromise the balance.

The variance inflation factors for the smoking ATE are given in Table 3. These are

Table 1: Smoking propensity score percentiles by design

	Percentile:				
Design	*0*	*25*	*50*	*75*	*100*
Original data, Logit model	0.0067	0.0888	0.1422	0.2434	0.9082
Original data, Robit model	0.0257	0.0956	0.1397	0.2302	0.8992
Outlier data, Logit model	0.0006	0.0899	0.1424	0.2422	0.8995
Outlier data, Robit model	0.0117	0.0960	0.1401	0.2301	0.8972

Table 2: Smoking propensity ATE weight percentiles by design

	Percentile:				
Design	*0*	*25*	*50*	*75*	*100*
Original data, Logit model	1.007	1.100	1.183	1.525	38.671
Original data, Robit model	1.026	1.108	1.178	1.500	22.349
Outlier data, Logit model	1.008	1.101	1.183	1.523	1808.188
Outlier data, Robit model	1.028	1.108	1.178	1.502	85.208

non–spectacular for all sets of weights, except for the weights from the logit model in the outlier scenario. The problem here is probably the outlier again. Outliers may or may not compromise the balance, but usually inflate the variance, at least if the covariates predict only the exposure, and not the outcome conditionally on the exposure.

On the basis of these design–stage results, we might choose to proceed to the analysis stage with either the robit or the logit for the original dataset, but would definitely prefer the robit for the outlier dataset. So the robit seems to rein in the effect of outlying pregnancies without doing any damage in the absence of outlying pregnancies.

The ATE estimates for smoking on birthweight in grams for the 4 designs (with confidence limits and P–values) are reported in Table 4. These are all similar to each other, except for the one for the logit model in the outlier dataset, which we would of course have rejected in the design phase.

5 Conclusions

We have developed a new user–friendly command (`robit`) for robit regression, and made available a set of user–written robit link functions (via the SSC package `xlink`) to be used with the `glm` command. This fills a gap in the pre–existing capabilities of Stata.

Robit models have been described in the literature as a simple robust alternative to logistic and probit models. In particular, they have been recommended for the estimation of inverse probability weights to adjust for missing–at–random values, or for deriving propensity scores for causal inference. Further work to evaluate the performance of robit models under various scenarios in these settings would be helpful.

We hope that our `robit` and `xlink` packages will be valuable additional tools in

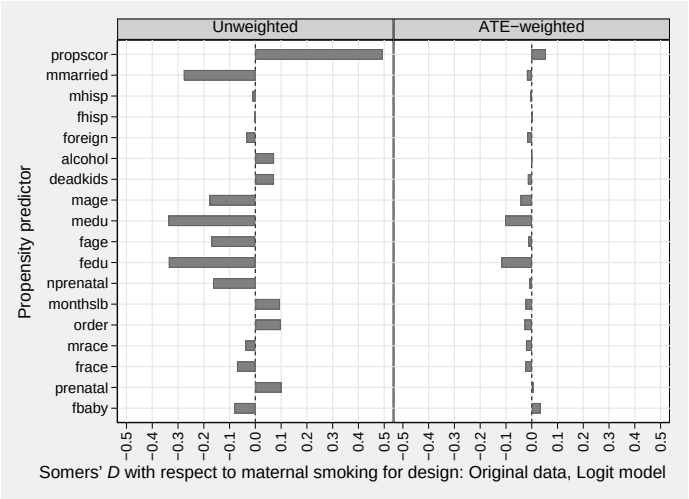

Figure 3: Somers' D indices with respect to maternal smoking under Design 1.

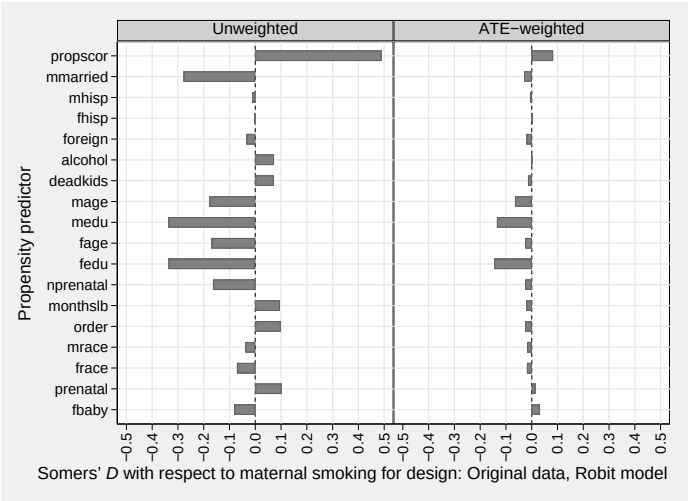

Figure 4: Somers' D indices with respect to maternal smoking under Design 2.

Table 3: Variance and SE inflation factors for the smoking ATE by design

Design	*Variance*	*SE*
Original data, Logit model	1.499	1.224
Original data, Robit model	1.351	1.162
Outlier data, Logit model	59.749	7.730
Outlier data, Robit model	1.565	1.251

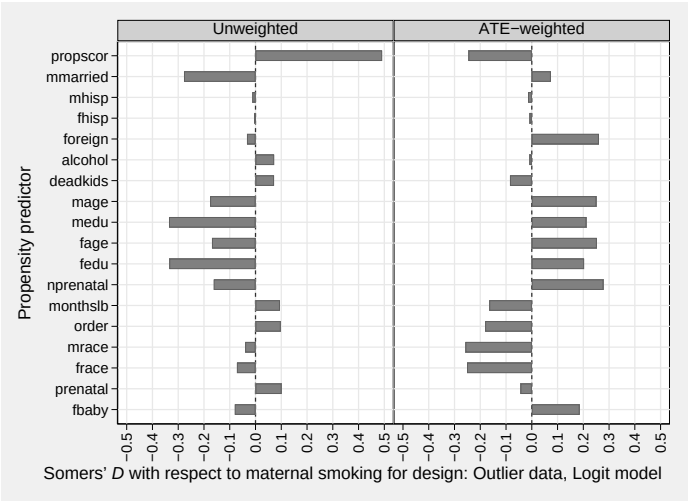

Figure 5: Somers' D indices with respect to maternal smoking under Design 3.

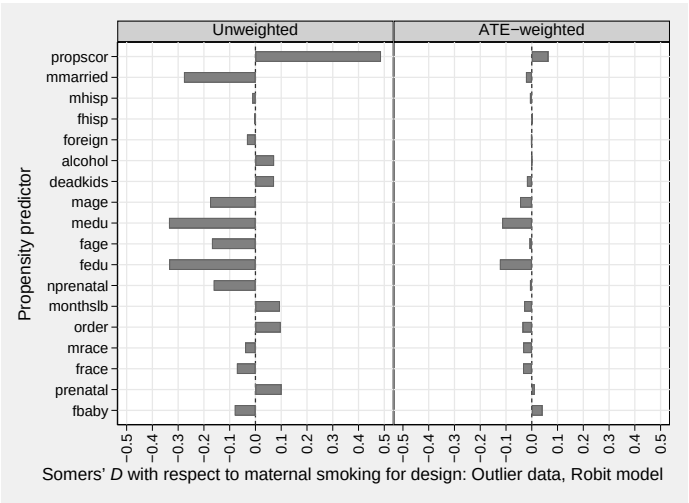

Figure 6: Somers' D indices with respect to maternal smoking under Design 4.

Table 4: Smoking ATE estimates for birthweight (grams) by design

Design	*ATE*	*(95%*	*CI)*	*P*
Original data, Logit model	−234.998	(−287.926,	−182.071)	4.4×10^{-18}
Original data, Robit model	−236.552	(−286.111,	−186.993)	1.2×10^{-20}
Outlier data, Logit model	−456.157	(−764.952,	−147.362)	.0038
Outlier data, Robit model	−251.693	(−307.925,	−195.461)	2.4×10^{-18}

Stata and will also promote sensitivity analyses, and further simulation studies.

6 Acknowledgements

This work was supported by Cancer Research UK (grant number: C8162/A27047). This article has benefited from very helpful discussions with our colleague Professor Peter Sasieni of King's College London.

7 References

Albert, J. H., and S. Chib. 1998. Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association* 88: 669–679.

Buis, M. L. 2007. Stata Tip 48: Discrete uses for `uniform()`. *The Stata Journal* 7: 434–435.

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi–valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. Downloaded from https://doi.org/10.1016/j.jeconom.2009.09.023 on October 7, 2022.

Gelman, A., J. Hill, and A. Vehtari. 2020. *Regression and Other Stories*. Cambridge University Press.

Kang, J. D. Y., and J. L. Schafer. 2007. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science* 22: 523–539.

Lange, K. L., R. J. A. Little, and J. M. G. Taylor. 1989. Robust Statistical Modeling Using the t Distribution. *Journal of the American Statistical Association* 84: 881–896.

Li, R., and S. Nadarajah. 2020. A review of Student's t distribution and its generalizations. *Empirical Economics* 58: 1461–1490.

Liu, C. H. 2004. Robit Regression: A Simple Robust Alternative to Logistic and Probit Regression. In *Bayesian Modeling and Causal Inference from Incomplete–Data Perspectives: An Essential Journey with Donald Rubin's Statistical Family*, ed. A. Gelman and X.-L. Meng, chap. 21. John Wiley & Sons, Ltd.

———. 2006. Robit Regression: A Simple Robust Alternative to Logistic and Probit Regression. Technical report, Bell Laboratories, Lucent Technologies. Downloaded from https://www.stat.purdue.edu/∼chuanhai/docs/robit.pdf on January 14, 2022.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Mudholkar, G. S., and E. O. George. 1982. A remark on the shape of the logistic distribution. *Biometrika* 65: 667–668.

Newson, R. 2002. Parameters behind "nonparametric" statistics: Kendall's tau, Somers' *D* and median differences. *The Stata Journal* 2: 45–64.

———. 2006. Confidence intervals for rank statistics: Somers' *D* and extensions. *The Stata Journal* 6: 309–334.

Newson, R. B. 2016. The role of Somers' *D* in propensity analysis. In *22nd United Kingdom Stata Users' Group Meeting*. Downloaded from the conference website at http://ideas.repec.org/p/boc/usug16/01.html on October 7, 2022.

Ridgeway, G., and D. F. McCaffrey. 2007. Comment: Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science* 22: 540–543.

Rubin, D. B. 2008. For objective causal inference, design trumps analysis. *The Annals of Applied Statistics* 2: 808–840.

Seaman, S. R., and I. R. White. 2011. Review of inverse probability weighting for dealing with missing data. *Statistical Methods in Medical Research* 22: 278–295.

Seber, G. A. F., and A. Lee. 2003. *Generalized Linear Models*, chap. 3.7. 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc.

Zellner, A. 1976. Bayesian and non–Bayesian analysis of the regression model with multivariate Student *t* error term. *Journal of the American Statistical Association* 66: 601–616.

About the authors

Roger B. Newson is a Statistician at King's College London, working principally in cancer research. He has written over 120 SSC packages (including `xlink`), some of which have been described in detail in articles in The Stata Journal.

Milena Falcaro is a Senior Statistician at King's College London. Her main research interests are in survival analysis, methods for missing values, longitudinal and multilevel data, simulation studies and programming.