The Stata Journal (2008)
8, Number 3, pp. 354–373

A Stata package for the estimation of the dose–response function through adjustment for the generalized propensity score

Michela Bia
Laboratorio Riccardo Revelli
Centre for Employment Studies
Collegio Carlo Alberto
Moncalieri, Italy
michela.bia@laboratoriorevelli.it

Alessandra Mattei
Department of Statistics
University of Florence
Florence, Italy
mattei@ds.unifi.it

Abstract. In this article, we briefly review the role of the propensity score in estimating dose–response functions as described in Hirano and Imbens (2004, *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, 73–84). Then we present a set of Stata programs that estimate the propensity score in a setting with a continuous treatment, test the balancing property of the generalized propensity score, and estimate the dose–response function. We illustrate these programs by using a dataset collected by Imbens, Rubin, and Sacerdote (2001, *American Economic Review* 91: 778–794).

Keywords: st0150, gpscore, doseresponse, doseresponse_model, bias removal, dose–response function, generalized propensity score, weak unconfoundedness

1 Introduction

Much of the work on propensity-score analysis has focused on cases where the treatment is binary. Matching estimators for causal effects of a binary treatment based on propensity scores have also been implemented in Stata (e.g., Becker and Ichino [2002] and Leuven and Sianesi [2003]).

In many observational studies, the treatment may not be binary or even categorical. In such a case, one may be interested in estimating the dose–response function where the treatment might take on a continuum of values. For example, in economics, an important quantity of interest is the effect of aid to firms (e.g., Bia and Mattei [2007]). In socioeconomic studies, one may be interested in the effect of the amount of a lottery prize on subsequent labor earnings (e.g., Hirano and Imbens [2004]).

Hirano and Imbens (2004) developed an extension to the propensity-score method in a setting with a continuous treatment. Following Rosenbaum and Rubin (1983) and most of the literature on propensity-score analysis, they make an unconfoundedness assumption, which allows them to remove all biases in comparisons by treatment status by adjusting for differences in a set of covariates. Then they define a generalization of the propensity score for the binary case—henceforth labeled generalized propensity score (GPS)—which has many of the attractive properties of the binary-treatment propensity score.

In this article, we briefly review the method developed by Hirano and Imbens (2004), and we provide a set of Stata programs that estimate the GPS, assess the adequacy of the underlying assumptions on the distribution of the treatment variable, test whether the estimated GPS satisfies the balancing property, and estimate the dose–response function. Following Hirano and Imbens (2004), our Stata programs address the problem of estimation and inference by using parametric models.

We illustrate these programs with a dataset collected from Imbens, Rubin, and Sacerdote (2001). The population consists of individuals who won the Megabucks lottery in Massachusetts in the mid-1980s. We apply our programs to estimate the average potential post-winning labor earnings for each level of the lottery prize (the dose–response function). Although the assignment of the prize is obviously random, substantial item and unit nonresponse led to a selected sample where the amount of the prize is no longer independent of background characteristics. In using these programs, remember that they only allow you to reduce, not to eliminate, the bias generated by unobservable confounding factors. As in the binary-treatment case, the extent to which this bias is reduced depends crucially on the richness and quality of the control variables, on which the GPS is computed.

2 The propensity score with continuous treatments

Suppose we have a random sample of size N from a large population. For each unit i in the sample, we observe a $p \times 1$ vector of pretreatment covariates, X_i; the treatment received, T_i; and the value of the outcome variable associated with this treatment, Y_i. Using the Rubin causal model (Holland 1986) as a framework for causal inference, we define a set of potential outcomes, $\{Y_i(t)\}_{t\in\mathcal{T}}$, $i = 1, \ldots, N$, where $\mathcal{T}$ is a continuous set of potential treatment values, and $Y_i(t)$ is a random variable that maps a particular potential treatment, t, to a potential outcome. Hirano and Imbens (2004) refer to $\{Y_i(t)\}_{t\in\mathcal{T}}$ as the unit-level dose–response function. We are interested in the average dose–response function, $\mu(t) = E\{Y_i(t)\}$. Following Hirano and Imbens (2004), we assume that $\{Y_i(t)\}_{t\in\mathcal{T}}$, T_i, and X_i, $i = 1, \ldots, N$, are defined on a common probability space; that T_i is continuously distributed with respect to the Lebesgue measure on $\mathcal{T}$; and that $Y_i = Y_i(T_i)$ is a well-defined random variable. To simplify the notation, we will drop the i subscript in the sequel.

The propensity function is defined by Hirano and Imbens (2004) as the conditional density of the actual treatment given the observed covariates.

Definition 2.1 (GPS) Let $r(t, x)$ be the conditional density of the treatment given the covariates:

$$r(t, x) = f_{T\,|\,X}(t\,|\,x)$$

Then the GPS is $R = r(T, X)$.

The GPS has a balancing property similar to that of the standard propensity score; 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:

$$X \perp I(T=t) \,|\, r(t,x)$$

where $I(\cdot)$ is the indicator function. Hirano and Imbens (2004) show that, in combination with a suitable unconfoundedness assumption, this balancing property implies that assignment to treatment is unconfounded, given the GPS.

Theorem 2.1 (Weak unconfoundedness given the GPS) Suppose that assignment to the treatment is weakly unconfounded, given pretreatment variables X:

$$Y(t) \perp T \,|\, X \qquad \text{for all } t \in \mathcal{T}$$

Then, for every t,

$$f_T\left\{t \,|\, r(t,X), Y(t)\right\} = f_T\left\{t \,|\, r(t,X)\right\}$$

Using this theorem, Hirano and Imbens (2004) show that the GPS can be used to eliminate any biases associated with differences in the covariates.

Theorem 2.2 (Bias removal with GPS) Suppose that assignment to the treatment is weakly unconfounded, given pretreatment variables X. Then

$$\beta(t,r) = E\left\{Y(t) \,|\, r(t,X)=r\right\} = E\left(Y \,|\, T=t, R=r\right)$$

and

$$\mu(t) = E\left[\beta\{t, r(t,X)\}\right]$$

3 Estimation and inference

The implementation of the GPS method consists of three steps. In the first step, we estimate the score $r(t,x)$. In the second step, we estimate the conditional expectation of the outcome as a function of two scalar variables, the treatment level T and the GPS R: $\beta(t,r) = E\left(Y \,|\, T=t, R=r\right)$. In the third step, we estimate the dose–response function, $\mu(t) = E[\beta\{t, r(t,X)\}]$, $t \in \mathcal{T}$, by averaging the estimated conditional expectation, $\widehat{\beta}\{t, r(t,X)\}$, over the GPS at each level of the treatment we are interested in.

3.1 Modeling the conditional distribution of the treatment given the covariates

The first step is to estimate the conditional distribution of the treatment given the covariates. We assume that the treatment (or its transformation) has a normal distribution conditional on the covariates:

$$g(T_i) \,|\, X_i \sim N\left\{h(\gamma, X_i), \sigma^2\right\} \tag{1}$$

where $g(T_i)$ is a suitable transformation of the treatment variable [$g(\cdot)$ may be the identity function], and $h(\gamma, X_i)$ is a function of covariates with linear and higher-order terms, which depends on a vector of parameters, γ. The choice of the higher-order terms to include is only determined by the need to obtain an estimate of the GPS that satisfies the balancing property.

The program `gpscore.ado` estimates the GPS and tests the balancing property according to the following algorithm:

1. Estimate the parameters γ and σ^2 of the conditional distribution of the treatment given the covariates (1) by maximum likelihood.[1]

2. Assess the validity of the assumed normal distribution model by one of the following user-specified goodness-of-fit tests: the Kolmogorov–Smirnov, the Shapiro–Francia, the Shapiro–Wilk, or the Stata skewness and kurtosis test for normality.

   a. If the normal distribution model is statistically disapproved, inform the user that the assumption of normality is not satisfied. The user is invited to use a different transformation of the treatment variable $g(T_i)$.

3. Estimate the GPS as

$$\widehat{R}_i = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} \exp\left[-\frac{1}{2\widehat{\sigma}^2}\left\{g(T_i) - h(\widehat{\gamma}, X_i)\right\}\right]$$

   where $\widehat{\gamma}$ and $\widehat{\sigma}^2$ are the estimated parameters in step 1.

4. Test the balancing property and inform the user whether and to what extent the balancing property is supported by the data. Following Hirano and Imbens (2004), the program `gpscore.ado` tests for balancing of covariates according to the following scheme:

   a. Divide the set of potential treatment values, $\mathcal{T}$, into K intervals according to a user-specified rule, which should be defined on the basis of the sample distribution of the treatment variable. Let $G_1, \ldots, G_K$ denote the K treatment intervals.

1. The model (1) is specified in the auxiliary ado-file `gpscore_model.ado`.

b. Within each treatment interval G_k, $k = 1, \ldots, K$, compute the GPS at a user-specified representative point (e.g., the mean, the median, or another percentile) of the treatment variable, which we denote by t_{G_k}, for each unit. Let $r(t_{G_k}, X_i)$ be the value of the GPS computed at $t_{G_k} \in G_k$ for unit i.

c. For each k, $k = 1, \ldots, K$, block on the scores $r(t_{G_k}, X_i)$, using m intervals, defined by the quantiles of order j/m, $j = 1, \ldots, m-1$, of the GPS evaluated at t_{G_k}, $r(t_{G_k}, X_i)$, $i = 1, \ldots, N$. Let $B_1^{(k)}, \ldots, B_m^{(k)}$ denote the m GPS intervals for the kth treatment interval, G_k.

d. Within each interval $B_j^{(k)}$, $j = 1, \ldots, m$, calculate the mean difference of each covariate between units that belong to the treatment interval, G_k, $\{i \colon T_i \in G_k\}$, and units that are in the same GPS interval, $\{i \colon r(t_{G_k}, X_i) \in B_j^{(k)}\}$, but belong to another treatment interval, $\{i \colon T_i \notin G_k\}$.

e. Combine the m differences in means, calculated in step d, by using a weighted average, with weights given by the number of observations in each GPS interval $B_j^{(k)}$, $j = 1, \ldots, m$. Specifically, the following weighted average is calculated for each of the p covariates X_l, $l = 1, \ldots, p$:

$$\frac{1}{N} \sum_{j=1}^{m} N_{B_j^{(k)}} \left\{ \overline{x}_{l,j}(G_k) - \overline{x}_{l,j}(G_k^c) \right\}$$

where $N_{B_j^{(k)}}$ is the number of observations in the $B_j^{(k)}$ GPS interval; $\overline{x}_{l,j}(G_k)$ is the mean of the covariate X_l for units i, such that $r(t_{G_k}, X_i) \in B_j^{(k)}$ and $T_i \in G_k$; and $\overline{x}_{l,j}(G_k^c)$ is the mean of the covariate X_l for units i', such that $r(t_{G_k}, X_{i'}) \in B_j^{(k)}$ and $T_{i'} \notin G_k$. The test statistics we use to evaluate the balancing property are functions of this weighted average.

f. For each G_k, $k = 1, \ldots, K$, test statistics (the Student's t statistics or the Bayes factors) are calculated and shown in the Results window. Finally, the most extreme value of the test statistics (the highest absolute value of the Student's t statistics or the lowest value of the Bayes factors) is compared with reference values, and the user is informed of the extent to which the balancing property is supported by the data.

3.2 Estimating the conditional expectation of the outcome given the treatment and GPS

In the second stage, we model the conditional expectation of the outcome, Y_i, given T_i and R_i, as a flexible function of its two arguments. We use polynomial approximations of order not higher than three. Specifically, the most complex model we consider is

$$\begin{aligned} &\varphi\left\{E\left(Y_i \,|\, T_i, R_i\right)\right\} = \psi(T_i, R_i; \alpha) \\ &\quad = \alpha_0 + \alpha_1 \cdot T_i + \alpha_2 \cdot T_i^2 + \alpha_3 \cdot T_i^3 + \alpha_4 \cdot R_i + \alpha_5 \cdot R_i^2 + \alpha_6 \cdot R_i^3 + \alpha_7 \cdot T_i \cdot R_i \end{aligned}$$

where $\varphi(\cdot)$ is a link function that relates the predictor, $\psi(T_i, R_i; \alpha)$, to the conditional expectation, $E(Y_i \mid T_i, R_i)$.

We assume that the main effects of T_i and R_i cannot be removed so that we have 18 possible submodels. The program `doseresponse_model.ado` defines all these models and estimates each of them by using the estimated GPS, $\widehat{R}_i$. When fitting the selected model, the program takes into account the nature of the outcome variable—which may be binary, categorical (nominal or ordinal), or continuous—by choosing the appropriate link function.

As Hirano and Imbens (2004) emphasize, there is no direct meaning to the estimated coefficients in the selected model, except that testing whether all coefficients involving the GPS are equal to zero can be interpreted as a test of whether the covariates introduce any bias.

3.3 Estimating the dose–response function

The last step consists of averaging the estimated regression function over the score function evaluated at the desired level of the treatment. Specifically, in order to obtain an estimate of the entire dose–response function, we estimate the average potential outcome for each level of the treatment we are interested in as

$$E\{\widehat{Y(t)}\} = \frac{1}{N}\sum_{i=1}^{N} \widehat{\beta}\{t, \widehat{r}(t, X_i)\} = \frac{1}{N}\sum_{i=1}^{N} \varphi^{-1}\left[\widehat{\psi}\{t, \widehat{r}(t, X_i); \widehat{\alpha}\}\right]$$

where $\widehat{\alpha}$ is the vector of the estimated parameters in the second stage.

The program `doseresponse.ado` estimates the dose–response function according to the following algorithm:

1. Estimate the GPS, verify the normal model used for the GPS, and test the balancing property calling the routine `gpscore.ado`.
2. Estimate the conditional expectation of the outcome, given the treatment and the GPS, by calling the routine `doseresponse_model.ado`.
3. Estimate the average potential outcome for each level of the treatment the user is interested in.
4. Estimate standard errors of the dose–response function via bootstrapping.[2]
5. Plot the estimated dose–response function and, if requested, its confidence intervals.

2. Hirano and Imbens (2004) state that asymptotic standard errors of the estimated dose–response function could be calculated by using expansions based on the estimating equations; these should take into account the estimation of the GPS as well as the α parameters. For practical reasons, our program uses bootstrap methods to obtain standard errors and confidence intervals of the dose–response function that take into account estimation of the GPS and the α parameters.

Some remarks on step 4 of the algorithm can be useful. When bootstrapped standard errors are requested, by activating the appropriate option (see sections 4 and 5), the bootstrap encompasses both the estimation of the GPS based on the specification given by the user and the estimation of the α parameters. Reestimating the GPS and the α parameters at each replication of the bootstrap procedure allows us to account for the uncertainty associated with the estimation of the GPS and the α parameters.

Typically, users would first identify a transformation of the treatment variable and a specification of the function h in (1), satisfying the normality assumption and the balancing property, respectively (by using, for instance, the routine `gpscore.ado`), and then provide exactly this transformation and this specification in the input to the program `doseresponse.ado`.

4 Syntax

`gpscore` *varlist* [*if*] [*in*] [*weight*] , `t(`*varname*`)` `gpscore(`*newvar*`)`
`predict(`*newvar*`)` `sigma(`*newvar*`)` `cutpoints(`*varname*`)` `index(`*string*`)`
`nq_gps(`#`)` [`t_transf(`*transformation*`)` `normal_test(`*test*`)` `norm_level(`#`)`
`test_varlist(`*varlist*`)` `test(`*type*`)` `flag(`#`)` `detail`]

`doseresponse_model` *treat_var GPS_var* [*if*] [*in*] [*weight*] , `outcome(`*varname*`)`
[`cmd(`*regression_cmd*`)` `reg_type_t(`*string*`)` `reg_type_gps(`*type*`)`
`interaction(`#`)`]

`doseresponse` *varlist* [*if*] [*in*] [*weight*] , `outcome(`*varname*`)` `t(`*varname*`)`
`gpscore(`*newvar*`)` `predict(`*newvar*`)` `sigma(`*newvar*`)` `cutpoints(`*varname*`)`
`index(`*string*`)` `nq_gps(`#`)` `dose_response(`*newvarlist*`)`
[`t_transf(`*transformation*`)` `normal_test(`*test*`)` `norm_level(`#`)`
`test_varlist(`*varlist*`)` `test(`*type*`)` `flag(`#`)` `cmd(`*regression_cmd*`)`
`reg_type_t(`*type*`)` `reg_type_gps(`*type*`)` `interaction(`#`)` `tpoints(`*vector*`)`
`npoints(`#`)` `delta(`#`)` `filename(`*filename*`)` `bootstrap(`*string*`)` `boot_reps(`#`)`
`analysis(`*string*`)` `analysis_level(`#`)` `graph(`*filename*`)` `detail`]

In the `gpscore` and `doseresponse` commands, the argument *varlist* represents the list of control variables, which are used to estimate the GPS. In the `doseresponse_model` command, the variable list consists of only two variables: the treatment variable (*treat_var*) and the GPS (*GPS_var*).

5 Options

We describe only the options for the `doseresponse` command, because they include all the options for the `gpscore` command and the `doseresponse_model` command. Therefore, all the options described in sections 5.1 and 5.2 apply to `doseresponse`, and we specify, if applicable, whether the option also applies to `gpscore` or `doseresponse_model`.

5.1 Required

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

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

`gpscore(`*newvar*`)` (`gpscore`) specifies the variable name for the estimated GPS.

`predict(`*newvar*`)` (`gpscore`) creates a new variable to hold the fitted values of the treatment variable.

`sigma(`*newvar*`)` (`gpscore`) creates a new variable to hold the maximum likelihood estimate of the conditional standard error of the treatment given the covariates.

`cutpoints(`*varname*`)` (`gpscore`) divides the set of potential treatment values, $\mathcal{T}$, into intervals according to the sample distribution of the treatment variable, cutting at *varname* quantiles.

`index(`*string*`)` (`gpscore`) specifies the representative point of the treatment variable at which the GPS has to be evaluated within each treatment interval. *string* identifies either the mean (*string* = `mean`) or a percentile (*string* = `p1`, ..., `p100`) of the treatment.

`nq_gps(#)` (`gpscore`) specifies that the values of the GPS evaluated at the representative point `index(`*string*`)` of each treatment interval have to be divided into # (# $\in \{1, \ldots, 100\}$) intervals, defined by the quantiles of the GPS evaluated at the representative point `index(`*string*`)`.

`dose_response(`*newvarlist*`)` specifies the variable name(s) for the estimated dose–response function(s).

(*Continued on next page*)

5.2 Optional

`t_transf(`*transformation*`)` (`gpscore`) specifies the transformation of the treatment variable used in estimating the GPS. The default *transformation* is the identity function. The supported transformations are the logarithmic transformation, `t_transf(ln)`; the zero-skewness log transformation, `t_transf(lnskew0)`; the zero-skewness Box–Cox transformation, `t_transf(bcskew0)`; and the Box–Cox transformation, `t_transf(boxcox)`. The Box–Cox transformation finds the maximum likelihood estimates of the parameters of the Box–Cox transform regressing the treatment variable `t(`*varname*`)` on the control variables listed in the input variable list.[3]

`normal_test(`*test*`)` (`gpscore`) specifies the goodness-of-fit test that `gpscore` will perform to assess the validity of the assumed normal distribution model for the treatment conditional on the covariates. By default, `gpscore` performs the Kolmogorov–Smirnov test (`normal_test(ksmirnov)`). Possible alternatives are the Shapiro–Francia test, `normal_test(sfrancia)`; the Shapiro–Wilk test, `normal_test(swilk)`; and the Stata skewness and kurtosis test for normality, `normal_test(sktest)`.

`norm_level(#)` (`gpscore`) sets the significance level of the goodness-of-fit test for normality. The default is `norm_level(0.05)`.

`test_varlist(`*varlist*`)` (`gpscore`) specifies that the extent of covariate balancing has to be inspected for each variable of *varlist*. The default *varlist* consists of the variables used to estimate the GPS. This option is useful when there are categorical variables among the covariates. `gpscore`, which is a regression-like command, requires that categorical variables are expanded into indicator (also called dummy) variable sets and that one dummy-variable set is dropped in estimating the GPS. However, the balancing test should also be performed on the omitted group. This can be done by using the `test_varlist(`*varlist*`)` option and by listing in *varlist* all the variables, including the complete set of indicator variables for each categorical covariate.

3. The problem is whether the treatment variable takes zero value. In such a case, the program continues, forcing a transformation of the treatment variable to take a suitable value. Specifically, we assume that $\ln(0) = 0$, t_transf$(0) = -1/\lambda$ if $\lambda > 0$, and t_transf$(0) = \ln(0) = 0$ if $\lambda = 0$, for t_transf = `bcskew0` or `boxcox`. Allowing for zero values of the treatment implies that untreated units might be included in the study. Because the GPS methods are designed for analyzing the effect of a treatment intensity, they specifically refer to the subpopulation of treated units. This implies that including untreated units might lead to misleading results.

`test(`*type*`)` (`gpscore`) specifies whether the balancing property has to be tested using either a standard two-sided t test (the default) or a Bayes-factor–based method (`test(Bayes_factor)`). The program informs the user if there is some evidence that the balancing property is satisfied. Recall that the test is performed for each single variable in `test_varlist(`*varlist*`)` and for each treatment interval. Specifically, let p be the number of control variables in `test_varlist(`*varlist*`)`, and let K be the number of the treatment intervals. We first calculate $p \times K$ values of the test statistic; then we select the worst value (the highest t value in modulus, or the lowest Bayes factor) and compare it with standard values. Table 1 shows the "order of magnitude" interpretations of the test statistics we consider.

Table 1. "Order of magnitude" interpretations of the test statistics

t value	Bayes factor (BF)*	Evidence for the balancing property (BP)
$\lvert t \rvert < 1.282$	$\text{BF} > 1.00$	Evidence supports the BP
$1.282 < \lvert t \rvert < 1.645$	$\sqrt{0.10} < \text{BF} < 1.00$	Very slight evidence against the BP
$1.645 < \lvert t \rvert < 1.960$	$0.10 < \text{BF} < \sqrt{0.10}$	Moderate evidence against the BP
$1.960 < \lvert t \rvert < 2.576$	$0.01 < \text{BF} < 0.10$	Strong to very strong evidence against the BP
$\lvert t \rvert > 2.576$	$\text{BF} < 0.01$	Decisive evidence against the BP

* The order of magnitude interpretations of the Bayes factor we applied were proposed by Jeffreys (1961).

`flag(#)` (`gpscore`) specifies that `gpscore` estimates the GPS without performing either a goodness-of-fit test for normality or a balancing test. The default # is 1, meaning that both the normal distribution model and the balancing property are tested; the default level is recommended. We introduced this option for practical reasons. Recall that `doseresponse` estimates the standard errors of the dose–response function by using bootstrap methods. In each bootstrap iteration, we want to reestimate the GPS without testing either the normality assumption or the balancing property.

`cmd(`*regression_cmd*`)` (`doseresponse_model`) defines the regression command to be used for estimating the conditional expectation of the outcome given the treatment and the GPS. The default for the outcome variable is `cmd(logit)` when there are two distinct values, `cmd(mlogit)` when there are 3–5 values, and `cmd(regress)` otherwise. The supported regression commands are `logit`, `probit`, `mlogit`, `mprobit`, `ologit`, `oprobit`, and `regress`.

`reg_type_t(`*type*`)` (`doseresponse_model`) defines the maximum power of the treatment variable in the polynomial function used to approximate the predictor for the conditional expectation of the outcome given the treatment and the GPS. The default *type* is `linear`, meaning that the predictor, $\psi(T, \widehat{R}; \alpha)$, is a linear function of the treatment. Alternatively, *type* can be `quadratic` or `cubic`.

`reg_type_gps(`*type*`)` (`doseresponse_model`) defines the maximum power of the estimated GPS in the polynomial function used to approximate the predictor for the conditional expectation of the outcome given the treatment and the GPS. The default *type* is `linear`, meaning that the predictor, $\psi(T, \widehat{R}; \alpha)$, is a linear function of the estimated GPS. Alternatively, *type* can be `quadratic` or `cubic`.

`interaction(#)` (`doseresponse_model`) specifies whether the model for the conditional expectation of the outcome given the treatment and the GPS has the interaction between treatment and GPS. The default # is 1, meaning that the interaction is included.

`tpoints(`*vector*`)` specifies that `doseresponse` estimates the average potential outcome for each level of the treatment in *vector*. By default, `doseresponse` creates a vector with the ith element equal to the ith observed treatment value. This option cannot be used with the `npoints(#)` option (see below).

`npoints(#)` specifies that `doseresponse` estimates the average potential outcome for 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 the `tpoints(`*vector*`)` option (see above).

`delta(#)` specifies that `doseresponse` also estimates the treatment-effect function considering a #-treatment gap, which is defined as $\mu(t + \#) - \mu(t)$. The default # is 0, meaning that `doseresponse` estimates only the dose–response function, $\mu(t)$.

`filename(`*filename*`)` specifies that the treatment levels specified through the `tpoints(`*vector*`)` option or the `npoints(#)` option, the estimated dose–response function, and, eventually, the estimated treatment-effect function, along with their standard errors (if calculated), be stored to a new file called *filename*.

`bootstrap(`*string*`)` specifies the use of bootstrap methods to derive standard errors and confidence intervals. By default, `doseresponse` does not apply bootstrap techniques. In such a case, no standard error is calculated. To activate this option, *string* should be set to `yes`.

`boot_reps(#)` specifies the number of bootstrap replications to be performed. The default is `boot_reps(50)`. This option produces an effect only if the `bootstrap()` option is set to `yes`.

`analysis(`*string*`)` specifies that `doseresponse` plots the estimated dose–response function(s) and, eventually, the estimated treatment-effect function(s), along with the corresponding confidence intervals if they are calculated with bootstrapping. By default, `doseresponse` plots only the estimated dose–response and treatment function(s). In order to plot confidence intervals, *string* has to be set to `yes`. If the user types `analysis(no)`, no plot is shown.

`analysis_level(`#`)` sets the confidence level of the confidence intervals. The default is `analysis_level(0.95)`.

`graph(`*filename*`)` stores the plots of the estimated dose–response function and the estimated treatment effects to a new file called *filename*. When the outcome variable is categorical, `doseresponse` creates a new file for each category i of the outcome variable and names it *filename_i*.

`detail` (`gpscore`) displays more detailed output. Specifically, this option specifies that `gpscore` shows the results of the goodness-of-fit test for normality, some summary statistics of the distribution of the GPS evaluated at the representative point of each treatment interval, and the results of the balancing test within each treatment interval. When this option is specified for `doseresponse`, the results of the regression of the outcome on the treatment and the GPS are also shown.

6 Example: The Imbens–Rubin–Sacerdote lottery sample

We use data from the survey of Massachusetts lottery winners; the data are described in detail in Imbens, Rubin, and Sacerdote (2001). We are interested in estimating the effect of the prize amount on subsequent labor earnings (from U.S. Social Security records). Although the lottery prize is obviously randomly assigned, substantial unit and item nonresponse led to a selected sample, where the amount of the prize is potentially correlated with background characteristics and potential outcomes. To remove such biases, we make the weak unconfoundedness assumption specifying that, conditional on the covariates, the lottery prize is independent of the potential outcomes.[4]

The sample we use in this analysis is the "winners" sample of 237 individuals who won a major prize in the lottery. The outcome of interest is `year6` (earnings six years after winning the lottery), and the treatment is `prize`, the prize amount. Control variables are age, gender, years of high school, years of college, winning year, number of tickets bought, work status after winning, and earnings s years before winning the lottery (with $s = 1, 2, \ldots, 6$).

We tried to replicate the results produced by Hirano and Imbens (2004) but have not been able to numerically replicate all their estimates because of restrictions of our

4. In this context, the nonignorability of the assignment mechanism is due to the presence of nonresponse. Therefore, saying that the unconfoundedness assumption allows us to remove all biases associated with differences in the observed covariates means that we are implicitly assuming that the outcome variable is missing at random (Rubin 1976).

programs. Specifically, our programs do not allow us to consider a function of the treatment variable or a function of the GPS in the estimation of the conditional expectation of the outcome, given the treatment and the GPS. However, we get qualitatively similar results.

6.1 Output from gpscore

We first choose the quantiles of the treatment variable to divide the sample into groups. Following Hirano and Imbens (2004), we divide the range of prizes into three treatment intervals, [0–23], (23–80], and (80–485]. Then we run `gpscore` using the specification applied by Hirano and Imbens (2004). The output looks like the following:

```
. use lotterydataset.dta
. qui generate cut = 23 if prize<=23
. qui replace cut = 80 if prize>23 & prize<=80
. qui replace cut = 485 if prize>80
. gpscore agew male ownhs owncoll tixbot workthen yearw yearm1 yearm2 yearm3
> yearm4 yearm5 yearm6, t(prize) gpscore(pscore) predict(hat_treat) sigma(sd)
> cutpoints(cut) index(p50) nq_gps(5) t_transf(ln) detail

Generalized Propensity Score
**************************************************
Algorithm to estimate the generalized propensity score
**************************************************

Estimation of the propensity score
The log transformation of the treatment variable prize is used
                            T
-------------------------------------------------------------

      Percentiles      Smallest
 1%     1.609438       .1301507
 5%     2.283851       .1301507
10%     2.420012       1.609438       Obs                 237
25%     2.835211        1.67818       Sum of Wgt.         237
50%      3.45783                      Mean           3.558185
                        Largest       Std. Dev.      .9553768
75%     4.143008       5.598792
90%     4.875426       5.720607       Variance       .9127448
95%     5.128892       5.778643       Skewness      -.0165889
99%     5.720607       6.183716       Kurtosis       3.452439

initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -4917.4112
rescale:       log likelihood = -480.91803
rescale eq:    log likelihood = -348.62357
Iteration 0:   log likelihood = -348.62357
  (output omitted)
Iteration 4:   log likelihood = -307.68186
```

```
                                                Number of obs   =        237
                                                Wald chi2(13)   =      37.22
Log likelihood = -307.68186                     Prob > chi2     =     0.0004
```

T	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
eq1						
agew	.0151905	.0048563	3.13	0.002	.0056724	.0247086
male	.4379826	.1351124	3.24	0.001	.1731672	.702798
ownhs	.0192025	.060835	0.32	0.752	-.1000319	.1384368
owncoll	.0372805	.0397666	0.94	0.349	-.0406607	.1152217
tixbot	.0043423	.0182546	0.24	0.812	-.031436	.0401206
workthen	.1270879	.1645602	0.77	0.440	-.1954442	.44962
yearw	-.0014367	.0464566	-0.03	0.975	-.09249	.0896166
yearm1	.0062064	.010379	0.60	0.550	-.014136	.0265488
yearm2	-.0123161	.0162758	-0.76	0.449	-.044216	.0195839
yearm3	.0119446	.0166256	0.72	0.472	-.0206411	.0445302
yearm4	.0242245	.0158217	1.53	0.126	-.0067855	.0552344
yearm5	-.0216437	.0153635	-1.41	0.159	-.0517555	.0084682
yearm6	-.0050021	.0110455	-0.45	0.651	-.0266509	.0166467
_cons	2.315546	.4693959	4.93	0.000	1.395547	3.235545
eq2						
_cons	.886297	.040709	21.77	0.000	.806509	.9660851

```
Test for normality of the disturbances

Kolmogorov-Smirnov equality-of-distributions test
Normal Distribution of the disturbances

One-sample Kolmogorov-Smirnov test against theoretical distribution
          normal((res_etreat - r(mean))/sqrt(r(Var)))
 Smaller group       D       P-value  Corrected
 ---------------------------------------------
 res_etreat:         0.0517    0.281
 Cumulative:        -0.0420    0.434
 Combined K-S:       0.0517    0.550      0.517

The assumption of Normality is statistically satisfied at .05 level

           Estimated generalized propensity score
-------------------------------------------------------------

      Percentiles      Smallest
 1%     .0131817       .0003053
 5%     .0869414       .0011738
10%     .1272663       .0131817       Obs                 237
25%     .2255553       .0163113       Sum of Wgt.         237

50%     .3536221                      Mean           .3196603
                       Largest        Std. Dev.      .1222106
75%     .4343045       .4500003
90%     .4481351       .4500911       Variance       .0149354
95%     .4497166        .450096       Skewness      -.7723501
99%     .4500911       .4501086       Kurtosis       2.510499

******************************************
End of the algorithm to estimate the gpscore
******************************************
```

```
*******************************************************************************
The set of the potential treatment values is divided into 3 intervals
The values of the gpscore evaluated at the representative point of each
treatment interval are divided into 5 intervals
*******************************************************************************

************************************************************
Summary statistics of the distribution of the GPS evaluated
at the representative point of each treatment interval
************************************************************
```

Variable	Obs	Mean	Std. Dev.	Min	Max
gps_1	237	.262852	.0956436	.0583948	.4486237
Variable	Obs	Mean	Std. Dev.	Min	Max
gps_2	237	.4178101	.0373217	.2433839	.4501224
Variable	Obs	Mean	Std. Dev.	Min	Max
gps_3	237	.1814998	.088236	.0181741	.4141454

```
*******************************************************************************
Test that the conditional mean of the pre-treatment variables given the
generalized propensity score is not different between units who belong to a
particular treatment interval and units who belong to all other treatment
intervals
*******************************************************************************
Treatment Interval No 1 - [1.139000058174133, 22.98200035095215]
```

	Mean Difference	Standard Deviation	t-value
agew	-.25322	1.814	-.13959
male	.04799	.04246	1.1304
ownhs	.15044	.156	.96433
owncoll	.20765	.23738	.87476
tixbot	.33298	.48448	.68729
workthen	.00154	.05608	.0275
yearw	.00156	.19135	.00813
yearm1	.33117	1.9052	.17382
yearm2	.90872	1.7719	.51284
yearm3	1.2445	1.6756	.74274
yearm4	.74998	1.5625	.47999
yearm5	.96299	1.7175	.5607
yearm6	1.4414	1.7774	.81098

```
Treatment Interval No 2 - [23.08799934387207, 79.11299896240234]
              Mean        Standard
              Difference  Deviation   t-value
       agew  -.13308       1.8294    -.07275
       male  -.03419        .0657    -.52041
      ownhs   -.2294       .13927    -1.6471
    owncoll  -.20996       .21228    -.98908
     tixbot  -.26933       .43812    -.61474
   workthen   .03013       .05266     .57227
      yearw  -.32817       .17008    -1.9295
     yearm1   .51467       1.7741      .2901
     yearm2   .23703       1.7038     .13912
     yearm3   .41572       1.6656     .24959
     yearm4   .46856        1.571     .29826
     yearm5  -.00903       1.6242    -.00556
     yearm6  -.33587       1.6445    -.20423
Treatment Interval No 3 - [82.98699951171875, 484.7900085449219]
              Mean        Standard
              Difference  Deviation   t-value
       agew  -1.7504       2.3202    -.75444
       male  -.04742       .06211    -.76342
      ownhs   .34062        .1914     1.7796
    owncoll   .23199       .28116     .82512
     tixbot  -.03159       .56716     -.0557
   workthen  -.07006       .07448    -.94069
      yearw    .3672       .22613     1.6238
     yearm1  -.63678       1.9428    -.32777
     yearm2  -.83409       1.8356    -.45441
     yearm3  -1.2074       1.7322    -.69707
     yearm4   -1.351       1.5982    -.84534
     yearm5  -1.6137       1.8792     -.8587
     yearm6  -2.2111       1.8615    -1.1878
According to a standard two-sided t-test:
Moderate evidence against the balancing property
The balancing property is satisfied at level 0.05
```

This output is the most detailed we can have because we specified the `detail` option. When this option is not specified, some information is omitted from the output. Specifically, the results of the goodness-of-fit test for normality, the summary statistics of the distribution of the GPS evaluated at the representative point of each treatment interval, and the results of the balancing test within each treatment interval are not shown. In such a case, the program provides only short sentences informing the user whether the normal distribution model and the balancing property are statistically satisfied.

6.2 Output from doseresponse

Before running `doseresponse`, we have to decide about the treatment levels, which estimate the average potential outcome. Following Hirano and Imbens (2004), we focus on the values $10, 20, \ldots, 100$, which we store to a 10-dimensional vector named `tp` (see below). The output from running `doseresponse` is as follows:

```
. use lotterydataset.dta, clear
. qui generate cut = 23 if prize<=23
. qui replace cut = 80 if prize>23 & prize<=80
. qui replace cut = 485 if prize>80
. matrix define tp = (10\20\30\40\50\60\70\80\90\100)
. doseresponse agew ownhs male tixbot owncoll workthen yearw yearm1 yearm2
> yearm3 yearm4 yearm5 yearm6, outcome(year6) t(prize) gpscore(pscore)
> predict(hat_treat) sigma(sd) cutpoints(cut) index(p50) nq_gps(5)
> t_transf(ln) dose_response(dose_response) tpoints(tp) delta(1)
> reg_type_t(quadratic) reg_type_gps(quadratic) interaction(1) bootstrap(yes)
> boot_reps(100) filename("output") analysis(yes) graph("graph_output") detail
*****************************************
ESTIMATE OF THE GENERALIZED PROPENSITY SCORE
*****************************************
  (output omitted)
The outcome variable ``year6´´ is a continuous variable
The regression model is: Y = T + T^2 + GPS +  GPS^2 + T*GPS
```

Source	SS	df	MS		
Model	2945.92738	5	589.185477	Number of obs	= 202
				F(5, 196)	= 3.01
Residual	38378.9633	196	195.811037	Prob > F	= 0.0122
				R-squared	= 0.0713
				Adj R-squared	= 0.0476
Total	41324.8907	201	205.596471	Root MSE	= 13.993

year6	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
prize	-.2254371	.0748156	-3.01	0.003	-.3729839	-.0778902
prize_sq	.0003537	.0001669	2.12	0.035	.0000245	.0006828
pscore	-103.3373	48.37076	-2.14	0.034	-198.7312	-7.943281
pscore_sq	131.949	79.40569	1.66	0.098	-24.65021	288.5482
prize_pscore	.5499933	.2197661	2.50	0.013	.1165835	.9834031
_cons	31.26845	6.955419	4.50	0.000	17.55138	44.98552

```
Bootstrapping of the standard errors
..................................................................................
> ..................

The program is drawing graphs of the output
This operation may take a while
(file graph_output.gph saved)

End of the Algorithm
```

The estimated coefficients of the regression of the outcome, earnings six years after winning the lottery, the prize, and the score are shown because we have required a detailed output. Otherwise, `doseresponse` provides only a graphic output, such as that shown in figure 1. Figure 1 shows both the estimated dose–response function and the estimated treatment-effect function, which can be interpreted as a derivate, because we have specified a treatment gap equal to 1 (`delta(1)`). Only information concerning the GPS estimation is provided when `detail` is not specified and the `analysis()` option is set to `no`.

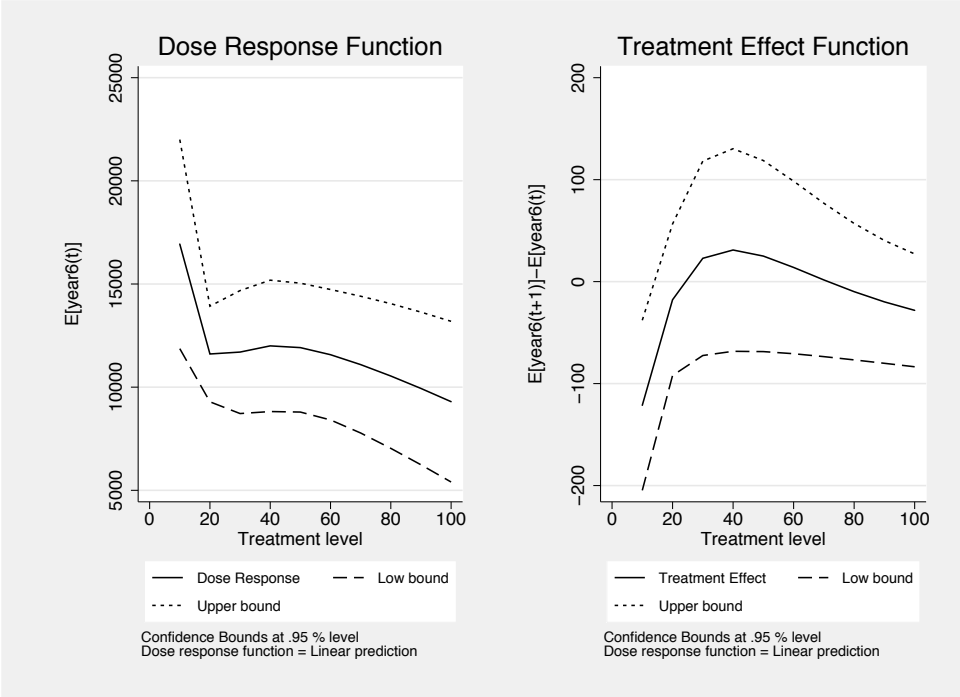

Figure 1. Estimated dose–response function, estimated derivative, and 95% confidence bands

(*Continued on next page*)

The results generated by `doseresponse` are stored in a new Stata file, which we have named `output`. This file has 10 observations and 6 variables: `treatment_level`, containing the treatment levels, at which we estimate the average potential outcome; `treatment_level_plus`, containing the #-shifted treatment levels, where # is equal to 1; `dose_response`, the estimated dose–response function; `se_dose_response_bs`, the standard errors of the estimated dose–response function; `diff_dose_response`, the estimated treatment-effect function; and `se_diff_dose_response_bs`, the standard errors of the estimated treatment-effect function. The graphic output is also stored to a new file, which we have named `graph_output`.

7 Acknowledgments

We thank Fabrizia Mealli, Guido Imbens, and Keisuke Hirano for their insightful suggestions and discussions, and Guido Imbens and Keisuke Hirano for providing the data.

8 References

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. 2007. Application of the generalized propensity score. Evaluation of public contributions to Piedmont enterprises. POLIS Working Paper 80, University of Eastern Piedmont.

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. West Sussex, England: Wiley InterScience.

Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 8: 945–960.

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.

Jeffreys, H. 1961. *Theory of Probability*. 3rd ed. Oxford: Oxford University Press.

Leuven, E., and B. Sianesi. 2003. psmatch2: Stata module to perform full Mahalanobis and propensity score matching, common support graphing, and covariate imbalance testing. Boston College Department of Economics, Statistical Software Components. Downloadable from http://ideas.repec.org/c/boc/bocode/s432001.html.

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. 1976. Inference and missing data. *Biometrika* 63: 581–592.

About the authors

Michela Bia is a research assistant at Laboratorio Revelli, Centre for Employment Studies, Collegio Carlo Alberto, Turin, Italy.

Alessandra Mattei is a statistics research assistant in the Department of Statistics, "Giuseppe Parenti", University of Florence, Italy. On February 25, 2008, she won a competitive exam to work as an assistant professor of statistics in the faculty of psychology at the University of Florence.