1 Introduction

The Toolkit for Weighting and Analysis of Nonequivalent Groups, \texttt{twang}, contains a set of functions and procedures to support causal modeling of observational data through the estimation and evaluation of propensity scores and associated weights. This package was developed in 2004. After extensive use, it received a major update in 2012.

The propensity score is the probability that a particular case would be assigned or exposed to a treatment condition. Rosenbaum & Rubin (1983) showed that knowing the propensity score is sufficient to separate the effect of a treatment on an outcome from observed confounding factors that influence both treatment assignment and outcomes, provided the necessary conditions hold. The propensity score has the balancing property that given the propensity score the distribution of features for the treatment cases is the same as that for the control cases. While the treatment selection probabilities are generally not known, good estimates of them can be effective at diminishing or eliminating confounds between pretreatment group differences and treatment outcomes in the estimation of treatment effects.

There are now numerous propensity scoring methods in the literature. They differ in how they estimate the propensity score (e.g. logistic regression, CART), the target estimand (e.g. treatment effect on the treated, population treatment effect), and how they utilize the resulting estimated propensity scores (e.g. stratification, matching, weighting, doubly robust estimators). We originally developed the \texttt{twang} package with a particular process in mind, generalized boosted regression, to estimate the propensity scores and weighting of the comparison cases to estimate a treatment effect on the treated. However, we have updated the package to also meaningfully handle the case where interest lies in using the population weights (e.g., weighting of comparison and treatment cases to estimate the population average treatment effect.) The main workhorse of \texttt{twang} is the \texttt{ps()} function which implements generalized boosted regression modeling to estimate the propensity scores. However, the framework of the package is flexible enough to allow the user to use propensity score estimates from other methods and to assess the usefulness of those estimates for ensuring equivalence (or “balance”) in the pretreatment covariate distributions of treatment and control groups using tools from the package. The same set of functions is also useful for other tasks such as non-response weighting, as discussed in Section 4.

The \texttt{twang} package aims to compute from the data estimates of the propensity scores which yield accurate causal effect estimates, check the quality of the weights by assessing whether or
not they have the balancing properties that we expect in theory, and use them in computing
treatment effect estimates.

2 An ATT example to start

If you have not already done so, install twang by typing install.packages("twang"). twang
relies on other R packages, especially gbm, survey, and lattice. You may have to run install.packages() for these as well if they are not already installed. You will only need to do
this step once. In the future running update.packages() regularly will ensure that you have
the latest versions of the packages, including bug fixes and new features.

To start using twang, first load the package. You will have to do this step once for each R
session that you run. We also set the seed of R’s pseudo random number generator so that the
results are exactly replicable. (There is a stochastic element to the fitting of the propensity score
models.)

> library(twang)
> set.seed(1)

To demonstrate the package we utilize data from Lalonde’s National Supported Work Demon-
nsndata.html). This dataset is provided with the twang package.

> data(lalonde)

R can read data from many other sources. The manual “R Data Import/Export,” available

For the lalonde dataset, the variable treat is the 0/1 treatment indicator, 1 indicates “treat-
mant” by being part of the National Supported Work Demonstration and 0 indicates “comparison”
cases drawn from the Current Population Survey. In order to estimate a treatment effect for this
demonstration program that is unbiased by pretreatment group differences on other observed
covariates, we include these covariates in a propensity score model of treatment assignment:
age, education, black, Hispanic, having no degree, married, earnings in 1974 (pretreatment), and
earnings in 1975 (pretreatment). Note that we specify no outcome variables at this time. The
ps() function is the primary method in twang for estimating propensity scores. This step is
computationally intensive and can take a few minutes.

> ps.lalonde <- ps(treat ~ age + educ + black + hispan + nodegree +
+       married + re74 + re75,
+       data = lalonde,
+       n.trees=5000,
+       interaction.depth=2,
+       shrinkage=0.01,
+       perm.test.iters=0,
+       stop.method=c("es.mean","ks.max"),
+       estimand = "ATT",
+       verbose=FALSE)

The arguments to ps() require some discussion. The first argument specifies a formula
indicating that treat is the 0/1 treatment indicator and that the propensity score model should
predict treat from the eight covariates listed there separated by “+”. The “+” does not mean
that these variables are being summed nor does it mean that the model is linear. This is
just R's notation for including predictor variables in the model. There is no need to specify interaction terms in the formula. There is also no need — and it can be counterproductive — to create indicator, or "dummy coded," variables to represent categorical covariates, provided the categorical variables are stored as a factor or as ordered (see help(factor) for more details).

The next argument, data, indicates the dataset. n.trees, interaction.depth, and shrinkage are parameters for the gbm model that ps() computes and stores. The resulting gbm object describes a family of candidate propensity score models indexed by the number of GBM iterations from one to n.trees. The argument n.trees is the maximum number of iterations that gbm will run. ps() will issue a warning if the estimated optimal number of iterations is too close to the bound selected in this argument because it indicates that balance may improve if more complex models (i.e., those with more trees) are considered. The user should increase n.trees or decrease shrinkage if this warning appears.

perm.test.iters specifies whether p-values for KS statistics should be calculated using Monte Carlo methods, which is slow but can be accurate, or estimated using an analytic approximation that is fast, but produces poor estimates in the presence of many ties. If perm.test.iters=0 is called, then analytic approximations are used. If perm.test.iters=500 is called, then 500 Monte Carlo trials are run to establish the reference distribution of KS statistics for each covariate. Higher numbers of trials will produce more precise p-values.

The estimand argument is used to indicate whether the analyst is interested in estimating the average treatment effect (ATE) or the average treatment effect on the treated (ATT), as we do above. ATE addresses the question of how outcomes would differ if everyone in the sample were given the treatment versus everyone being given the control (Wooldridge, 2002). ATT, on the other hand, estimates the analogous quantity averaging only over the subjects who were actually treated. The estimand argument was added to the 2012 revision of the package which integrated ATE weighting into the package and the ps function estimate of the propensity score.

The stop.method argument specifies a set (or sets) of rules and measures for assessing the balance, or equivalence, established on the pretreatment covariates of the treatment and weighted control group. The ps function selects the optimal number of GBM iterations to minimize the differences between the treatment and control groups as measured by the rules of the given stop.method object. The package includes four built-in stop.method objects. They are es.mean, es.max, ks.mean, and ks.max. The four stopping rules are defined by two components: a balance metric for covariates and rule for summarizing across covariates. The balance metric summarizes the difference between two univariate distributions of a single pre-treatment variable (e.g., age). The default stopping rules in twang use two balance metrics: absolute standardized bias (also referred to as the absolute standardized mean difference of the Effect Size) and the Kolmogorov-Smirnov (KS) statistic. The stopping rule use two different rules for summarizing across covariates: the mean of the covariate balance metrics ("mean") or the maximum of the balance metrics ("max"). The first piece of the stopping rule name identifies the balance metric (ES or KS) and the second piece specifies the method for summarizing across covariates. The balance metric summarizes the difference between two univariate distributions of a single pre-treatment variable (e.g., age). The default stopping rules in twang use two balance metrics: absolute standardized bias (also referred to as the absolute standardized mean difference of the Effect Size) and the Kolmogorov-Smirnov (KS) statistic. The stopping rule use two different rules for summarizing across covariates: the mean of the covariate balance metrics ("mean") or the maximum of the balance metrics ("max"). The first piece of the stopping rule name identifies the balance metric (ES or KS) and the second piece specifies the method for summarizing across balance metrics. For instance, es.mean uses the effect size or the absolute standardized bias and summarizes across variables with the mean and the ks.max uses the KS statistics to assess balances and summarizes using the maximum across variables and the other two stopping rules use the remaining two combinations of balance metrics and summary statistics. The variable distributions used in the balance metrics depend on whether we are interested in estimating the ATT or ATE, and correct specification of these distributions is set automatically by the specification of the estimand in the ps() function.

Having fit the ps object, the analyst should perform several diagnostic checks before estimating the causal effect in question. The first of these diagnostic checks makes sure that the specified value of n.trees allowed GBM to explore sufficiently complicated models. We can do
this quickly with the `plot()` function. As a default, the `plot()` function applied to a `ps` object gives the balance measures as a function of the number of iterations in the GBM algorithm, with higher iterations corresponding to more complicated fitted models. In the example below, 2127 iterations minimized the average effect size difference and 1756 iterations minimized the largest of the eight Kolmogorov-Smirnov (KS) statistics computed for the covariates. If it appears that additional iterations would be likely to result in lower values of the balance statistic, `n.trees` should be increased. However, after a point, additional complexity typically makes the balance worse, as in the example below. This figure also gives information on how compatible two or more stopping rules are: if the minima for multiple stopping rules under consideration are near one another, the results should not be sensitive to which stopping rule one uses for the final analysis. See Section 5.3 for a discussion of these and other balance measures.

```r
> plot(ps.lalonde)
```

If we wish to focus on only one stopping rule, the plotting commands also take a `subset` argument.

---

1In versions 1.0.x of the `twang` package, the `ps` function itself included some plotting functions. This is no longer the case (and the function no longer includes a `plots` argument); these functions have been moved to the generic `plot()` function.
The \texttt{gbm} package has various tools for exploring the relationship between the covariates and the treatment assignment indicator if these are of interest. \texttt{summary()} computes the relative influence of each variable for estimating the probability of treatment assignment. The \texttt{gbm} estimates depend on the number of iterations at which the \texttt{gbm} model is evaluated, which is specified by the \texttt{n.trees} argument in the \texttt{summary} method for \texttt{gbm}. In this example, we choose the number of iterations to be the optimal number for minimizing the largest of the KS statistics. This value can be found in the \texttt{ps.lalonde$desc$ks.max.ATT$n.trees}. Figure 1 shows the barchart of the relative influence and is produced when \texttt{plot=TRUE} in the call to \texttt{summary()}. 

```r
> summary(ps.lalonde$gbm.obj, + n.trees=ps.lalonde$desc$ks.max.ATT$n.trees, + plot=FALSE)
```

<table>
<thead>
<tr>
<th>var</th>
<th>rel.inf</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>57.8309</td>
</tr>
<tr>
<td>age</td>
<td>16.5159</td>
</tr>
<tr>
<td>re74</td>
<td>15.6136</td>
</tr>
<tr>
<td>re75</td>
<td>3.5801</td>
</tr>
<tr>
<td>married</td>
<td>3.0214</td>
</tr>
<tr>
<td>educ</td>
<td>2.9185</td>
</tr>
<tr>
<td>nodegree</td>
<td>0.4388</td>
</tr>
<tr>
<td>hispan</td>
<td>0.0807</td>
</tr>
</tbody>
</table>

2.1 Assessing “balance” using balance tables

Having estimated the propensity scores, \texttt{bal.table()} produces a table that shows how well the resulting weights succeed in manipulating the control group so that its weighted pretreatment characteristics match, or balance, those of the unweighted treatment group if \texttt{estimand = "ATT"} or the control and treatment groups so that the weighted pretreatment characteristics match, or balance, with one another if \texttt{estimand = "ATE"}. By default, the \texttt{bal.table()} function uses the value of \texttt{estimand} set with the \texttt{ps()} function call. For example, in the analysis we
Figure 1: Relative influence of the covariates on the estimated propensity score

set estimand = "ATT" when calling ps() to estimate the propensity scores and the resulting ps.object, ps.lalonde, contains an element "estimand" which takes the value "ATT". The function bal.table() checks this value and automatically uses ATT weights when checking balance and comparing the distributions of pre-treatment variables for the weighted control group with those from the unweighted treatment group.
> lalonde.balance <- bal.table(ps.lalonde)
> lalonde.balance

\$unw

<table>
<thead>
<tr>
<th></th>
<th>tx.mn</th>
<th>tx.sd</th>
<th>ct.mn</th>
<th>ct.sd</th>
<th>std.eff.sz</th>
<th>stat</th>
<th>p</th>
<th>ks</th>
<th>ks.pval</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>25.816</td>
<td>7.155</td>
<td>28.030</td>
<td>10.787</td>
<td>-0.309</td>
<td>-2.994</td>
<td>0.003</td>
<td>0.158</td>
<td>0.003</td>
</tr>
<tr>
<td>educ</td>
<td>10.346</td>
<td>2.011</td>
<td>10.235</td>
<td>2.855</td>
<td>0.055</td>
<td>0.547</td>
<td>0.584</td>
<td>0.111</td>
<td>0.074</td>
</tr>
<tr>
<td>black</td>
<td>0.843</td>
<td>0.365</td>
<td>0.203</td>
<td>0.403</td>
<td>1.757</td>
<td>19.371</td>
<td>0.000</td>
<td>0.640</td>
<td>0.000</td>
</tr>
<tr>
<td>hispan</td>
<td>0.059</td>
<td>0.237</td>
<td>0.142</td>
<td>0.350</td>
<td>-0.349</td>
<td>-3.413</td>
<td>0.001</td>
<td>0.083</td>
<td>0.317</td>
</tr>
<tr>
<td>nodelg</td>
<td>0.708</td>
<td>0.456</td>
<td>0.597</td>
<td>0.491</td>
<td>0.244</td>
<td>2.716</td>
<td>0.007</td>
<td>0.111</td>
<td>0.074</td>
</tr>
<tr>
<td>married</td>
<td>0.189</td>
<td>0.393</td>
<td>0.513</td>
<td>0.500</td>
<td>-0.824</td>
<td>-8.607</td>
<td>0.000</td>
<td>0.324</td>
<td>0.000</td>
</tr>
<tr>
<td>re74</td>
<td>2095.574</td>
<td>4886.620</td>
<td>5619.237</td>
<td>6788.751</td>
<td>-0.721</td>
<td>-7.254</td>
<td>0.000</td>
<td>0.447</td>
<td>0.000</td>
</tr>
<tr>
<td>re75</td>
<td>1532.055</td>
<td>3219.251</td>
<td>2466.484</td>
<td>3291.996</td>
<td>-0.290</td>
<td>-3.282</td>
<td>0.001</td>
<td>0.288</td>
<td>0.000</td>
</tr>
</tbody>
</table>

\$es.mean.ATT

<table>
<thead>
<tr>
<th></th>
<th>tx.mn</th>
<th>tx.sd</th>
<th>ct.mn</th>
<th>ct.sd</th>
<th>std.eff.sz</th>
<th>stat</th>
<th>p</th>
<th>ks</th>
<th>ks.pval</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>25.816</td>
<td>7.155</td>
<td>25.802</td>
<td>7.279</td>
<td>0.002</td>
<td>0.015</td>
<td>0.988</td>
<td>0.122</td>
<td>0.892</td>
</tr>
<tr>
<td>educ</td>
<td>10.346</td>
<td>2.011</td>
<td>10.573</td>
<td>2.089</td>
<td>-0.113</td>
<td>-0.706</td>
<td>0.480</td>
<td>0.099</td>
<td>0.977</td>
</tr>
<tr>
<td>black</td>
<td>0.843</td>
<td>0.365</td>
<td>0.842</td>
<td>0.365</td>
<td>0.003</td>
<td>0.027</td>
<td>0.978</td>
<td>0.001</td>
<td>1.000</td>
</tr>
<tr>
<td>hispan</td>
<td>0.059</td>
<td>0.237</td>
<td>0.042</td>
<td>0.202</td>
<td>0.072</td>
<td>0.804</td>
<td>0.421</td>
<td>0.017</td>
<td>1.000</td>
</tr>
<tr>
<td>nodelg</td>
<td>0.708</td>
<td>0.456</td>
<td>0.609</td>
<td>0.489</td>
<td>0.218</td>
<td>0.967</td>
<td>0.334</td>
<td>0.099</td>
<td>0.977</td>
</tr>
<tr>
<td>married</td>
<td>0.189</td>
<td>0.393</td>
<td>0.189</td>
<td>0.393</td>
<td>0.002</td>
<td>0.012</td>
<td>0.990</td>
<td>0.001</td>
<td>1.000</td>
</tr>
<tr>
<td>re74</td>
<td>2095.574</td>
<td>4886.620</td>
<td>1556.930</td>
<td>3801.566</td>
<td>0.110</td>
<td>1.027</td>
<td>0.305</td>
<td>0.066</td>
<td>1.000</td>
</tr>
<tr>
<td>re75</td>
<td>1532.055</td>
<td>3219.251</td>
<td>1211.575</td>
<td>2647.615</td>
<td>0.100</td>
<td>0.833</td>
<td>0.405</td>
<td>0.103</td>
<td>0.969</td>
</tr>
</tbody>
</table>

\$ks.max.ATT

<table>
<thead>
<tr>
<th></th>
<th>tx.mn</th>
<th>tx.sd</th>
<th>ct.mn</th>
<th>ct.sd</th>
<th>std.eff.sz</th>
<th>stat</th>
<th>p</th>
<th>ks</th>
<th>ks.pval</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>25.816</td>
<td>7.155</td>
<td>25.764</td>
<td>7.408</td>
<td>0.007</td>
<td>0.055</td>
<td>0.956</td>
<td>0.107</td>
<td>0.919</td>
</tr>
<tr>
<td>educ</td>
<td>10.346</td>
<td>2.011</td>
<td>10.572</td>
<td>2.140</td>
<td>-0.113</td>
<td>-0.712</td>
<td>0.477</td>
<td>0.107</td>
<td>0.919</td>
</tr>
<tr>
<td>black</td>
<td>0.843</td>
<td>0.365</td>
<td>0.835</td>
<td>0.371</td>
<td>0.022</td>
<td>0.187</td>
<td>0.852</td>
<td>0.008</td>
<td>1.000</td>
</tr>
<tr>
<td>hispan</td>
<td>0.059</td>
<td>0.237</td>
<td>0.043</td>
<td>0.203</td>
<td>0.069</td>
<td>0.779</td>
<td>0.436</td>
<td>0.016</td>
<td>1.000</td>
</tr>
<tr>
<td>nodelg</td>
<td>0.708</td>
<td>0.456</td>
<td>0.601</td>
<td>0.490</td>
<td>0.235</td>
<td>1.100</td>
<td>0.272</td>
<td>0.107</td>
<td>0.919</td>
</tr>
<tr>
<td>married</td>
<td>0.189</td>
<td>0.393</td>
<td>0.199</td>
<td>0.400</td>
<td>-0.024</td>
<td>-0.169</td>
<td>0.866</td>
<td>0.010</td>
<td>1.000</td>
</tr>
<tr>
<td>re74</td>
<td>2095.574</td>
<td>4886.620</td>
<td>1673.666</td>
<td>3944.600</td>
<td>0.086</td>
<td>0.800</td>
<td>0.424</td>
<td>0.054</td>
<td>1.000</td>
</tr>
<tr>
<td>re75</td>
<td>1532.055</td>
<td>3219.251</td>
<td>1257.242</td>
<td>2674.922</td>
<td>0.085</td>
<td>0.722</td>
<td>0.471</td>
<td>0.094</td>
<td>0.971</td>
</tr>
</tbody>
</table>

bal.table() returns information on the pretreatment covariates before and after weighting. The object is a list with named components, one for an unweighted analysis (named unw) and one for each stop.method specified, here es.mean and ks.max. McCaffrey et al (2004) essentially used es.mean for the analyses, but our more recent work has sometimes used ks.max. See Section 5.3 for a more detailed description of these choices.

If there are missing values (represented as NA) in the covariates, twang will attempt to construct weights that also balance rates of missingness in the treatment and control arms. In this case, the bal.table() will have an extra row for each variable that has missing entries.

The columns of the table consist of the following items:

tx.mn, ct.mn The treatment means and the control means for each of the variables. The unweighted table (unw) shows the unweighted means. For each stopping rule the means are weighted using weights corresponding to the gbm model selected by ps() using the stopping rule. When estimand = "ATT" the weights for the treatment group always equal 1 for all cases and there is no difference between unweighted and propensity score weighted tx.mn.
**tx.sd, ct.sd** The propensity score weighted treatment and control groups’ standard deviations for each of the variables. The unweighted table (unw) shows the unweighted standard deviations.

**std.eff.sz** The standardized effect size, defined as the treatment group mean minus the control group mean divided by the treatment group standard deviation if \( \text{estimand} = \text{"ATT"} \) or divided by the pooled sample (treatment and control) standard deviation if \( \text{estimand} = \text{"ATE"} \). (In discussions of propensity scores this value is sometimes referred to as “standardized bias”.) Occasionally, lack of treatment group or pooled sample variance on a covariate results in very large (or infinite) standardized effect sizes. For purposes of analyzing mean effect sizes across multiple covariates, we set all standardized effect sizes larger than 500 to \( \text{NA} \) (missing values).

**stat, p** Depending on whether the variable is continuous or categorical, \( \text{stat} \) is a t-statistic or a \( \chi^2 \) statistic. \( p \) is the associated p-value.

**ks, ks.pval** The Kolmogorov-Smirnov test statistic and its associated p-value. P-values for the KS statistics are either derived from Monte Carlo simulations or analytic approximations, depending on the specifications made in the `perm.test.iters` argument of the `ps` function. For categorical variables this is just the \( \chi^2 \) test p-value.

Components of these tables are useful for demonstrating that pretreatment differences between groups on observed variables have been eliminated using the weights. The `xtable` package aids in formatting for \( \LaTeX \) and Word documents. Table 1 shows the results for `ks.max` reformatted for a \( \LaTeX \) document. For Word documents, paste the \( \LaTeX \) description of the table into a Word document, highlight it and use Word tools to convert the text to a table using “&” as the separator.

```r
> library(xtable)
> pretty.tab <- lalonde.balance$ks.max.ATT[,c("tx.mn","ct.mn","ks")]
> pretty.tab <- cbind(pretty.tab, lalonde.balance$unw[, "ct.mn"])
> names(pretty.tab) <- c("E(Y1|t=1)", "E(Y0|t=1)", "KS", "E(Y0|t=0)")
> xtable(pretty.tab,
+ caption = "Balance of the treatment and comparison groups",
+ label = "tab:balance",
+ digits = c("1", "r", "r", "r", "r"),
+ align=c("l", "r", "r", "r", "r")
)`

|         | E(Y1|t=1) | E(Y0|t=1) | KS  | E(Y0|t=0) |
|---------|----------|----------|-----|-----------|
| age     | 25.82    | 25.76    | 0.11| 28.03     |
| educ    | 10.35    | 10.57    | 0.11| 10.23     |
| black   | 0.84     | 0.83     | 0.01| 0.20      |
| hispan  | 0.06     | 0.04     | 0.02| 0.14      |
| nodelree| 0.71     | 0.60     | 0.11| 0.60      |
| married | 0.19     | 0.20     | 0.01| 0.51      |
| re74    | 2095.57  | 1673.67  | 0.05| 5619.24   |
| re75    | 1532.06  | 1257.24  | 0.09| 2466.48   |
```

Table 1: Balance of the treatment and comparison groups.

The `summary()` method for \( \text{ps} \) objects offers a compact summary of the sample sizes of the groups and the balance measures. If `perm.test.iters>0` was used to create the `ps` object, then
Monte Carlo simulation is used to estimate p-values for the maximum KS statistic that would be expected across the covariates, had individuals with the same covariate values been assigned to groups randomly. Thus, a p-value of 0.04 for `max.ks.p` indicates that the largest KS statistic found across the covariates is larger than would be expected in 96% of trials in which the same cases were randomly assigned to groups.

```r
> summary(ps.lalonde)

  n.treat n.ctrl ess.treat ess.ctrl  max.es
  unw   185 429  185 429.00000  1.7567745
  es.mean.ATT  185 429  185 22.96430  0.2177817
  ks.max.ATT  185 429  185 27.05472  0.2348846

  mean.es max.ks max.ks.p mean.ks iter
  unw 0.56872589  0.6404460  NA  0.27024507 NA
  es.mean.ATT  0.07746175  0.1223384  NA  0.06361021  2127
  ks.max.ATT  0.08025994  0.1070761  NA  0.06282432  1756
```

In general, weighted means can have greater sampling variance than unweighted means from a sample of equal size. The effective sample size (ESS) of the weighted comparison group captures this increase in variance as

\[
\text{ESS} = \frac{\left(\sum_{i \in C} w_i\right)^2}{\sum_{i \in C} w_i^2}.
\]  

(1)

The ESS is approximately the number of observations from a simple random sample that yields an estimate with sampling variation equal to the sampling variation obtained with the weighted comparison observations. Therefore, the ESS will give an estimate of the number of comparison participants that are comparable to the treatment group when `estimand = "ATT"`. The ESS is an accurate measure of the relative size of the variance of means when the weights are fixed or they are uncorrelated with outcomes. Otherwise the ESS underestimates the effective sample size (Little & Vartivarian, 2004). With propensity score weights, it is rare that weights are uncorrelated with outcomes. Hence the ESS typically gives a lower bound on the effective sample size, but it still serves as a useful measure for choosing among alternative models and assessing the overall quality of a model, even if it provides a possibly conservative picture of the loss in precision due to weighting.

The `ess.treat` and `ess.ctrl` columns in the summary results shows the ESS for the estimated propensity scores. Note that although the original comparison group had 429 cases, the propensity score estimates effectively utilize only 23 or 27.1 of the comparison cases, depending on the rules and measures used to estimate the propensity scores. While this may seem like a large loss of sample size, this indicates that many of the original cases were unlike the treatment cases and, hence, were not useful for isolating the treatment effect. Moreover, similar or even greater reductions in ESS would be expected from alternative approaches to using propensity scores, such as matching or stratification. Since the estimand of interest in this example is ATT, `ess.treat = n.treat` throughout (i.e., all treatment cases have a weight of 1).

### 2.2 Graphical assessments of balance

The `plot()` method can generate useful diagnostic plots from the propensity score objects. The full set of plots available in `twang` and the argument value of `plot` to produce each one are given in Table 2. The convergence plot — the default — was discussed above.

The `plot()` function takes a `plots` argument in order to produce other diagnostic plots. For example, specifying `plots = 2` or `plots = "boxplot"` produces boxplots illustrating the spread...
of the estimated propensity scores in the treatment and comparison groups. Whereas propensity score stratification requires considerable overlap in these spreads, excellent covariate balance can often be achieved with weights, even when the propensity scores estimated for the treatment and control groups show little overlap.

```r
> plot(ps.lalonde, plots=2)
```

![Propensity scores plot](image)

<table>
<thead>
<tr>
<th>Descriptive numeric description</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;optimize&quot;</td>
<td>1</td>
<td>Balance measure as a function of GBM iterations</td>
</tr>
<tr>
<td>&quot;boxplot&quot;</td>
<td>2</td>
<td>Boxplot of treatment/control propensity scores</td>
</tr>
<tr>
<td>&quot;es&quot;</td>
<td>3</td>
<td>Standardized effect size of pretreatment variables</td>
</tr>
<tr>
<td>&quot;t&quot;</td>
<td>4</td>
<td>t-test p-values for weighted pretreatment variables</td>
</tr>
<tr>
<td>&quot;ks&quot;</td>
<td>5</td>
<td>Kolmogorov-Smirnov p-values for weighted pretreatment variables</td>
</tr>
<tr>
<td>&quot;histogram&quot;</td>
<td>6</td>
<td>Histogram of weights for treatment/control</td>
</tr>
</tbody>
</table>

Table 2: Available options for `plots` argument to `plot()` function.

The effect size plot illustrates the effect of weights on the magnitude of differences between groups on each pretreatment covariate. These magnitudes are standardized using the standardized effect size described earlier. In these plots, substantial reductions in effect sizes are observed for most variables (blue lines), with only one variable showing an increase in effect size (red lines), but only a seemingly trivial increase. Closed red circles indicate a statistically significant difference, many of which occur before weighting, none after. In some analyses variables can have very little variance in the treatment group sample or the entire sample and group differences can be very large relative to the standard deviations. In these situations, the user is warned that some effect sizes are too large to plot.

```r
> plot(ps.lalonde, plots=3)
```
P-values from independent tests in which the null hypothesis is true have a uniform distribution. Therefore, a QQ plot comparing the quantiles of the observed $p$-values to the quantiles of the uniform distribution illustrate whether group differences observed before and after weighting are consistent with what we would expect to see had groups been formed by random assignment (and hence the null hypothesis would be true). Setting `plots = 4` or `plots="t"` generates such QQ plots.

```r
> plot(ps.lalonde, plots = 4)
```

Before weighting (closed circles), the groups have statistically significant differences on many variables (i.e., $p$-values are near zero). After weighting (open circles) the $p$-values are generally above the 45-degree line, which represents the cumulative distribution of a uniform variable on
This indicates that the p-values are even larger than would be expected in a randomized study.

One can inspect similar plots for the KS statistic with the argument `plots = "ks"` or

```
> plot(ps.lalonde, plots = 5)
```

In all cases, the `subset` argument can be used if we wish to focus on results from one stopping rule.

```
> plot(ps.lalonde, plots = 3, subset = 2)
```
2.3 Analysis of outcomes

A separate R package, the `survey` package, is useful for performing the outcomes analyses using weights. Its statistical methods account for the weights when computing standard error estimates. It is not a part of the standard R installation but installing `twang` should automatically install `survey` as well.

> library(survey)

The `get.weights()` function extracts the propensity score weights from a `ps` object. Those weights may then be used as case weights in a `svydesign` object. By default, it returns weights corresponding to the estimand (ATE or ATT) that was specified in the original call to `ps()`. If needed, the user can override the default via the optional `estimand` argument.

> lalonde$w <- get.weights(ps.lalonde, stop.method="es.mean")
> design.ps <- svydesign(ids=~1, weights=~w, data=lalonde)

The `stop.method` argument specifies which GBM model, and consequently which weights, to utilize.

The `svydesign` function from the `survey` package creates an object that stores the dataset along with design information needed for analyses. See `help(svydesign)` for more details on setting up `svydesign` objects.

The aim of the National Supported Work Demonstration analysis is to determine whether the program was effective at increasing earnings in 1978. The propensity score adjusted test can be computed with `svyglm`.

> glm1 <- svyglm(re78 ~ treat, design=design.ps)
> summary(glm1)

Call:
svyglm(formula = re78 ~ treat, design = design.ps)

Survey design:
svydesign(ids = ~1, weights = ~w, data = lalonde)

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)  5616.600    884.931  6.347  4.28e-10 ***
treat         732.547    1056.616  0.693    0.488
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 49804197)

Number of Fisher Scoring iterations: 2

The analysis estimates an increase in earnings of $733 for those that participated in the NSW compared with similarly situated people observed in the CPS. The effect, however, does not appear to be statistically significant.

Some authors have recommended utilizing both propensity score adjustment and additional covariate adjustment to minimize mean square error or to obtain “doubly robust” estimates of the treatment effect (Huppler-Hullsiek & Louis 2002, Bang & Robins 2005). These estimators
are consistent if either the propensity scores are estimated correctly or the regression model is specified correctly. For example, note that the balance table for \texttt{ks.max.ATT} made the two groups more similar on \texttt{nodegree}, but still some differences remained, 70.8\% of the treatment group had no degree while 60.1\% of the comparison group had no degree. While linear regression is sensitive to model misspecification when the treatment and comparison groups are dissimilar, the propensity score weighting has made them more similar, perhaps enough so that additional modeling with covariates can adjust for any remaining differences. In addition to potential bias reduction, the inclusion of additional covariates can reduce the standard error of the treatment effect if some of the covariates are strongly related to the outcome.

```r
> glm2 <- svyglm(re78 ~ treat + nodegree, design=design.ps)
> summary(glm2)
```

Call:  
\texttt{svyglm(formula = re78 \sim treat + nodegree, design = design.ps)}

Survey design:  
\texttt{svydesign(ids = \sim 1, weights = \sim w, data = lalonde)}

Coefficients:  
\begin{tabular}{lccccc}
 & Estimate & Std. Error & t value & Pr(>|t|) \\
(Intercept) & 6768.4 & 1471.0 & 4.601 & 5.11e-06 *** \\
treat & 920.3 & 1082.8 & 0.850 & 0.396 \\
nodegree & -1891.8 & 1261.9 & -1.499 & 0.134 \\
\end{tabular}

---

Signif. codes:  \texttt{0 \textquotesingle	extquotesingle	extquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle 0.001 \textquotesingle	extquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle 0.01 \textquotesingle	extquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle 0.05 \textquotesingle	extquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle 0.1 \textquotesingle	extquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle \textquotesingle 1}

(Dispersion parameter for \texttt{gaussian} family taken to be 49013778)

Number of Fisher Scoring iterations: 2

Adjusting for the remaining group difference in the \texttt{nodegree} variable slightly increased the estimate of the program's effect to $920, but the difference is still not statistically significant. We can further adjust for the other covariates, but that too in this case has little effect on the estimated program effect.

```r
> glm3 <- svyglm(re78 ~ treat + age + educ + black + hispan + nodegree +
+ married + re74 + re75, +
+ design=design.ps)
> summary(glm3)
```

Call:  
\texttt{svyglm(formula = re78 \sim treat + age + educ + black + hispan +
nodegree + married + re74 + re75, design = design.ps)}

Survey design:  
\texttt{svydesign(ids = \sim 1, weights = \sim w, data = lalonde)}

Coefficients:  
\begin{tabular}{lccccc}
 & Estimate & Std. Error & t value & Pr(>|t|) \\
(Intercept) & -2.459e+03 & 4.289e+03 & -0.573 & 0.56671 \\
\end{tabular}
2.4 Estimating the program effect using linear regression

The more traditional regression approach to estimating the program effect would fit a linear model with a treatment indicator and linear terms for each of the covariates.

```r
> glm4 <- lm(re78 ~ treat + age + educ + black + hispan + nodegree +
+ married + re74 + re75,
+ data=lalonde)
> summary(glm4)
```

```
Call:
  lm(formula = re78 ~ treat + age + educ + black + hispan + nodegree +
     married + re74 + re75, data = lalonde)

Residuals:
     Min      1Q  Median      3Q     Max
-13595.4  -4894.0  -1662.2   3929.3  54570.2

Coefficients:  Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.651e+01 2.437e+03  0.027   0.978
  treat      1.548e+03 7.813e+02  1.982   0.048 *
   age       1.298e+01 3.249e+01  0.399   0.680
  educ       4.039e+02 1.589e+02  2.542   0.011 *
  black     -1.241e+03 7.688e+02 -1.614   0.107
 hispan     4.989e+02 9.419e+02  0.530   0.597
nodegree   2.598e+02 8.474e+02  0.307   0.759
 married  4.066e+02 6.955e+02  0.585   0.559
    re74    2.964e-01 5.827e-02  5.086  4.89e-07 ***
    re75   2.315e-01 1.046e-01  2.213   0.027 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6948 on 604 degrees of freedom

(Dispersion parameter for gaussian family taken to be 47150852)
Number of Fisher Scoring iterations: 2
Multiple R-squared: 0.1478,  Adjusted R-squared: 0.1351
F-statistic: 11.64 on 9 and 604 DF,  p-value: < 2.2e-16

This model estimates a rather strong treatment effect, estimating a program effect of $1548 with a p-value=0.048. Several variations of this regression approach also estimate strong program effects. For example using square root transforms on the earnings variables yields a p-value=0.016. These estimates, however, are very sensitive to the model structure since the treatment and control subjects differ greatly as seen in the unweighted balance comparison ($unw) from `bal.table(ps.lalonde)`.

2.5 Propensity scores estimated from logistic regression

Propensity score analysis is intended to avoid problems associated with the misspecification of covariate adjusted models of outcomes, but the quality of the balance and the treatment effect estimates can be sensitive to the method used to estimate the propensity scores. Consider estimating the propensity scores using logistic regression instead of `ps()`.

```r
> ps.logit <- glm(treat ~ age + educ + black + hispan + nodegree +
+ married + re74 + re75,
+ data = lalonde,
+ family = binomial)
> lalonde$w.logit <- rep(1,nrow(lalonde))
> lalonde$w.logit[lalonde$treat==0] <- exp(predict(ps.logit,subset(lalonde,treat==0)))
```

`predict()` for logistic regression model produces estimates on the log-odds scale by default. Exponentiating those predictions for the comparison subjects gives the ATT weights $p/(1-p)$. `dx.wts()` from the `twang` package diagnoses the balance for an arbitrary set of weights producing a balance table. This function requires the user to specify the estimand argument in order to perform the appropriate calculations relative to the target group on which we are drawing inferences.

```r
> bal.logit <- dx.wts(x = lalonde$w.logit,
+ data=lalonde,
+ vars=c("age","educ","black","hispan","nodegree",
+ "married","re74","re75"),
+ treat.var="treat",
+ perm.test.iters=0, estimand = "ATT")
```

```r
> bal.logit

   type n.treat n.ctrl ess.treat ess.ctrl max.es
1 unw 185 429 185 429.00000 1.7567745
2 185 429 185 99.81539 0.1188496

   mean.es max.ks mean.ks iter
1 0.5687259 0.6404460 0.27024507 NA
2 0.0318841 0.3078039 0.09302319 NA
```

Applying the `bal.table()` function to this object returns a variable-by-variable summary of balance, just like it did for the `ps` object.
For weights estimated with logistic regression, the largest KS statistic was reduced from the unweighted sample's largest KS of 0.64 to 0.31, which is still quite a large KS statistic. Table 3 shows the details of the balance of the treatment and comparison groups. The means of the two groups appear to be quite similar while the KS statistic shows substantial differences in their distributions.
> names(pretty.tab) <- c("E(Y1|t=1)","E(Y0|t=1)","KS","E(Y0|t=0)")
> xtable(pretty.tab,
+ caption = "Logistic regression estimates of the propensity scores",
+ label = "tab:balancelogit",
+ digits = c(0, 2, 2, 2, 2),
+ align=c("l","r","r","r","r"))

|       | E(Y1|t=1) | E(Y0|t=1) | KS   | E(Y0|t=0) |
|-------|----------|----------|------|-----------|
| age   | 25.82    | 24.97    | 0.31 | 28.03     |
| educ  | 10.35    | 10.40    | 0.04 | 10.23     |
| black | 0.84     | 0.84     | 0.00 | 0.20      |
| hispan| 0.06     | 0.06     | 0.00 | 0.14      |
| nodegree | 0.71   | 0.69     | 0.02 | 0.60      |
| married | 0.19   | 0.17     | 0.02 | 0.51      |
| re74  | 2095.57  | 2106.05  | 0.23 | 5619.24   |
| re75  | 1532.06  | 1496.54  | 0.13 | 2466.48   |

Table 3: Logistic regression estimates of the propensity scores

Table 4 compares the balancing quality of the weights directly with one another.

<table>
<thead>
<tr>
<th></th>
<th>n.treat</th>
<th>ess.ctrl</th>
<th>max.es</th>
<th>mean.es</th>
<th>max.ks</th>
<th>mean.ks</th>
</tr>
</thead>
<tbody>
<tr>
<td>unw</td>
<td>185</td>
<td>429.00</td>
<td>1.76</td>
<td>0.57</td>
<td>0.64</td>
<td>0.27</td>
</tr>
<tr>
<td>logit</td>
<td>185</td>
<td>99.82</td>
<td>0.12</td>
<td>0.03</td>
<td>0.31</td>
<td>0.09</td>
</tr>
<tr>
<td>es.mean.ATT</td>
<td>185</td>
<td>22.96</td>
<td>0.22</td>
<td>0.08</td>
<td>0.12</td>
<td>0.06</td>
</tr>
<tr>
<td>ks.max.ATT</td>
<td>185</td>
<td>27.05</td>
<td>0.23</td>
<td>0.08</td>
<td>0.11</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Table 4: Summary of the balancing properties of logistic regression and gbm

> design.logit <- svydesign(ids=~1, weights=~w.logit, data=lalonde)
> glm6 <- svyglm(re78 ~ treat, design=design.logit)
> summary(glm6)

Call:
svyglm(formula = re78 ~ treat, design = design.logit)

Survey design:
svydesign(ids = ~1, weights = ~w.logit, data = lalonde)

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
(Intercept) 5135.1 588.9 8.719 <2e-16 ***
treat 1214.1 824.7 1.472 0.142
---
Signif. codes:  0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 49598072)

Number of Fisher Scoring iterations: 2
The analysis estimates an increase in earnings of $1214 for those that participated in the NSW compared with similarly situated people observed in the CPS. Table 5 compares all of the treatment effect estimates.

<table>
<thead>
<tr>
<th>Treatment effect</th>
<th>PS estimate</th>
<th>Linear adjustment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$733</td>
<td>GBM, minimize KS</td>
<td>none</td>
</tr>
<tr>
<td>$920</td>
<td>GBM, minimize KS</td>
<td>nodegree</td>
</tr>
<tr>
<td>$758</td>
<td>GBM, minimize KS</td>
<td>all</td>
</tr>
<tr>
<td>$1548</td>
<td>None</td>
<td>all</td>
</tr>
<tr>
<td>$1214</td>
<td>Logistic regression</td>
<td>none</td>
</tr>
<tr>
<td>$1237</td>
<td>Logistic regression</td>
<td>all</td>
</tr>
</tbody>
</table>

Table 5: Treatment effect estimates by various methods

3 An ATE example

In the analysis of Section 2, we focused on estimating ATT for the lalonde dataset. In this situation, the ATE is not of great substantive interest because not all people who are offered entrance into the program could be expected to take advantage of the opportunity. Further, there is some evidence that the treated subjects were drawn from a subset of the covariate space. In particular, in an ATE analysis, we see that we are unable to achieve balance, especially for the “black” indicator.

We now turn to an ATE analysis that is feasible and meaningful. We focus on the lindner dataset, which was included in the USPS package (Obenchain 2011), and is now included in twang for convenience. A tutorial by Helmreich and Pruzek (2009; HP) for the PSAgraphics package also uses propensity scores to analyze a portion of these data. HP describe the data as follows on p. 3 with our minor recodings in square braces:

The lindner data contain data on 996 patients treated at the Lindner Center, Christ Hospital, Cincinnati in 1997. Patients received a Percutaneous Coronary Intervention (PCI). The data consists of 10 variables. Two are outcomes: [sixMonthSurvive] ranges over two values... depending on whether patients survived to six months post treatment [denoted by TRUE] or did not survive to six months [FALSE]... Secondly, cardbill contains the costs in 1998 dollars for the first six months (or less if the patient did not survive) after treatment... The treatment variable is abcix, where 0 indicates PCI treatment and 1 indicates standard PCI treatment and additional treatment in some form with abciximab. Covariates include acutemi, 1 indicating a recent acute myocardial infarction and 0 not; ejecfrac for the left ventricle ejection fraction, a percentage from 0 to 90; ves1proc giving the number of vessels (0 to 5) involved in the initial PCI; stent with 1 indicating coronary stent inserted, 0 not; diabetic where 1 indicates that the patient has been diagnosed with diabetes, 0 not; height in centimeters and female coding the sex of the patient, 1 for female, 0 for male.

HP focus on cardbill — the cost for the first months after treatment — as their outcome of interest. However, since not all patients survived to six months, it is not clear whether a lower value of cardbill is good or not. For this reason, we choose six-month survival (sixMonthSurvive) as our outcome of interest.

Ignoring pre-treatment variables, we see that abcix is associated with lower rates of 6-month mortality:
> data(lindner)
> table(lindner$sixMonthSurvive, lindner$abcix)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td>TRUE</td>
<td>283</td>
<td>687</td>
</tr>
</tbody>
</table>

> chisq.test(table(lindner$sixMonthSurvive, lindner$abcix))

```
Pearson's Chi-squared test with Yates' continuity correction
data: table(lindner$sixMonthSurvive, lindner$abcix)
X-squared = 8.5077, df = 1, p-value = 0.003536
```

The question is whether this association is causal. If health care policies were to be made on the basis of these data, we would wish to elicit expert opinion as to whether there are likely to be other confounding pretreatment variables. For this tutorial, we simply follow HP in choosing the pre-treatment covariates. The `twang` model is fit as follows

> set.seed(1)
> ps.lindner <- ps(abcix ~ stent + height + female + diabetic +
+                        acutemi + ejecfrac + ves1proc, data = lindner,
+                        verbose = FALSE, estimand = "ATE")

We set `estimand = "ATE"` because we are interested in the effects of abciximab on everyone in the population. We do not specify the stopping rules. Consequently `ps()` uses the defaults: `es.mean` and `ks.max`. We then inspect pre- and post-weighting balance with the command

> bal.table(ps.lindner)

```
$unw
  tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p  ks ks.pval
stent 0.705 0.456 0.584 0.494 0.257 3.624 0.000 0.121 0.004
height 171.443 10.695 171.446 10.589 0.000 -0.005 0.996 0.025 0.999
female 0.331 0.471 0.386 0.488 -0.115 -1.647 0.100 0.055 0.531
diabetic 0.205 0.404 0.268 0.444 -0.152 -2.127 0.034 0.064 0.349
acutemi 0.331 0.471 0.386 0.488 -0.115 -1.647 0.100 0.055 0.531
ejecfrac 50.403 10.419 52.289 10.297 -0.181 -2.640 0.008 0.114 0.008
ves1proc 1.463 0.706 1.205 0.480 0.393 6.693 0.000 0.188 0.000

$ks.mean.ATE
  tx.mn tx.sd ct.mn ct.sd std.eff.sz stat p  ks ks.pval
stent 0.683 0.466 0.657 0.475 0.054 0.718 0.473 0.025 1.000
height 171.470 10.550 171.591 10.589 -0.011 -0.155 0.877 0.015 1.000
female 0.338 0.473 0.345 0.476 -0.015 -0.200 0.841 0.007 1.000
diabetic 0.215 0.411 0.229 0.421 -0.033 -0.432 0.666 0.014 1.000
acutemi 0.148 0.355 0.107 0.310 0.120 1.331 0.183 0.040 0.935
jejefrac 51.052 10.333 51.604 9.110 -0.181 -2.640 0.008 0.114 0.008
ves1proc 1.395 0.666 1.337 0.573 0.094 1.203 0.229 0.028 0.999

$es.mean.ATE
```
This balance table shows that *stent*, *acutemi*, *ejectfrac* and *ves1proc* were all significantly imbalanced before weighting. After weighting (using either *stop.method* considered) we do not see problems in this regard. Examining `plot(ps.lindner, plots = x)` for `x` running from 1 to 5 does not reveal problems, either. In regard to the optimize plot, we note that the scales of the KS and ES statistics presented in the optimize plots are not necessarily comparable. The fact that the KS values are lower than the ES values in the optimize plot does not suggest that the KS stopping rule is finding superior models. Each panel of the optimize plot indicates the gbm model that minimizes each stopping rule. The panels should not be compared other than to compare the number of iterations selected by each rule.

```r
> plot(ps.lindner, plots = 1)
```
> plot(ps.lindner, plots = 2)

![Plot showing propensity scores and treatment]  

> plot(ps.lindner, plots = 3)

![Plot showing absolute standard differences]
From a call to `summary()`, we see that the `es.mean.ATE` stopping rule results in a slightly higher ESS with comparable balance measures, so we proceed with those weights. Also, we note that `ess.treat` is no longer equal to `n.treat` since we are focusing on ATE rather than ATT.

```r
> summary(ps.lindner)

n.treat n.ctrl ess.treat ess.ctrl max.es
unw 698 298 698.0000 298.0000 0.3925637
```
ks.mean.ATE  698   298   655.6754  228.8501  0.1197287
es.mean.ATE  698   298   658.4838  230.7292  0.1185830

mean.es  max.ks  max.ks.p  mean.ks  iter
unw   0.20528943 0.18841945 NA  0.09791845  NA
ks.mean.ATE  0.05495511 0.04012757 NA  0.02235951 2603
es.mean.ATE  0.05423131 0.03977072 NA  0.02273098 2096

As before, we use the survey package to reweight our sample and perform the analysis.

> lindner$w <- get.weights(ps.lindner, stop.method = "es.mean")
> design.ps <- svydesign(ids=~1, weights = ~w, data = lindner)
> svychisq(~sixMonthSurvive + abcix, design = design.ps)

Pearson's X^2: Rao & Scott adjustment

data: svychisq(~sixMonthSurvive + abcix, design = design.ps)
F = 9.3894, ndf = 1, ddf = 995, p-value = 0.002241

The reweighting does not diminish the association between the treatment and the outcome. Indeed, it is marginally more significant after the reweighting.

4 Non-response weights

The twang package was designed to estimate propensity score weights for the evaluation of treatment effects in observational or quasi-experimental studies. However, we find that the package includes functions and diagnostic tools that are highly valuable for other applications, such as for generating and diagnosing nonresponse weights for survey nonresponse or study attrition. We now present an example that uses the tools in twang. This example uses the subset of the US Sustaining Effects Study data distributed with the HLM software (Bryk, Raudenbush, Congdon, 1996) and also available in the R package mlmRev. The data include mathematics test scores for 1721 students in kindergarten to fourth grade. They also include student race (black, Hispanic, or other), gender, an indicator for whether or not the student had been retained in grade, the percent low income students at the school, the school size, the percent of mobile students, the students’ grade-levels, student and school IDs, and grades converted to year by centering. The study analysis plans to analyze growth in math achievement from grade 1 to grade 4 using only students with complete data. However, the students with complete data differ from other students. To reduce bias that could potentially result from excluding incomplete cases, our analysis plan is to weight complete cases with nonresponse weights.

The goal of nonresponse weighting is to develop weights for the respondents that make them look like the entire sample — both the respondents and nonrespondents. Since the respondents already look like themselves, the hard part is to figure out how well each respondent represents the nonrespondents. Nonresponse weights equal the reciprocal of the probability of response and are applied only to respondents.

Note that the weights p are equivalent to the propensity score if we consider subjects with an observed outcome to be the “treated” group, and those with an unobserved outcome to be the “controls”. We wish to reweight the sample to make it equivalent to the population from which the sample was drawn, so ATE weights are more appropriate in this case. Further, recall that the weights for the treated subjects are 1/p in an ATE analysis. Therefore we can reweight the sample of respondents using the get.weights() function.

Before we can generate nonresponse weights, we need to prepare the data using the following commands. First we load the data.
Next we create the patterns of grades for which students have responses

```r
> tmp <- sapply(split(egsingle, egsingle$childid), function(x){
+   paste(as.character(x$grade), collapse="")})
```

identify students with test scores for every grade from 1 to 4

```r
> tmp <- data.frame(childid=names(tmp), gpatt=tmp,
+                   resp=as.numeric((1:length(tmp)) %in%
+                   grep("1234", as.character(tmp))))
```

and merge this back to create a single data frame

```r
> egsingle <- merge(egsingle, tmp)
```

Because nonresponse is a student-level variable rather than a student-by-year-level variable we create one record per student.

```r
> egsingle.one <- unique(egsingle[, -c(3:6)])
```

We also create a race variable

```r
> egsingle.one$race <- as.factor(race <- ifelse(egsingle.one$black==1, 1,
+                                    ifelse(egsingle.one$hispanic==1, 2, 3)))
```

As discussed above, to use `ps()` to estimate nonresponse, we need to let respondents be the treatment group by modeling an indicator of response.

```r
> egsingle.ps <-
+   ps(resp ~ race + female + size + lowinc + mobility,
+     data=egsingle.one,
+     stop.method=c("es.mean", "ks.max"),
+     n.trees=2500,
+     verbose=FALSE,
+     estimand = "ATE")
```

The final steps are to find the ATE weights

```r
> egsingle.one$wgt <- get.weights(egsingle.ps, stop.method="ks.max")
```

and select only the records with an observed outcome.

```r
> egsinge.resp <- merge(subset(egsingle, subset=resp==1),
+                       subset(egsingle.one, subset=resp==1,
+                      select=c(childid, wgt)))
```
Figure 2: Optimization of es.mean.ATE and ks.max.ATE for nonresponse weighting of eg single data. The horizontal axes indicate the number of iterations and the vertical axes indicate the measure of imbalance between the two groups. For es.mean.ATE the measure is the average effect size difference between the two groups and for ks.max.ATE the measure is the largest of the KS statistics.
Non-responders  Weighted responders  Std ES  KS
race:1  0.69  0.70  -0.03  0.01
race:2  0.14  0.14  -0.01  0.00
race:3  0.17  0.15  0.04  0.02
female:Female  0.49  0.49  -0.01  0.00
female:Male  0.51  0.51  0.01  0.00
size  756.72  758.89  -0.01  0.02
lowinc  78.52  78.48  0.00  0.03
mobility  34.23  34.79  -0.04  0.02

Table 6: Balance of the nonrespondents and respondents

5 The details of twang

5.1 Propensity scores and weighting

Propensity scores can be used to reweight comparison cases so that the distribution of their features match the distribution of features of the treatment cases, for ATT, or cases from both treatment and control groups to match each other, for ATE (Rosenbaum 1987, Wooldridge 2002, Hirano and Imbens 2001, McCaffrey et al. 2004) Let \( f(x|t=1) \) be the distribution of features for the treatment cases and \( f(x|t=0) \) be the distribution of features for the comparison cases. If treatments were randomized then we would expect these two distributions to be similar. When they differ for ATT we will construct a weight, \( w(x) \), so that

\[
\frac{f(x|t=1)}{f(x|t=0)} = w(x) = \frac{\frac{f(t=1|x)}{f(t=0|x)}}{1 - P(t=1|x)}, \tag{3}
\]

where \( K \) is a normalization constant that will cancel out in the outcomes analysis. Equation (3) indicates that if we assign a weight to comparison case \( i \) equal to the odds that a case with features \( x_i \) would be exposed to the treatment, then the distribution of their features would balance. Note that for comparison cases with features that are atypical of treatment cases, the propensity score \( P(t=1|x) \) would be near 0 and would produce a weight near 0. On the other hand, comparison cases with features typical of the treatment cases would receive larger weights.

For ATE, each group is weighted to match the population. The weight must satisfy:

\[
\begin{align*}
f(x|t=1) &= w(x)f(x), \quad \text{and} \tag{4} \\
f(x|t=0) &= w(x)f(x), \quad \text{and} \tag{5}
\end{align*}
\]

Again using Bayes Theorem we obtain \( w(x) = 1/f(x|t=1) \) for the treatment group and \( w(x) = 1/f(x|t=0) \) for the control group.
5.2 Estimating the propensity score

In randomized studies $P(t = 1|x)$ is known and fixed in the study design. In observational studies the propensity score is unknown and must be estimated, but poor estimation of the propensity scores can cause just as much of a problem for estimating treatment effects as poor regression modeling of the outcome. Linear logistic regression is the common method for estimating propensity scores, and can suffice for many problems. Linear logistic regression for propensity scores estimates the log-odds of a case being in the treatment given $x$ as

$$\log \frac{P(t = 1|x)}{1 - P(t = 1|x)} = \beta^T x \quad (6)$$

Usually, $\beta$ is selected to maximize the logistic log-likelihood

$$\ell(\beta) = \frac{1}{n} \sum_{i=1}^{n} t_i \beta^T x_i - \log (1 + \exp(\beta^T x_i)) \quad (7)$$

Maximizing (7) provides the maximum likelihood estimates of $\beta$. However, in an attempt to remove as much confounding as possible, observational studies often record data on a large number of potential confounders, many of which can be correlated with one another. Standard methods for fitting logistic regression models to such data with the iteratively reweighted least squares algorithm can be statistically and numerically unstable. To improve the propensity score estimates we might also wish to include non-linear effects and interactions in $x$. The inclusion of such terms only increases the instability of the models.

One increasingly popular method for fitting models with numerous correlated variables is the lasso (least absolute subset selection and shrinkage operator) introduced in statistics in Tibshirani (1996). For logistic regression, lasso estimation replaces (7) with a version that penalizes the absolute magnitude of the coefficients

$$\ell(\beta) = \frac{1}{n} \sum_{i=1}^{n} t_i \beta^T x_i - \log (1 + \exp(\beta^T x_i)) - \lambda \sum_{j=1}^{J} |\beta_j| \quad (8)$$

The second term on the right-hand side of the equation is the penalty term since it decreases the overall of $\ell(\beta)$ when there are coefficient that are large in absolute value. Setting $\lambda = 0$ returns the standard (and potentially unstable) logistic regression estimates of $\beta$. Setting $\lambda$ to be very large essentially forces all of the $\beta_j$ to be equal to 0 (the penalty excludes $\beta_0$). For a fixed value of $\lambda$ the estimated $\hat{\beta}$ can have many coefficients exactly equal to 0, not just extremely small but precisely 0, and only the most powerful predictors of $t$ will be non-zero. As a result the absolute penalty operates as a variable selection penalty. In practice, if we have several predictors of $t$ that are highly correlated with each other, the lasso tends to include all of them in the model, shrink their coefficients toward 0, and produce a predictive model that utilizes all of the information in the covariates, producing a model with greater out-of-sample predictive performance than models fit using variable subset selection methods.

Our aim is to include as covariates all piecewise constant functions of the potential confounders and their interactions. That is, in $x$ we will include indicator functions for continuous variables like $I(\text{age} < 15), I(\text{age} < 16), \ldots, I(\text{age} < 90)$, etc., for categorical variables like $I(\text{sex} = \text{male}), I(\text{prior MI} = \text{TRUE})$, and interactions among them like $I(\text{age} < 16)I(\text{sex} = \text{male})I(\text{prior MI} = \text{TRUE})$. This collection of basis functions spans a plausible set of propensity score functions, are computationally efficient, and are flat at the extremes of $x$ reducing the likelihood of propensity score estimates near 0 and 1 that can occur with linear basis functions of $x$. Theoretically with the lasso we can estimate the model in (8), selecting a $\lambda$ small enough so that it will eliminate most of the irrelevant terms and yield a sparse model with only the
most important main effects and interactions. Boosting (Friedman 2001, 2003, Ridgeway 1999) effectively implements this strategy using a computationally efficient method that Efron et al. (2004) showed is equivalent to optimizing (8). With boosting it is possible to maximize (8) for a range of values of $\lambda$ with no additional computational effort than for a specific value of $\lambda$. We use boosted logistic regression as implemented in the generalized boosted modeling (gbm) package in R (Ridgeway 2005).

5.3 Evaluating the weights

As with regression analyses, propensity score methods cannot adjust for unmeasured covariates that are uncorrelated with the observed covariates. Nonetheless, the quality of the adjustment for the observed covariates achieved by propensity score weighting is easy to evaluate. The estimated propensity score weights should equalize the distributions of the cases’ features as in (2). This implies that weighted statistics of the covariates of the comparison group should equal the same statistics for the treatment group. For example, the weighted average of the age of comparison cases should equal the average age of the treatment cases. To assess the quality of the propensity score weights one could compare a variety of statistics such as means, medians, variances, and Kolmogorov-Smirnov statistics for each covariate as well as interactions. The twang package provides both the standardized effect sizes and KS statistics and p-values testing for differences in the means and distributions of the covariates for analysts to use in assessing balance.

5.4 Analysis of outcomes

With propensity score analyses the final outcomes analysis is generally straightforward, while the propensity score estimation may require complex modeling. Once we have weights that equalize the distribution of features of treatment and control cases by reweighting. For ATT, we give each treatment case a weight of 1 and each comparison case a weight $w_i = p(x_i)/(1 - p(x_i))$. To estimate the ATE, we give control cases weight $w_i = 1/p(x_i)$ and we give the treatment cases $w_i = 1/(1 - p(x_i))$. We then estimate the treatment effect estimate with a weighted regression model that contains only a treatment indicator. No additional covariates are needed if the weights account for differences in $x$.

A combination of propensity score weighting and covariate adjustment can be useful for several reasons. First, the propensity scores may not have been able to completely balance all of the covariates. The inclusion of these covariates in addition to the treatment indicator in a weighted regression model may correct this if the imbalance is relatively small. Second, in addition to exposure, the relationship between some of the covariates and the outcome may also be of interest. Their inclusion can provide coefficients that can estimate the direction and magnitude of the relationship. Third, as with randomized trials, stratifying on covariates that are highly correlated with the outcome can improve the precision of estimates. Lastly, the some treatment effect estimators that utilize an outcomes regression model and propensity scores are “doubly robust” in the sense that if either the propensity score model is correct or the regression model is correct then the treatment effect estimator will be unbiased (Bang & Robins 2005).

References


