The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$ 98	1-year subscription	$138
2-year subscription	$165	2-year subscription	$245
3-year subscription	$225	3-year subscription	$345
1-year student subscription	$ 75	1-year student subscription	$ 99
1-year institutional subscription	$245	1-year institutional subscription	$285
2-year institutional subscription	$445	2-year institutional subscription	$525
3-year institutional subscription	$645	3-year institutional subscription	$765
Electronic only		**Electronic only**	
1-year subscription	$ 75	1-year subscription	$ 75
2-year subscription	$125	2-year subscription	$125
3-year subscription	$165	3-year subscription	$165
1-year student subscription	$ 45	1-year student subscription	$ 45

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.


The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, stata, Stata Press, Mata, mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2014)
14, Number 3, pp. 580–604

A Stata package for the application of semiparametric estimators of dose–response functions

Michela Bia
CEPS/INSTEAD
Esch-Sur-Alzette, Luxembourg
michela.bia@ceps.lu

Carlos A. Flores
Department of Economics
California Polytechnic State University
San Luis Obispo, CA
cflore32@calpoly.edu

Alfonso Flores-Lagunes
Department of Economics
State University of New York, Binghamton
Binghamton, NY
aflores@binghamton.edu

Alessandra Mattei
Department of Statistics, Informatics, Applications "Giuseppe Parenti"
University of Florence
Florence, Italy
mattei@disia.unifi.it

Abstract. In many observational studies, the treatment may not be binary or categorical but rather continuous, so the focus is on estimating a continuous dose–response function. In this article, we propose a set of programs that semiparametrically estimate the dose–response function of a continuous treatment under the unconfoundedness assumption. We focus on kernel methods and penalized spline models and use generalized propensity-score methods under continuous treatment regimes for covariate adjustment. Our programs use generalized linear models to estimate the generalized propensity score, allowing users to choose between alternative parametric assumptions. They also allow users to impose a common support condition and evaluate the balance of the covariates using various approaches. We illustrate our routines by estimating the effect of the prize amount on subsequent labor earnings for Massachusetts lottery winners, using data collected by Imbens, Rubin, and Sacerdote (2001, *American Economic Review*, 778–794).

Keywords: st0352, drf, dose–response function, generalized propensity score, kernel estimator, penalized spline estimator, weak unconfoundedness

1 Introduction

The evaluation process in economics, sociology, law, and many other fields generally relies on applying nonexperimental techniques to estimate average treatment effects.

Propensity-score methods (Rosenbaum and Rubin 1983) are attractive empirical tools to balance the distribution of covariates between treatment groups and compare the groups in terms of observed covariates. Under the unconfoundedness assumption, which requires that potential outcomes are independent of the treatment conditional on the observed covariates, propensity-score methods allow one to eliminate (or at least reduce) the potential bias in treatment-effects estimates in observational studies. Most applications aim to evaluate causal effects of a binary treatment. There is extensive literature on identifying and estimating causal effects of binary treatments (for example, Imbens and Wooldridge [2009]; Stuart [2010]; Angrist, Imbens, and Rubin [1996]), and many statistical software packages have built-in or add-on functions for implementing methods to estimate causal effects of programs or policies. For example, Becker and Ichino (2002) developed a set of programs (`pscore.ado`) for estimating average treatment effects on the treated using propensity-score matching by focusing on four matching estimators: nearest-neighbor, radius, kernel, and stratification matching. More recently, building on the work of Becker and Ichino (2002), Dorn (2012) proposed a routine that helps improve covariate balance, and so the specification of the propensity-score model, using data-driven approaches.

In many empirical studies, treatments may take on many values, implying that participants in the study may receive different treatment levels. In such cases, one may want to assess the heterogeneity of treatment effects arising from variation in the amount of treatment exposure, that is, estimate a dose–response function (DRF). Over the past years, propensity-score methods have been generalized and applied to multivalued treatments (for example, Imbens [2000]; Lechner [2001]) and, more recently, to continuous treatments and arbitrary treatment regimes (for example, Hirano and Imbens [2004]; Imai and van Dyk [2004]; Flores et al. [2012]; Bia and Mattei [2012]; Kluve et al. [2012]).

In this article, we build on work by Hirano and Imbens (2004), who introduced the concept of the generalized propensity score (GPS) and used it to estimate the entire DRF of a continuous treatment. Hirano and Imbens (2004) used a parametric partial-mean approach to estimate the DRF. Here we focus on semiparametric techniques. Specifically, we present a set of programs that allows users to i) estimate the GPS under alternative parametric assumptions using generalized linear models;[1] ii) impose the common support condition as defined in Flores et al. (2012) and assess the balance of covariates after adjusting for the estimated GPS; and iii) estimate the DRF using the estimated GPS by applying either the nonparametric inverse-weighting (IW) kernel estimator developed in Flores et al. (2012) or a new set of semiparametric estimators based on penalized spline techniques.

1. Guardabascio and Ventura (2014) proposed the routine `gpscore2.ado` to estimate the GPS using generalized linear models.

We use a dataset collected by Imbens, Rubin, and Sacerdote (2001) to illustrate these programs and to evaluate the effect of the prize amount on subsequent labor earnings of winners of the Megabucks lottery in Massachusetts in the mid-1980s. We implement our programs to semiparametrically estimate the average potential postwinning labor earnings for each lottery prize amount. The prize is obviously assigned at random, but unit and item nonresponse lead to a self-selected sample where the prize amount received is no longer independent of background characteristics.

This article is organized as follows: Section 2 describes the methodological approach we refer to in the analysis. Section 3 introduces the GPS model and the semiparametric estimators of the DRF. Sections 3 and 3.2 show, respectively, the syntax and the options of the `drf` command. Section 5 illustrates the methods and the program using data from Imbens, Rubin, and Sacerdote (2001). Section 6 concludes.

2 Estimation strategy

We estimate a continuous DRF that relates each value of the dose (for example, lottery prize amount) to the outcome variable (for example, postwinning labor earnings) within the potential-outcome approach to causal inference (Rubin 1974, 1978). Formally, consider a set of N individuals, and denote each of them by subscript i: $i = 1, \ldots, N$. Under the stable unit treatment value assumption (Rubin 1980, 1990), for each unit i, there is a set of potential outcomes $\{Y_i(t)\}_{t\in\mathcal{T}}$, where $\mathcal{T}$ is a subset of the real line, $\mathcal{T} \subset \mathcal{R}$. We are interested in estimating the average DRF, $\mu(t) = E\{Y_i(t)\}$.

For each individual i, we observe a vector of pretreatment covariates, X_i, the received treatment level, T_i, and the corresponding value of the outcome for this treatment level, $Y_i = Y_i(T_i)$.

The central assumption of our approach is that the assignment to treatment levels is weakly unconfounded given the set of observed variables, that is, $Y_i(t) \perp T_i|X_i$ for all $t \in \mathcal{T}$ (Hirano and Imbens 2004). This assumption is described as weak unconfoundedness because it requires only conditional independence for each potential outcome $Y_i(t)$ rather than joint independence of all potential outcomes.

Under weak unconfoundedness, we can apply the GPS techniques for continuous treatments introduced by Hirano and Imbens (2004). Let $r(t, x) = f_{T|X}(t|x)$ be the conditional density of the treatment given the covariates. The GPS is defined as $R_i = r(T_i, X_i)$. The GPS is a balancing score (Rosenbaum and Rubin 1983; Hirano and Imbens 2004); that is, within strata with the same value of $r(t, x)$, the probability that $T = t$ does not depend on the value of X. The weak unconfoundedness assumption, combined with the balancing score property, implies that assignment to treatment is weakly unconfounded given the GPS. Formally,

$$f_T\left\{t|r(t, X_i), Y_i(t)\right\} = f_T\left\{t|r(t, X_i)\right\}$$

for every $t \in \mathcal{T}$ (theorem 1.2.2 in Hirano and Imbens [2004]). Thus any bias associated with differences in the distribution of covariates across groups with different treatment levels can be removed using the GPS. Formally, Hirano and Imbens (2004) showed that

if the assignment to the treatment is weakly unconfounded given pretreatment variables X_i, then $\mu(t) = E[\beta\{t, r(t, X_i)\}]$, where $\beta(t, r) = E\{Y_i(t)|r(t, X_i) = r\} = E(Y_i|T_i = t, R_i = r)$ (theorem 1.3.1 in Hirano and Imbens [2004]).

3 Inference

We use two-step semiparametric estimators of the DRF. The first step is to parametrically model and estimate the GPS, $R_i = r(T_i, X_i)$, and to assess the common support condition and the balance of the covariates. The second step is to estimate the average DRF, $\mu(t)$, using either the nonparametric IW kernel estimator proposed by Flores et al. (2012) or a semiparametric spline-based estimator. Here we describe these two steps, implemented in the routine `drf`.

3.1 Estimation of the GPS

The first part of the `drf` program estimates the GPS, allows users to impose an overlap condition, and tests the balancing property of the GPS.

The GPS is estimated parametrically and alternative distributional assumptions can be specified. Specifically, we assume that

$$g(T_i|X_i) \sim \psi\{h(\gamma, X_i), \theta\}$$

where g is a link function, ψ is a probability density function, h is a flexible function of the covariates depending on an unknown parameter vector γ, and θ is a scale parameter. In the `drf` program, we consider the Gaussian, inverse Gaussian, and Gamma distributions using the identity function, the logarithm, and the power function as link functions. We also implement a two-parameter beta distribution to address evaluation problems where the treatment variable takes on values in the interval $(0, 1)$, representing, for instance, a proportion. We use maximum likelihood methods to fit these models by using the official Stata command `glm` (see [R] **glm**) or the user-written package `betafit` (Buis, Cox, and Jenkins 2003).[2]

An important issue in GPS applications is determining the "common support" or "overlap region". The `drf` program allows users to do this by using the approach proposed by Flores et al. (2012). Specifically, the sample is first divided into K intervals according to the distribution of the treatment, cutting at the $100 \times (k/K)$th, $k = 1, \ldots, K-1$ percentiles of the treatment empirical distribution. Let q_k, $k = 1, \ldots, K$, denote these intervals, and let Q_i be the interval unit i belongs to: $T_i \in Q_i$. For each interval q_k, let $\widehat{R}_i^k$ be the GPS evaluated at the median level of the treatment in that interval for unit i, which is calculated for all units. The common support region with respect to q_k, denoted by CS_k, is obtained by comparing the support of the distribution

2. `betafit` (version 1.0.0 at the time of this writing) is available from the Statistical Software Components archive (or `findit betafit`) and must be installed separately from `drf`.

of $\widehat{R}_i^k$ for those units with $Q_i = q_k$ with that of units with $Q_i \neq q_k$ and is given by the subsample

$$CS_k = \left\{ i : \widehat{R}_i^k \in \left[\max\left(\min_{j:Q_j=q_k} \widehat{R}_j^k, \min_{j:Q_j\neq q_k} \widehat{R}_j^k \right), \min\left(\max_{j:Q_j=q_k} \widehat{R}_j^k, \max_{j:Q_j\neq q_k} \widehat{R}_j^k \right) \right] \right\}$$

Finally, the sample is restricted to units that are comparable across all the K intervals simultaneously by keeping only individuals who are simultaneously in the common support region for all k intervals. Therefore, the common-support subsample is given by $CS = \bigcap_{k=1}^{K} CS_k$.

As in applications of standard propensity-score methods, in GPS applications, it is crucial to evaluate how well the estimated GPS balances the covariates. Several methods can be applied to evaluate the balancing properties of the GPS. The `drf` command implements two approaches: an approach based on blocking on the GPS and an approach that uses a likelihood-ratio (LR) test. The "blocking on the GPS" approach was proposed by Hirano and Imbens (2004), and it is implemented in the `drf` routine using two-sided t tests or Bayes factors (see also Bia and Mattei [2008]). The second approach was proposed by Flores et al. (2012), who suggested using an LR test to compare an unrestricted model for T_i that includes all covariates and the GPS (up to a cubic term) with a restricted model that sets the coefficients of all covariates equal to zero. If the GPS sufficiently balances the covariates, then the covariates should have little explanatory power conditional on the GPS.[3]

3.2 Estimation of the dose–response function

We estimate the DRF by applying spline and kernel techniques. The first technique is implemented using a partial mean approach (Newey 1994). Specifically, for the penalized spline methods, we first estimate the conditional expectation of the observed outcome Y_i given the treatment actually received, T_i, and the GPS previously estimated in the first stage, $\widehat{R}_i$, using bivariate penalized spline smoothing based on i) additive spline bases; ii) tensor products of spline bases; or iii) radial basis functions (for example, Ruppert, Wand, and Carroll [2003]). Mixed models provide a representation of the penalized splines that allows smoothing to be done using mixed-model methodologies and software. In our routine, we use the Stata routine `xtmixed`, renamed `mixed` in Stata 13, to fit penalized spline regressions. The average DRF at t is then estimated by averaging the estimated regression function over the estimated score function evaluated at the specific treatment level t; that is, $\widehat{R}_i^t \equiv \widehat{r}(t, X_i)$.

3. An alternative approach, which is not implemented in our program, was proposed by Kluve et al. (2012). It consists of regressing each covariate on the treatment variable and comparing the significance of the coefficients for specifications with and without conditioning on the GPS.

The simplest bivariate penalized spline smoothing relies on additive spline bases, which can be formally defined in our setting as

$$E\left(Y_i|T_i,\widehat{R}_i\right) = a_0 + a_t T_i + a_r \widehat{R}_i + \sum_{k=1}^{K^t} u_k^t (T_i - k_k^t)_+ + \sum_{k=1}^{K^r} u_k^r \left(\widehat{R}_i - k_k^r\right)_+ \quad (1)$$

where for any number z, z_+ is equal to z if z is positive and is equal to 0 otherwise, and $k_1^t < \cdots < k_{K^t}^t$ and $k_1^r < \cdots < k_{K^r}^r$ are K^t and K^r distinct knots in the support of T and the estimated GPS, $\widehat{R}_i$, respectively.

The additive models have many attractive features, one being their simplicity. However, an additive model may not provide a satisfactory fit, so more complex models including interaction terms are required. To this end, we consider tensor product bases, which are obtained by forming all pairwise products of the basis functions $1, T_i, (T_i - k_1^t), \ldots, (T_i - k_{K^t}^t)$ and $1, \widehat{R}_i, (\widehat{R}_i - k_1^r), \ldots, (\widehat{R}_i - k_{K^r}^r)$. Formally,

$$\begin{aligned} E\left(Y_i|T_i,\widehat{R}_i\right) &= a_0 + a_t T_i + a_r \widehat{R}_i + a_{tr} T_i \widehat{R}_i \\ &\quad + \sum_{k=1}^{K^t} u_k^t \left(T_i - k_k^t\right)_+ + \sum_{k=1}^{K^r} u_k^r \left(\widehat{R}_i - k_k^r\right)_+ + \sum_{k=1}^{K^t} v_k^t \widehat{R}_i \left(T_i - k_k^t\right)_+ \\ &\quad + \sum_{k=1}^{K^r} v_k^r T_i \left(\widehat{R}_i - k_k^r\right)_+ + \sum_{k=1}^{K^t}\sum_{k'=1}^{K^r} v_{kk'}^{tr} \left(T_i - k_k^t\right)_+ \left(\widehat{R}_i - k_{k'}^r\right)_+ \end{aligned} \quad (2)$$

Estimation problems may arise when the tensor product approach is applied, especially if the sample size is relatively small. When these problems arise, the `drf` program alerts users and suggests they adopt an additive model instead.

As an alternative to tensor product splines, we propose to use the so-called radial basis functions, which are basis functions of the form $C\{\|(t,r)' - (k,k')'\|\}$ for some univariate function C. Here we consider the following function

$$C\left\{\left\|\begin{pmatrix} t \\ r \end{pmatrix} - \begin{pmatrix} k^t \\ k^r \end{pmatrix}\right\|\right\} = \left\|\begin{pmatrix} t \\ r \end{pmatrix} - \begin{pmatrix} k^t \\ k^r \end{pmatrix}\right\|^2 \log\left\|\begin{pmatrix} t \\ r \end{pmatrix} - \begin{pmatrix} k^t \\ k^r \end{pmatrix}\right\|$$

where $\|\cdot\|$ is the Euclidean norm, and we assume that

$$E\left(Y_i|T_i,\widehat{R}_i\right) = a_0 + a_t T_i + a_r \widehat{R}_i + a_{tr} T_i \widehat{R}_i + \sum_{k=1}^{K} u_k C\left\{\left\|\begin{pmatrix} T_i \\ \widehat{R}_i \end{pmatrix} - \begin{pmatrix} k_k^t \\ k_k^r \end{pmatrix}\right\|\right\} \quad (3)$$

where $u_1, \ldots, u_k$ are random variables with mean 0 and variance–covariance matrix $\mathrm{Cov}(u) = \sigma_u^2(\Omega_k^{-1/2})(\Omega_k^{-1/2})'$, with $\Omega_k = \left[C\left\{\left\|\begin{pmatrix} k_k^t \\ k_k^r \end{pmatrix} - \begin{pmatrix} k_{k'}^t \\ k_{k'}^r \end{pmatrix}\right\|\right\}\right]_{1\le k,k'\le K}$.

Given the estimated parameters of the regression functions (1), (2), or (3), the average potential outcome at treatment level t is estimated by averaging the estimated regression function over $\widehat{R}_i^t$.

Flores et al. (2012) proposed to estimate the DRF using a nonparametric IW estimator based on kernel methods. In this approach, the estimated scores are used to weight observations to adjust for covariate differences. Let $K(u)$ be a kernel function with the usual properties, and let h be a bandwidth satisfying $h \to 0$ and $Nh \to \infty$ as $N \to \infty$. The IW approach is implemented using a local linear regression of Y on T with weighted kernel function $\widetilde{K}_{h,X}(T_i - t) = K_h(T_i - t)/\widehat{R}_i^t$, where $K_h(z) = h^{-1}K(z/h)$. Formally, the IW kernel estimator of the average DRF is defined as

$$\widehat{\mu}(t) = \frac{D_0(t)S_2(t) - D_1(t)S_1(t)}{S_0(t)S_2(t) - S_1^2(t)}$$

where $S_j(t) = \sum_{i=1}^N \widetilde{K}_{h,X}(T_i - t)(T_i - t)^j$ and $D_j(t) = \sum_{i=1}^N \widetilde{K}_{h,X}(T_i - t)(T_i - t)^j Y_i$, $j = 0, 1, 2$.

We implement the IW estimator using a normal kernel. By default, the global bandwidth is selected using the procedure proposed by Fan and Gijbels (1996), which estimates the unknown terms in the optimal global bandwidth by using a global polynomial of order $p + 3$, where p is the order of the local polynomial fitted. However, users can also choose an alternative global bandwidth.

4 The drf command

4.1 Syntax

`drf` *varlist* [*if*] [*in*] [*weight*], `outcome(`*varname*`) treatment(`*varname*`)`
`cutpoints(`*varname*`) index(`*string*`) nq_gps(#) method(`*type*`)` [`gps`
`family(`*familyname*`) link(`*linkname*`) vce(`*vcetype*`) nolog(#) search`
`common(#) numoverlap(#) test_varlist(`*varlist*`) test(`*type*`) flag(#)`
`tpoints(`*vector*`) npoints(#) npercentiles(#) det delta(#)`
`bandwidth(#) nknots(#) knots(#) standardized degree1(#)`
`degree2(#) nknots1(#) nknots2(#) knots1(#) knots2(#) additive`
`estopts(`*string*`)`]

Note that the argument *varlist* represents the observed pretreatment variables, which are used to estimate the GPS. Note that `spacefill` must be installed (Bia and Van Kerm 2014).[4]

4.2 Options

Required

`outcome(`*varname*`)` specifies that *varname* is the outcome variable.

4. `spacefill` requires the Mata package `moremata` (Jann 2005).

`treatment(`*varname*`)` specifies that *varname* is the treatment variable.

`cutpoints(`*varname*`)` divides the range or set of the possible treatment values, $\mathcal{T}$, into intervals within which the balancing properties of the GPS are checked using a "blocking on the GPS" approach. *varname* is a variable indicating to which interval each observation belongs. This option is required unless `flag()` is set to `0` (see below).

`index(`*string*`)` specifies the representative point of the treatment variable at which the GPS must be evaluated within each treatment interval specified in `cutpoints()`. *string* identifies either the mean (*string* = `mean`) or a percentile (*string* = `p1`, ..., `p100`). This is used when checking the balancing properties of the GPS using a "blocking on the GPS" approach. This option is required unless `flag()` is set to `0` (see below).

`nq_gps(`#`)` specifies that for each treatment interval defined in `cutpoints()`, the values of the GPS evaluated at the representative point `index()` have to be divided into # (# $\in \{1, \ldots, 100\}$) intervals, defined by the quantiles of the GPS evaluated at the representative point `index()`. This is used when checking the balancing properties of the GPS using a "blocking on the GPS" approach. This option is required unless `flag()` is set to `0` (see below).

`method(`*type*`)` specifies the *type* of approach to be used to estimate the DRF. The approaches are bivariate-penalized splines (*type* = `mtspline`), bivariate penalized radial splines (*type* = `radialpspline`), or IW kernel (*type* = `iwkernel`).[5]

Global options

`gps` stores the estimated generalized propensity score in the `gpscore` variable that is added to the dataset.[6]

`family(`*familyname*`)` specifies the distribution used to estimate the GPS. The available distributional families are Gaussian (normal) (`family(gaussian)`), inverse Gaussian (`family(igaussian)`), Gamma (`family(gamma)`), and Beta (`family(beta)`). The default is `family(gaussian)`. The Gaussian, inverse Gaussian, and Gamma distributional families are fit using `glm`, and the beta distribution is fit using `betafit`.

The following four options are for the `glm` command, so they can be specified only when the Gaussian, inverse Gaussian, or Gamma distribution is assumed for the treatment variable.

`link(`*linkname*`)` specifies the link function for the Gaussian, inverse Gaussian, and Gamma distributional families. The available links are `link(identity)`, `link(log)`, and `link(pow)`, and the default is the canonical link for the `family()` specified (see help for `glm` for further details).

5. The subroutines `mtpspline` and `radialpspline` are called, respectively, when estimators with penalized splines (*type* = `mtspline`) and radial penalized splines (*type* = `radialpspline`) are used.
6. This option must not be specified when running the `bootstrap`.

`vce(`*vcetype*`)` specifies the type of standard error reported for the GPS estimation when the Gaussian, inverse Gaussian, or Gamma distribution is assumed for the treatment variable. *vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `eim`, `opg`, `bootstrap`, `jackknife`, `hac`, `kernel`, `jackknife1` (see `help glm` for further details).

`nolog(#)` is a flag ($\# = 0, 1$) that suppresses the iterations of the algorithm toward eventual convergence when running the `glm` command. The default is `nolog(0)`.

`search` searches for good starting values for the parameters of the generalized linear model used to estimate the generalized propensity score (see `help glm` for further details).

Overlap options

`common(#)` is a flag ($\# = 0, 1$) that restricts the inference to the subsample satisfying the common support condition when it is implemented ($\# = 1$). The default is `common(1)`.

`numoverlap(#)` specifies that the common support condition is imposed by dividing the sample into # groups according to # quantiles of the treatment distribution. By default, the sample is divided into 5 groups, cutting at the 20th, 40th, 60th, and 80th percentiles of the distribution if `common(1)`.

Balancing property assessment options

`test_varlist(`*varlist*`)` specifies that the balancing property must be assessed for each variable in *varlist*. The default `test_varlist()` consists of all the variables used to estimate the GPS.

`test(`*type*`)` allows users to specify whether the balancing property is to be assessed using a "blocking on the GPS" approach employing either standard two-sided t tests (`test(t_test)`) or Bayes factors (`test(Bayes_factor)`) or using a model-comparison approach with an LR test (`test(L_like)`).

The "blocking on the GPS" approach using standard two-sided t tests provides the values of the test statistics before and after adjusting for the GPS for each pretreatment variable included in `test_varlist()` and for each prefixed treatment interval specified in `cutpoints()`. Specifically, let p be the number of control variables in `test_varlist()`, and let H be the number of treatment intervals specified in `cutpoints()`. Then the program calculates and shows $p \times H$ values of the test statistic before and after adjusting for the GPS, where the adjustment is done by dividing the values of the GPS evaluated at the representative point `index()` into the number of intervals specified in `nq_gps()`. (See Hirano and Imbens [2004] for further details.)

The model-comparison approach uses a LR test to compare an unrestricted model for T_i, including all the covariates and the GPS (up to a cubic term), with a restricted model that sets the coefficients of all covariates to zero. By default, both the "blocking on the GPS" approach and the model-comparison approach are applied.

`flag(#)` allows the user to specify that `drf` estimates the GPS without performing the balancing test. The default is `flag(1)`, which means that the balancing property is assessed.

DRF options

`tpoints(`*vector*`)` indicates that the DRF is evaluated at each level of the treatment in *vector*. By default, the `drf` program creates a vector with jth element equal to the jth observed treatment value. This option cannot be used with `npoints()` or `npercentiles()` (see below).

`npoints(#)` indicates that the DRF is evaluated at each level of the treatment belonging to a set of evenly spaced values $t_0, t_1, \ldots, t_{\#}$ that cover the range of the observed treatment. This option cannot be used with `tpoints()` (see above) or `npercentiles()` (see below).

`npercentiles(#)` indicates that the DRF is evaluated at each level of the treatment corresponding to the percentiles $t_{q0}, t_{q1}, \ldots, t_{q\#}$ of the treatment's empirical distribution. This option cannot be used with `tpoints()` or `npoints()` (see above).

`det` displays more detailed output on the DRF estimation. When `det` is not specified, the program displays only the chosen DRF estimator: `method(radialpspline)`, `method(mtpspline)`, or `method(iwkernel)`.

`delta(#)` specifies that `drf` also estimate the treatment-effect function $\mu(t+\#) - \mu(t)$. The default is `delta(0)`, which means that `drf` estimates only the DRF, $\mu(t)$.

Options for the IW kernel estimator (iwkernel)

`bandwith(#)` specifies the bandwidth to be used. By default, the global bandwidth is chosen using the automatic procedure described in Fan and Gijbels (1996). This procedure estimates the unknown terms in the optimal global bandwidth by using a global polynomial of order $p+3$, where p is the order of the local polynomial fitted.

Options for the radial penalized spline estimator (radialpspline)

`nknots(#)` specifies the number of knots to be selected in the two-dimensional space of the treatment variable and the GPS. The default is `nknots(max(20, min(`n`/4, 150)))`, where n is the number of unique (T_i, R_i) (Ruppert, Wand, and Carroll 2003). When this option is specified, the subroutines `radialpspline` and `spacefill` (Bia and Van Kerm 2014) are called. This option cannot be used with the `knots()` option (see below).

`knots(`*numlist*`)` specifies the list of knots for the treatment and the GPS variable. This option cannot be used with the `nknots()` option (see above).

`standardized` implies that the `spacefill` algorithm standardizes the treatment variable and the GPS variables before selecting the knots. The knots are chosen using the standardized variables.

Options for the tensor-product penalized spline estimator (mtpspline)

`degree1(#)` specifies the power of the treatment variable included in the penalized spline model. The default is `degree1(1)`.

`degree2(#)` specifies the power of the GPS included in the penalized spline model. The default is `degree2(1)`.

`nknots1(#)` specifies the number (#) of knots for the treatment variable. The location of the K_kth knot is defined as $\{(k+1)/(\#+2)\}$th sample quantile of the unique T_i for $k = 1, \ldots, \#$. The default is `nknots1(max(5, min(`n`/4, 35)))`, where n is the number of unique T_i (Ruppert, Wand, and Carroll 2003). This option cannot be used with the `knots1(`*numlist*`)` option (see below).

`nknots2(#)` specifies the number (#) of knots for the GPS. The location of the K_kth knot is defined as $\{(k+1)/(\#+2)\}$th sample quantile of the unique R_i for $k = 1, \ldots, \#$. The default is `nknots2(max(5, min(`n`/4, 35)))`, where n is the number of unique R_i (Ruppert, Wand, and Carroll 2003). This option cannot be used with the `knots2()` option (see below).

`knots1(`*numlist*`)` specifies the list of knots for the treatment variable. This option cannot be used with the `nknots1()` option (see above).

`knots2(`*numlist*`)` specifies the list of knots for the GPS. This option cannot be used with the `nknots2()` option (see above).

`additive` allows users to implement penalized splines using the additive model without including the product terms.

Mutual options for the tensor-product and radial penalized spline estimators

Mutual options for the tensor-product and radial penalized spline estimators involve either the `mtpspline` subroutine or the `radialpspline` subroutine, depending on which estimator is used.

`estopts(`*string*`)` specifies all the possible options allowed when running the `xtmixed` models to fit penalized spline models (see `help xtmixed` for further details).

5 Example: The lottery dataset

We illustrate the methods and the programs discussed by reanalyzing data from a survey of Massachusetts lottery winners (see Imbens, Rubin, and Sacerdote [2001] for details on the survey). We focus on evaluating how the prize amount affects future labor earnings (from social security records). This example is also considered in Hirano and Imbens (2004).

The sample we use consists of 237 individuals who won a major prize in the lottery. The outcome of interest is earnings six years after winning the lottery (`year6`), and the treatment is the prize amount (`prize`). The lottery prize is randomly assigned, but there is substantial unit and item nonresponse as well as heterogeneity in the sample with respect to background characteristics. Thus it is more reasonable to conduct the analysis conditioning on the observed pretreatment variables under the weak unconfoundedness assumption.

Pretreatment variables are age, gender, years of high school, years of college, winning year, number of tickets bought, working status at the time of playing the lottery, and earnings s years before winning the lottery, $s = 1, 2, \ldots, 6$. To avoid results driven by outliers, we drop observations belonging to the upper 5% of the treatment variable distribution.

The output from running `drf`, shown below, is organized as follows. First, the GPS model and summary statistics of the estimated GPS are shown, and the common support is determined. The results show that 31 observations were dropped after we imposed the common support condition. Second, the balancing property is assessed. We specify the `test(L_like)` option for the balancing test, so results from only the model-comparison approach using the LR test are reported. The LR test shows that the GPS balances the covariates: they have little explanatory power conditional on the GPS. Indeed, the restricted model for T_i that excludes the covariates cannot be rejected at the usual significance levels (p-value is 0.284), whereas the restricted model that excludes the GPS is soundly rejected (p-value is 0).

```
. use lotterydataset.dta
. * we delete the extreme values (1 and 99 percentile)
. drop if year6==.
(35 observations deleted)
. summarize prize, de
              Treatment variable = Prize amount
-------------------------------------------------------------
      Percentiles      Smallest
 1%       5.3558         1.139
 5%        10.05             5
10%       11.246        5.3558       Obs                 202
25%       17.034         6.844       Sum of Wgt.         202
50%      32.1835                     Mean           57.36918
                       Largest       Std. Dev.      64.84194
75%       71.642         270.1
90%       137.27         305.09      Variance       4204.477
95%       171.73         323.32      Skewness       2.821964
99%       305.09         484.79      Kurtosis       14.18278
```

```
. drop if  prize >= r(p95)
(11 observations deleted)
. replace year6 = year6/1000
year6 was long now double
(92 real changes made)
. matrix define tp = (10\20\30\40\50\60\70\80\90\100)
. set seed 2322
. drf agew ownhs owncoll male tixbot workthen yearm1 yearm2 yearm3 yearm4
> yearm5 yearm6, outcome(year6) treatment(prize) gps test(L_like)
> tpoints(tp) numoverlap(3) method(radialpspline) family(gaussian)
> link(log) nknots(10) nolog(1) search det delta(1)
**************************************************
Algorithm to estimate the generalized propensity score
**************************************************

Estimation of the propensity score
Generalized linear models                         No. of obs      =        191
Optimization     : ML                             Residual df     =        178
                                                  Scale parameter =    1365.58
Deviance         =  243073.1517                   (1/df) Deviance =    1365.58
Pearson          =  243073.1517                   (1/df) Pearson  =    1365.58
Variance function: V(u) = 1                       [Gaussian]
Link function    : g(u) = ln(u)                   [Log]
                                                  AIC             =   10.12285
Log likelihood   =  -953.731889                   BIC             =   242138.2
```

prize	Coef.	OIM Std. Err.	z	P>|z|	[95% Conf.	Interval]
agew	.0158337	.0053884	2.94	0.003	.0052727	.0263947
ownhs	.0585063	.0742126	0.79	0.430	-.0869477	.2039603
owncoll	-.0108263	.0389408	-0.28	0.781	-.0871488	.0654962
male	.3615542	.1564085	2.31	0.021	.0549991	.6681093
tixbot	-.0174202	.0188308	-0.93	0.355	-.0543279	.0194875
workthen	.0680442	.1819285	0.37	0.708	-.2885291	.4246174
yearm1	-.0033454	.0102149	-0.33	0.743	-.0233662	.0166754
yearm2	.0018299	.0151926	0.12	0.904	-.0279471	.0316069
yearm3	-.0190244	.0134829	-1.41	0.158	-.0454505	.0074016
yearm4	.0451296	.0194034	2.33	0.020	.0070997	.0831596
yearm5	-.0094795	.0147496	-0.64	0.520	-.0383882	.0194293
yearm6	-.0055688	.0084792	-0.66	0.511	-.0221877	.0110501
_cons	2.534394	.489911	5.17	0.000	1.574186	3.494602

```
Note: The common support condition is imposed
```

```
****************************************************************
31 observations are dropped after imposing common support
****************************************************************
                        drf_gpscore
-------------------------------------------------------------

      Percentiles      Smallest
 1%     .0000774       .0000308
 5%       .00118       .0000774
10%     .0033023       .0003464       Obs                 160
25%     .0077024       .0004499       Sum of Wgt.         160

50%     .0092675                      Mean           .0082089
                       Largest        Std. Dev.       .002953
75%     .0103387       .0107928
90%     .0107204        .010793       Variance       8.72e-06
95%     .0107831       .0107953       Skewness      -1.419599
99%     .0107953       .0107956       Kurtosis       3.908883
******************************************
End of the algorithm to estimate the gpscore
******************************************

*********************************************************
Log-Likelihood test for Unrestricted and Restricted Model
*********************************************************

*************************************************
             Unrestricted Model
 link(E[T]) = GPSCORE + GPSCORE^2  + GPSCORE^3  + X
*************************************************
Generalized linear models                         No. of obs      =       160
Optimization     : ML                             Residual df     =       144
                                                  Scale parameter =   383.389

Deviance         =  55208.02303                   (1/df) Deviance =   383.389
Pearson          =  55208.02303                   (1/df) Pearson  =   383.389

Variance function: V(u) = 1                       [Gaussian]
Link function    : g(u) = ln(u)                   [Log]

                                                  AIC             =  8.881567
Log likelihood   = -694.5253454                   BIC             =   54477.2

------------------------------------------------------------------------------
             |                 OIM
       prize |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 drf_gpscore |  -139.9919   107.5174    -1.30   0.193    -350.7222    70.73837
drf_gpscore2 |   -45688.7   24107.57    -1.90   0.058    -92938.68    1561.268
drf_gpscore3 |    4243995    1464344     2.90   0.004      1373934     7114055
        agew |   .0067685   .0036542     1.85   0.064    -.0003935    .0139306
       ownhs |   .0159357   .0348134     0.46   0.647    -.0522974    .0841687
     owncoll |   .0146014    .028581     0.51   0.609    -.0414163    .0706192
        male |  -.0071926   .0945985    -0.08   0.939    -.1926022     .178217
      tixbot |  -.0120352   .0108077    -1.11   0.265     -.033218    .0091475
    workthen |  -.0411355   .1226241    -0.34   0.737    -.2814743    .1992032
      yearm1 |   .0042786   .0080239     0.53   0.594     -.011448    .0200052
      yearm2 |  -.0129785   .0123375    -1.05   0.293    -.0371595    .0112024
      yearm3 |   .0191091    .015091     1.27   0.205    -.0104687     .048687
      yearm4 |    .001562   .0113064     0.14   0.890    -.0205982    .0237222
      yearm5 |   -.008559   .0116933    -0.73   0.464    -.0314774    .0143595
      yearm6 |   .0002114     .00695     0.03   0.976    -.0134105    .0138332
       _cons |    4.74533   .2766597    17.15   0.000     4.203088    5.287573
------------------------------------------------------------------------------
```

```
*******************************************************
 Restricted Model: Pretreatment variables are excluded
     link(E[T]) = GPSCORE + GPSCORE^2  + GPSCORE^3
*******************************************************
Generalized linear models                         No. of obs      =       160
Optimization     : ML                             Residual df     =       156
                                                  Scale parameter =  386.9127
Deviance         =  60358.37384                   (1/df) Deviance =  386.9127
Pearson          =  60358.37384                   (1/df) Pearson  =  386.9127

Variance function: V(u) = 1                       [Gaussian]
Link function    : g(u) = ln(u)                   [Log]

                                                  AIC             =  8.820758
Log likelihood   = -701.6606578                   BIC             =  59566.65
```

prize	Coef.	OIM Std. Err.	z	P>|z|	[95% Conf.	Interval]
drf_gpscore	-84.75421	83.03918	-1.02	0.307	-247.508	77.99958
drf_gpscore2	-53755.36	20238.49	-2.66	0.008	-93422.08	-14088.64
drf_gpscore3	4533115	1287859	3.52	0.000	2008958	7057273
_cons	5.034825	.0706282	71.29	0.000	4.896396	5.173253

```
*******************************************************
 Restricted Model: GPS terms are excluded (link(E[T]) = X)
*******************************************************
Generalized linear models                         No. of obs      =       160
Optimization     : ML                             Residual df     =       147
                                                  Scale parameter =  1311.924
Deviance         =  192852.8661                   (1/df) Deviance =  1311.924
Pearson          =  192852.8661                   (1/df) Pearson  =  1311.924

Variance function: V(u) = 1                       [Gaussian]
Link function    : g(u) = ln(u)                   [Log]

                                                  AIC             =  10.09489
Log likelihood   = -794.5908861                   BIC             =  192106.8
```

prize	Coef.	OIM Std. Err.	z	P>|z|	[95% Conf.	Interval]
agew	.0196754	.0078967	2.49	0.013	.0041982	.0351525
ownhs	.0445558	.0879733	0.51	0.613	-.1278687	.2169802
owncoll	.0102703	.0484571	0.21	0.832	-.0847039	.1052445
male	.3800062	.1676205	2.27	0.023	.051476	.7085364
tixbot	-.0179112	.0212375	-0.84	0.399	-.0595359	.0237135
workthen	.1593496	.2189032	0.73	0.467	-.2696929	.5883921
yearm1	.0158358	.0119526	1.32	0.185	-.0075909	.0392624
yearm2	-.0347405	.0256188	-1.36	0.175	-.0849524	.0154713
yearm3	-.0074285	.0246622	-0.30	0.763	-.0557656	.0409086
yearm4	.0487374	.0278511	1.75	0.080	-.0058497	.1033245
yearm5	-.013943	.018552	-0.75	0.452	-.0503042	.0224183
yearm6	.000416	.0150639	0.03	0.978	-.0291088	.0299408
_cons	2.285246	.6383848	3.58	0.000	1.034035	3.536457

```
*****************************************************************
                    Likelihood-ratio tests:
 Comparison between the unrestricted model and the restricted models
*****************************************************************
LR_TEST[3,4]
                    Lrtest  T-Statistics       p-value  Restrictions
Unrestricted    -694.52535             .             .             .
Covariates X    -701.66066     14.270625      .2837616            12
   GPS terms    -794.59089     200.13108     3.952e-43             3
Number of observations = 160

*********************************************************
 End of the assesment of the balancing property of the GPS
*********************************************************
```

Then we estimate the DRF and the treatment-effect function, which represents the marginal propensity to earn out of the yearly prize money, using both penalized spline techniques and the IW kernel estimator. Following Hirano and Imbens (2004), we obtain the estimates of these functions at 10 different prize-amount values, considering increments of \$1,000 between \$10,000 and \$100,000 for the estimation of the treatment-effect function. Note that we scaled the prize amount by dividing it by \$1,000. To avoid redundancies, we show details on the output from running `drf` for only the radial penalized spline estimator (`method(radialpspline)`). Note that the `det` option is specified, so details on estimating the DRF are shown.

```
***************
 DRF estimation
***************

Radial penalized spline estimator
Run 1  ..                                       (Cpq =         383.37)
Run 2  ..                                       (Cpq =         427.99)
Run 3  ...                                      (Cpq =         388.19)
Run 4  ..                                       (Cpq =         365.61)
Run 5  ...                                      (Cpq =         389.08)

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -509.60164
Iteration 1:   log restricted-likelihood = -509.58312
Iteration 2:   log restricted-likelihood = -509.58286
Iteration 3:   log restricted-likelihood = -509.58286
```

```
Computing standard errors:
Mixed-effects REML regression                   Number of obs      =       129
Group variable: _all                            Number of groups   =         1
                                                Obs per group: min =       129
                                                               avg =     129.0
                                                               max =       129

                                                Wald chi2(2)       =      5.01
Log restricted-likelihood = -509.58286          Prob > chi2        =    0.0818
```

year6	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
prize	-.2582684	.215657	-1.20	0.231	-.6809484	.1644115
drf_gpscore	-1355.627	897.2735	-1.51	0.131	-3114.25	402.997
_cons	34.56937	11.09994	3.11	0.002	12.8139	56.32485

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
_all: Identity sd(__00002U..__000033)(1)	.0285723	.0584111	.0005198	1.570645
sd(Residual)	13.36947	.8725761	11.76412	15.19389

```
LR test vs. linear regression: chibar2(01) =     0.06 Prob >= chibar2 = 0.4072
(1) __00002U __00002V __00002W __00002X __00002Y __00002Z __000030 __000031
    __000032 __000033
. matrix list e(b)
e(b)[1,20]
            c1          c2          c3          c4          c5          c6
y1   15.131775   12.106819   9.3763398   7.2519104   6.0217689   5.5866336
            c7          c8          c9         c10         c11         c12
y1   5.7080575   5.9898157   6.0769106   5.7288158   -.3081758   -.2900365
           c13         c14         c15         c16         c17         c18
y1  -.23826795  -.15935109  -.05448761  -.00673878   .02770708   .02217719
           c19         c20
y1  -.01213146  -.06489899
. matrix C = e(b)
. drop gpscore
. set seed 2322
```

```
. bootstrap _b, reps(50): drf  agew  ownhs owncoll male tixbot workthen yearm1
> yearm2 yearm3 yearm4 yearm5 yearm6, outcome(year6) treatment(prize)
> test(L_like) tpoints(tp) numoverlap(3) method(radialpspline) family(gaussian)
> link(log) nolog(1) search nknots(10) det delta(1)
(running drf on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Bootstrap results                               Number of obs      =       191
                                                Replications       =        50
```

	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal-based [95% Conf.	Interval]
c1	15.13177	24.33924	0.62	0.534	-32.57225	62.8358
c2	12.10682	6.628999	1.83	0.068	-.8857812	25.09942
c3	9.37634	6.500001	1.44	0.149	-3.363427	22.11611
c4	7.25191	7.843234	0.92	0.355	-8.120547	22.62437
c5	6.021769	12.20073	0.49	0.622	-17.89122	29.93475
c6	5.586634	15.15628	0.37	0.712	-24.11914	35.2924
c7	5.708057	18.95607	0.30	0.763	-31.44515	42.86127
c8	5.989816	23.01648	0.26	0.795	-39.12166	51.10129
c9	6.076911	26.94703	0.23	0.822	-46.7383	58.89212
c10	5.728816	31.02343	0.18	0.853	-55.07598	66.53361
c11	-.3081758	2.3051	-0.13	0.894	-4.826088	4.209736
c12	-.2900365	2.43639	-0.12	0.905	-5.065274	4.485201
c13	-.2382679	.5888614	-0.40	0.686	-1.392415	.9158791
c14	-.1593511	.641826	-0.25	0.804	-1.417307	1.098605
c15	-.0544876	.4563326	-0.12	0.905	-.9488831	.8399079
c16	-.0067388	.4477181	-0.02	0.988	-.8842501	.8707725
c17	.0277071	.5016994	0.06	0.956	-.9556057	1.01102
c18	.0221772	.4548985	0.05	0.961	-.8694075	.9137618
c19	-.0121315	.4958827	-0.02	0.980	-.9840437	.9597808
c20	-.064899	.5120701	-0.13	0.899	-1.068538	.93874

Figures 1 and 2 show the estimates of the DRF and the treatment-effect function by using the semiparametric techniques implemented in the `drf` routine and a parametric approach. The parametric estimates are derived using the `doseresponse` routine (Bia and Mattei 2008), which follows the parametric approach originally proposed by Hirano and Imbens (2004).[7] As can be seen in figures 1 and 2, the two penalized spline estimators and the IW kernel estimator lead to similar results: the DRFs have a U shape (which is more tenuous in the case of the radial spline method) and the treatment-effect functions have irregular shapes increasing over most of the treatment range and decreasing for high treatment levels. The parametric approach shows quite a different picture. The DRF goes down sharply for low prize amounts and follows an inverse J shape for prize amounts greater than \$20,000. The treatment-effect function reaches a maximum around \$30,000, and then it slowly decreases.

7. The code to derive the graphs is shown here for only the radial penalized spline estimator.

```
. line radialest treatment, lcolor(black)
> yscale(r(6 18)) title("Radial spline method")
> xtitle("Treatment") ylabel(6 7 8 9 10 11 12 13 14 15 16 17 18)
> xlabel(0 10 20 30 40 50 60 70 80 90 100)
> ytitle("Dose-response function") scheme(medim)

. graph save  DRF_RAD.gph, replace
(file DRF_RAD.gph saved)

. graph export  DRF_RAD.eps, replace
(note: file DRF_RAD.eps not found)
(file DRF_RAD.eps written in EPS format)

. line radialder treatment, lcolor(black)
> yscale(r(-0.45 0.15)) title("Radial spline method")
> xtitle("Treatment") ylabel(-0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2)
> xlabel(0 10 20 30 40 50 60 70 80 90 100)
> ytitle("Derivative") scheme(medim)

. graph save  dDRF_RAD.gph, replace
(file dDRF_RAD.gph saved)

. graph export  dDRF_RAD.eps, replace
(note: file dDRF_RAD.eps not found)
(file dDRF_RAD.eps written in EPS format)
```

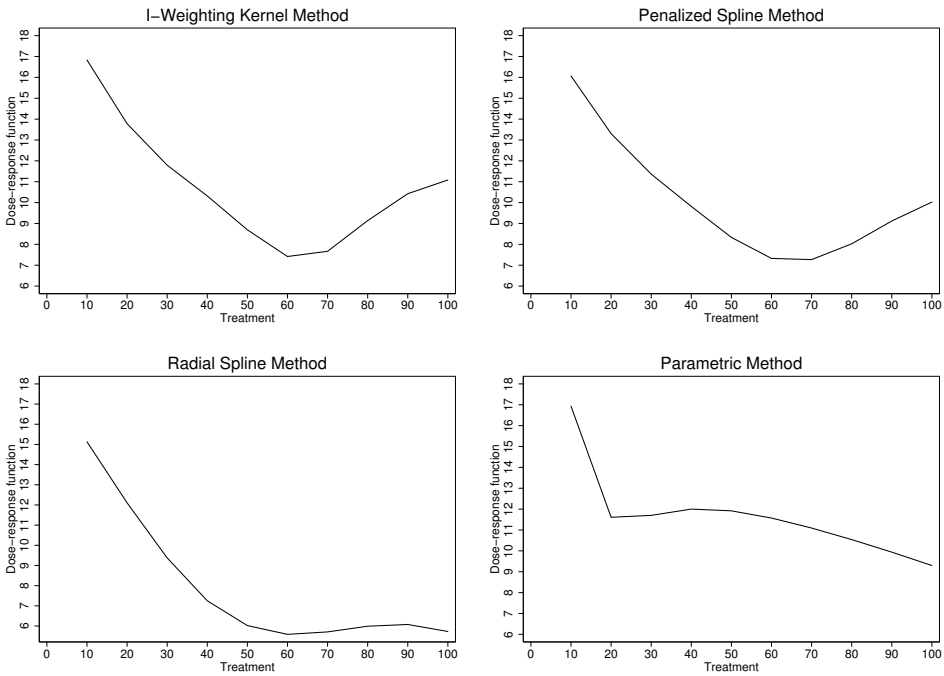

Figure 1. Estimated dose–response functions

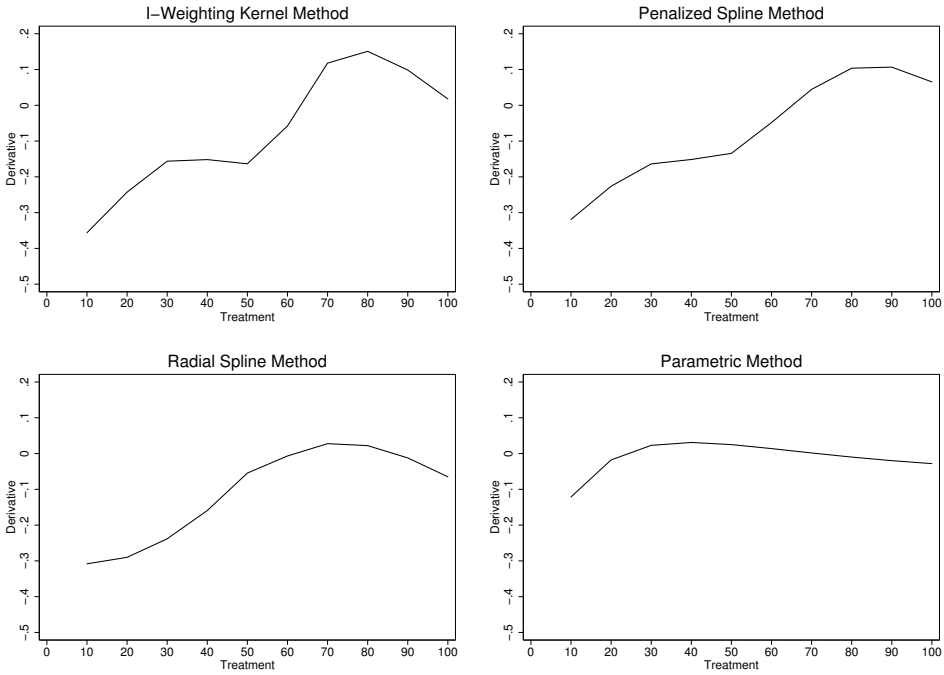

Figure 2. Estimated treatment-effect functions

Figures 3 and 4 show the DRFs and the treatment-effect functions estimated using the semiparametric and parametric techniques, now accompanied by pointwise 95% confidence bands. The confidence bands are based on a normal approximation using bootstrap standard errors, which are computed calling the `drf` program (or `doseresponse` program) in the `bootstrap` command.[8]

8. The radial spline-based models may produce slightly different estimates in different runs and when using the `bootstrap` command. This happens because within those models, an optimal set of "design points" is chosen via random selection of the knot values using the `spacefill` algorithm (see Bia and Van Kerm [2014] for further details). Some selected sets of knots may raise convergence issues depending on the data. Thus we recommend that users set a seed before running the `drf` code to make the results replicable.

```
. twoway (line upperEstRAD treatment, lcolor(black))
> (line radialest treatment, lcolor(black))
> (line lowerEstRAD treatment, lcolor(black)),
> yscale(r(-40 60)) xtitle("Treatment") ylabel(-40 -20 0 20 40 60)
> title("Radial spline method") ytitle("Dose-response function")
> xlabel(0 10 20 30 40 50 60 70 80 90 100) scheme(medim)

. graph save  CI_DRF_RAD.gph, replace
(file CI_DRF_RAD.gph saved)

. graph export  CI_DRF_RAD.eps, replace
(note: file CI_DRF_RAD.eps not found)
(file CI_DRF_RAD.eps written in EPS format)

. twoway (line upperDerRAD treatment, lcolor(black))
> (line radialder treatment, lcolor(black))
> (line lowerDerRAD treatment, lcolor(black)),
> yscale(r(-2 2)) xtitle("Treatment") ylabel(-2 -1 0.0 1 2)
> title("Radial spline method") ytitle("Derivative")
> xlabel(0 10 20 30 40 50 60 70 80 90 100) scheme(medim)

. graph save  CI_dDRF_RAD.gph, replace
(file CI_dDRF_RAD.gph saved)

. graph export  CI_dDRF_RAD.eps, replace
(note: file CI_dDRF_RAD.eps not found)
(file CI_dDRF_RAD.eps written in EPS format)
```

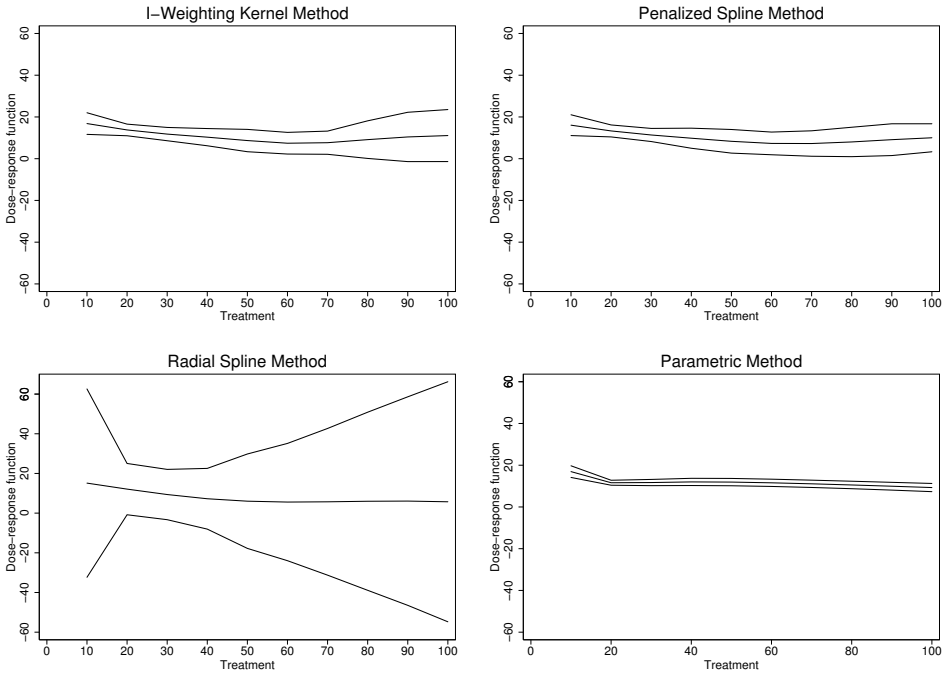

Figure 3. 95% confidence bands for the dose–response functions

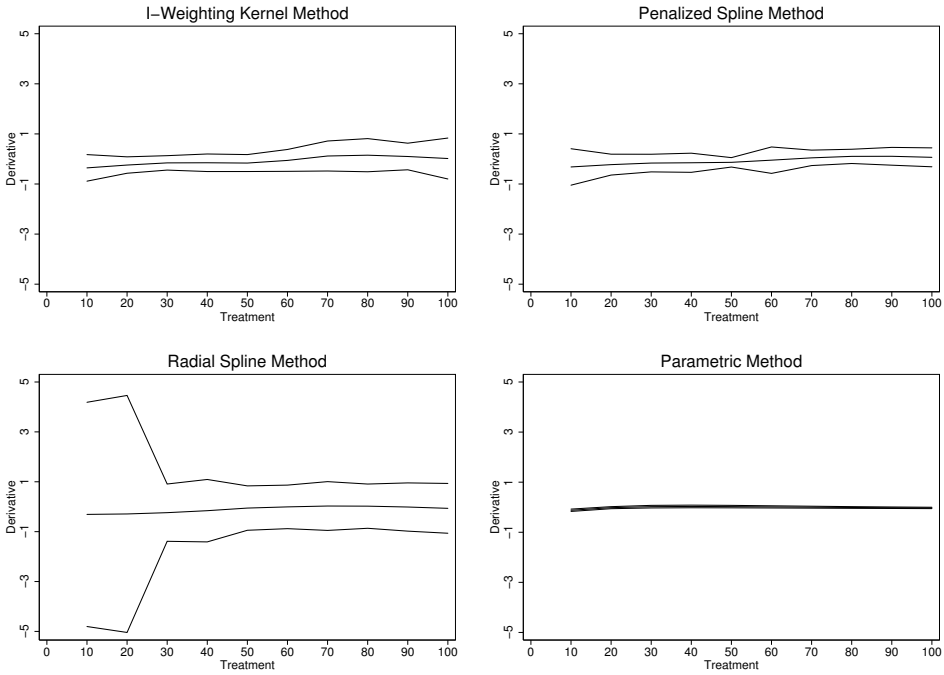

Figure 4. 95% confidence bands for the treatment-effect functions

The example allows us to highlight two important points. First, figures 3 and 4 show that differences in the point estimates and their precision among the three semiparametric estimators are more pronounced for low and high treatment levels. This is because our data are sparse for lower and higher values of the treatment.[9] Because of the nonparametric methods we use, estimation becomes noisier and the parameters are estimated less precisely in regions of the data with few observations, which is reflected in the wider confidence intervals. This is particularly evident for the radial spline approach, which seems to be more sensitive to the sample size than the IW and penalized splines estimators are. Second, it is clear from figures 3 and 4 that the parametric estimator produces much tighter confidence bands relative to the semiparametric estimators. This is due to the additional structure imposed by the parametric estimator, which allows extrapolation from regions where data are abundant to regions where data are scarce. However, if the assumptions behind the parametric structure are incorrect, the results, including their precision, are likely misleading.

9. In particular, there are very few observations for prizes lower than $15,000 and greater than $40,000.

6 Conclusion

We develop a program where we implement semiparametric estimators of the DRF based on the GPS, assuming that assignment to the treatment is weakly unconfounded given pretreatment variables. We propose three semiparametric estimators: the IW kernel estimator developed in Flores et al. (2012) and two estimators using penalized spline methods for bivariate smoothing. We use data from a survey of Massachusetts lottery winners to illustrate the proposed methods and program. We find that the semiparametric estimators provide estimates of the DRF and the treatment-effect function that are substantially different from those obtained when using the parametric approach originally proposed in Hirano and Imbens (2004). All the semiparametric estimators agree on a U-shaped DRF, which contrasts with the estimated inverse J shape uncovered by the parametric estimator. Although we cannot draw a firm conclusion about the relative performance of the estimators based on one dataset, we argue that a misspecification of the conditional expectation of the outcome given treatment and GPS could result in inappropriate removal of self-selection bias and in misleading estimates of the DRF. Therefore, it is advisable to also use semiparametric estimators that account for complicated structures that are difficult to model parametrically. Conversely, semiparametric estimators can be sensitive to the sample size and might not perform well in regions with few observations.

7 Acknowledgments

This research is part of the "Estimation of direct and indirect causal effects using semiparametric and nonparametric methods" project supported by the Luxembourg "Fonds National de la Recherche", which is cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND).

8 References

Angrist, J. D., G. W. Imbens, and D. B. Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association* 91: 444–455.

Becker, S. O., and A. Ichino. 2002. Estimation of average treatment effects based on propensity scores. *Stata Journal* 2: 358–377.

Bia, M., and A. Mattei. 2008. A Stata package for the estimation of the dose–response function through adjustment for the generalized propensity score. *Stata Journal* 8: 354–373.

———. 2012. Assessing the effect of the amount of financial aids to Piedmont firms using the generalized propensity score. *Statistical Methods & Applications* 21: 485–516.

Bia, M., and P. Van Kerm. 2014. Space-filling location selection. *Stata Journal* 14: 605–622.

Buis, M. L., N. J. Cox, and S. P. Jenkins. 2003. betafit: Stata module to fit a two-parameter beta distribution. Statistical Software Components S435303, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s435303.html.

Dorn, S. 2012. pscore2: Stata module to enforce balancing score property in each covariate dimension. UK Stata Users Group meeting. http://econpapers.repec.org/paper/bocusug12/11.htm.

Fan, J., and I. Gijbels. 1996. *Local Polynomial Modelling and Its Applications*. New York: Chapman & Hall/CRC.

Flores, C. A., A. Flores-Lagunes, A. Gonzalez, and T. C. Neumann. 2012. Estimating the effects of length of exposure to instruction in a training program: The case of job corps. *Review of Economics and Statistics* 94: 153–171.

Guardabascio, B., and M. Ventura. 2014. Estimating the dose–response function through a generalized linear model approach. *Stata Journal* 14: 141–158.

Hirano, K., and G. W. Imbens. 2004. The propensity score with continuous treatments. In *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, ed. A. Gelman and X.-L. Meng, 73–84. Chichester, UK: Wiley.

Imai, K., and D. A. van Dyk. 2004. Causal inference with general treatment regimes: Generalizing the propensity score. *Journal of the American Statistical Association* 99: 854–866.

Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710.

Imbens, G. W., D. B. Rubin, and B. I. Sacerdote. 2001. Estimating the effect of unearned income on labor earnings, savings, and consumption: Evidence from a survey of lottery players. *American Economic Review* 91: 778–794.

Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86.

Jann, B. 2005. moremata: Stata module (Mata) to provide various functions. Statistical Software Components S455001, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s455001.html.

Kluve, J., H. Schneider, A. Uhlendorff, and Z. Zhao. 2012. Evaluating continuous training programmes by using the generalized propensity score. *Journal of the Royal Statistical Society, Series A* 175: 587–617.

Lechner, M. 2001. Identification and estimation of causal effects of multiple treatments under the conditional independence assumption. In *Econometric Evaluation of Labour Market Policies*, ed. M. Lechner and F. Pfeiffer, 43–58. Heidelberg: Physica-Verlag.

Newey, W. K. 1994. Kernel estimation of partial means and a general variance estimator. *Econometric Theory* 10: 233–253.

Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55.

Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701.

———. 1978. Bayesian inference for causal effects: The role of randomization. *Annals of Statistics* 6: 34–58.

———. 1980. Bias reduction using Mahalanobis-metric matching. *Biometrics* 36: 293–298.

———. 1990. Comment: Neyman (1923) and causal inference in experiments and observational studies. *Statistical Science* 5: 472–480.

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

Stuart, E. A. 2010. Matching methods for causal inference: A review and a look forward. *Statistical Science* 25: 1–21.

About the authors

Michela Bia is a researcher at CEPS/INSTEAD, Population & Emploi, Esch-Sur-Alzette, Luxembourg.

Carlos A. Flores is an associate professor in the Department of Economics, Orfalea College of Business at the California Polytechnic State University.

Alfonso Flores-Lagunes is an associate professor in the Department of Economics at the State University of New York, Binghamton.

Alessandra Mattei is an assistant professor in the Department of Statistics, Informatics, Applications "Giuseppe Parenti" at the University of Florence.