Propensity scores for multiple treatments: A tutorial for the `mnps` function in the `twang` package

Lane Burgette, Beth Ann Griffin and Dan McCaffrey *
RAND Corporation

March 18, 2014

1 Introduction

The Toolkit for Weighting and Analysis of Nonequivalent Groups, `twang`, was designed to make causal estimates in the binary treatment setting. In `twang` versions 1.3 and later, we have extended this software package to handle more than two treatment conditions through the `mnps` function, which stands for multinomial propensity scores. McCaffrey et al. (2013) describe the methodology behind the `mnps` function; the purpose of this document is to describe the syntax and features related to the implementation in `twang`.

At a high level, the `mnps` function decomposes the propensity score estimation into several applications of the `ps` function, which was designed for the standard dichotomous treatment setting. For this reason, users who are new to `twang` are encouraged to learn about the `ps` function before using the `mnps` function. The other vignette that accompanies the package (Ridgeway et al., 2012) provides an extensive overview of the `ps` function, and much of that information will not be repeated here.

2 An ATE example

To demonstrate the package we utilize a random subset of the data described in McCaffrey et al. (2013). This truncated dataset is called `AOD`, and is included in the package. There are three treatment groups in the study, and the data include records for 200 youths in each treatment group of an alcohol and other drug treatment evaluation. We begin by loading the package and the data. Because there is a stochastic component to the subsequent model fits, we also set the random seed to ensure full replicability.

```
> library(twang)
> data(AOD)
> set.seed(1)
```

For the `AOD` dataset, the variable `treat` contains the treatment indicators, which have possible values `community`, `metcbt5`, and `scy`. The other variables included in the dataset are:

- `suf12`: outcome variable, substance use frequency at 12 month follow-up

*The development of this software and tutorial was funded by National Institute of Drug Abuse grants number 1R01DA015697 (PI: McCaffrey) and 1R01DA034065 (PI: Griffin).

- `illact`: pretreatment covariate, illicit activities scale
- `crimjust`: pretreatment covariate, criminal justice involvement
- `subprob`: pretreatment covariate, substance use problem scale
- `subdep`: pretreatment covariate, substance use dependence scale
- `white`: pretreatment covariate, indicator for non-Hispanic white youth

In such an observational study, there are several quantities that one may be interested in estimating. The estimands that are most commonly of interest are the average treatment effect on the population (ATE) and the average treatment effect on the treated (ATT). The differences between these quantities are explained at length in McCaffrey et al. (2013), but in brief the ATE answers the question of how, on average, the outcome of interest would change if everyone in the population of interest had been assigned to a particular treatment relative to if they had all received another single treatment. The ATT answers the question of how the average outcome would change if everyone who received one particular treatment had instead received another particular treatment.

The main argument for the `mnps` function is a formula with the treatment variable on the left-hand side of a tilde, and pre-treatment variables on the right-hand side, separated by plus signs. Other key arguments are `data`, which simply tells the function the name of the dataframe that contains the variables for the propensity score estimation; the `estimand`, which can either be "ATT" or "ATE"; and `verbose`, which if set as `TRUE` instructs the function to print updates on the model fitting process, which can take a few minutes.

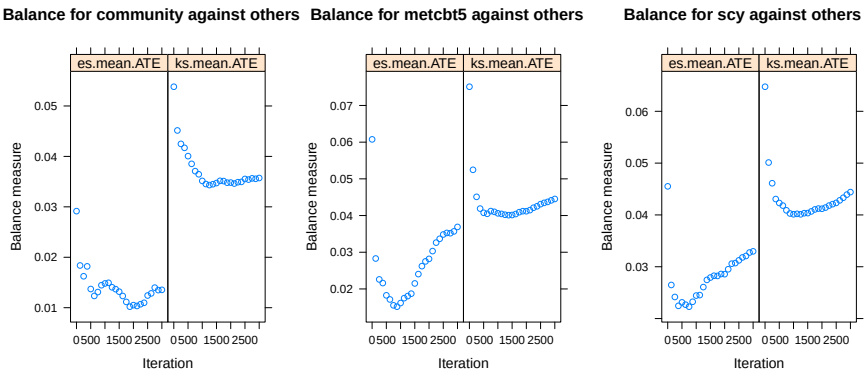
```
> mnps.AOD <- mnps(treat ~ illact + crimjust + subprob + subdep + white,
+                  data = AOD, estimand = "ATE", verbose = FALSE,
+                  stop.method = c("es.mean", "ks.mean"),
+                  n.trees = 3000)
```

The `twang` methods rely on tree-based regression models that are built in an iterative fashion. As the iterations or number of regression trees added to the model increases, the model becomes more complex. However, at some point, more complex models typically result in worse balance on the pre-treatment variables and therefore are less useful in a propensity score weighting context. The `n.trees` argument controls the maximum number of iterations.

Another key choice is the measure of balance that one uses when fitting these models. This is specified in the `stop.method` argument. As with the `ps` function, four `stop.method` objects are included in the package. 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., illicit activities scale). The default stopping rules in `twang` use two balance metrics: absolute standardized bias (also referred to as the absolute standardized mean difference or the effect size (ES)) 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. In this example, we chose to examine both `es.mean` and `ks.mean`, which is the default.

A first step is to make sure that we let the models run for a sufficiently large number of iterations in order to optimize the balance statistics of interest. We do this by seeing whether any of the balance measures of interest still appear to be decreasing after the number of iterations specified by the argument `n.trees` (10,000 iterations is the default).

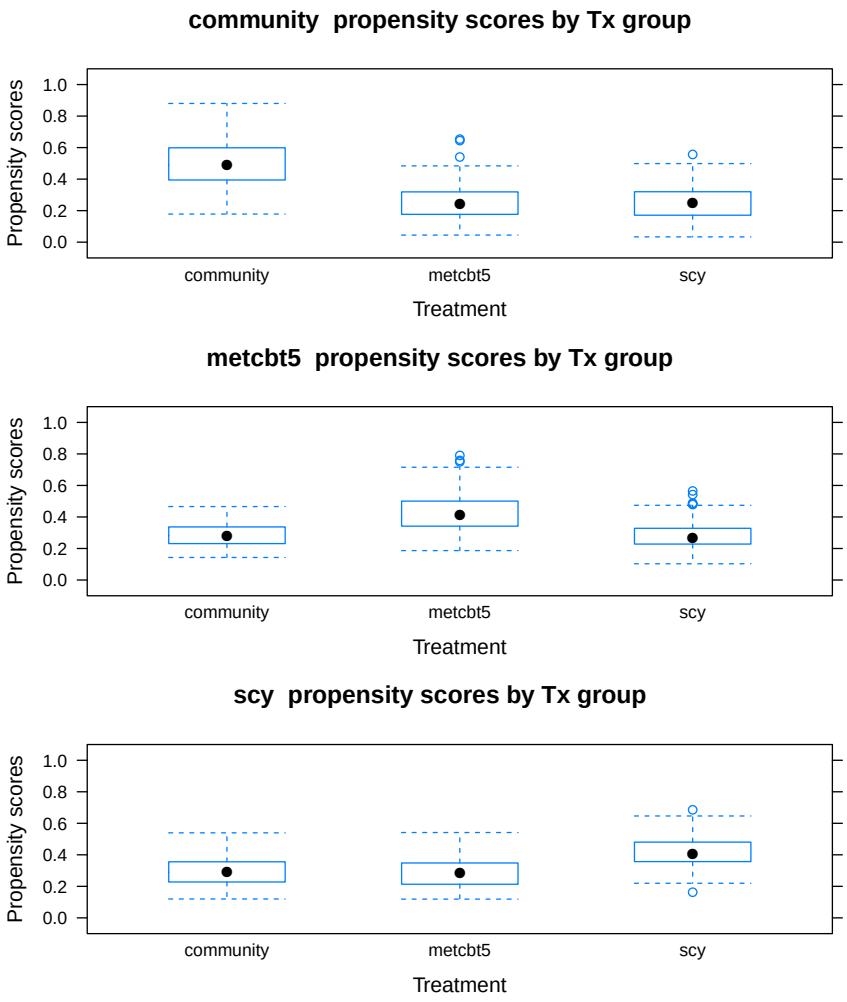
```
> plot(mnps.AOD, plots = 1)
```



In this figure, it appears that each of the balance measures are optimized with substantially fewer than 3,000 iterations, so we do not have evidence that we should re-run the `mnps()` call with a higher number of iterations or trees.

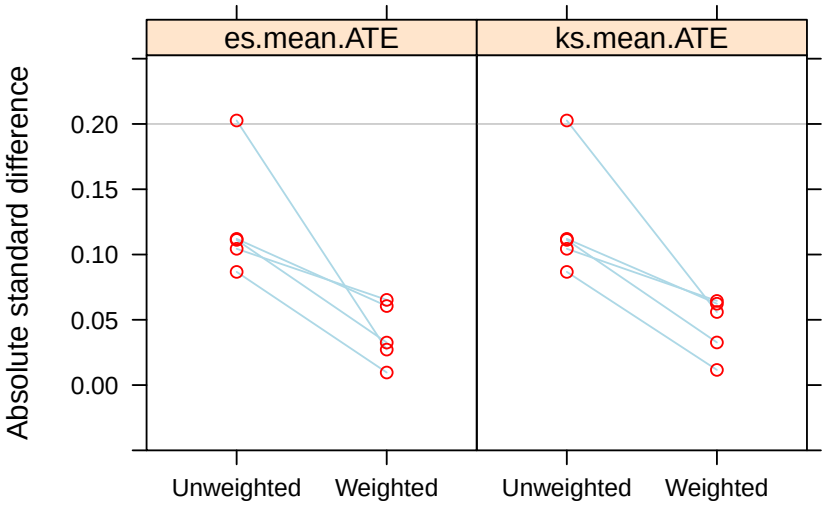
A key assumption in propensity score analyses is that each experimental unit has a non-zero probability of receiving each treatment. The plausibility of this assumption may be assessed by the overlap of the empirical propensity score distributions. This diagnostic is available using the `plots = 2` argument in the `plot` function. Here, the overlap assumption generally seems to be met, although there should be some concern that adolescents in the metcbt5 and scy conditions do not overlap well with the community group given the top most graphic. See McCaffrey et al. (2013) for more details on this issue.

```
>     plot(mnps.AOD, plots = 2)
```



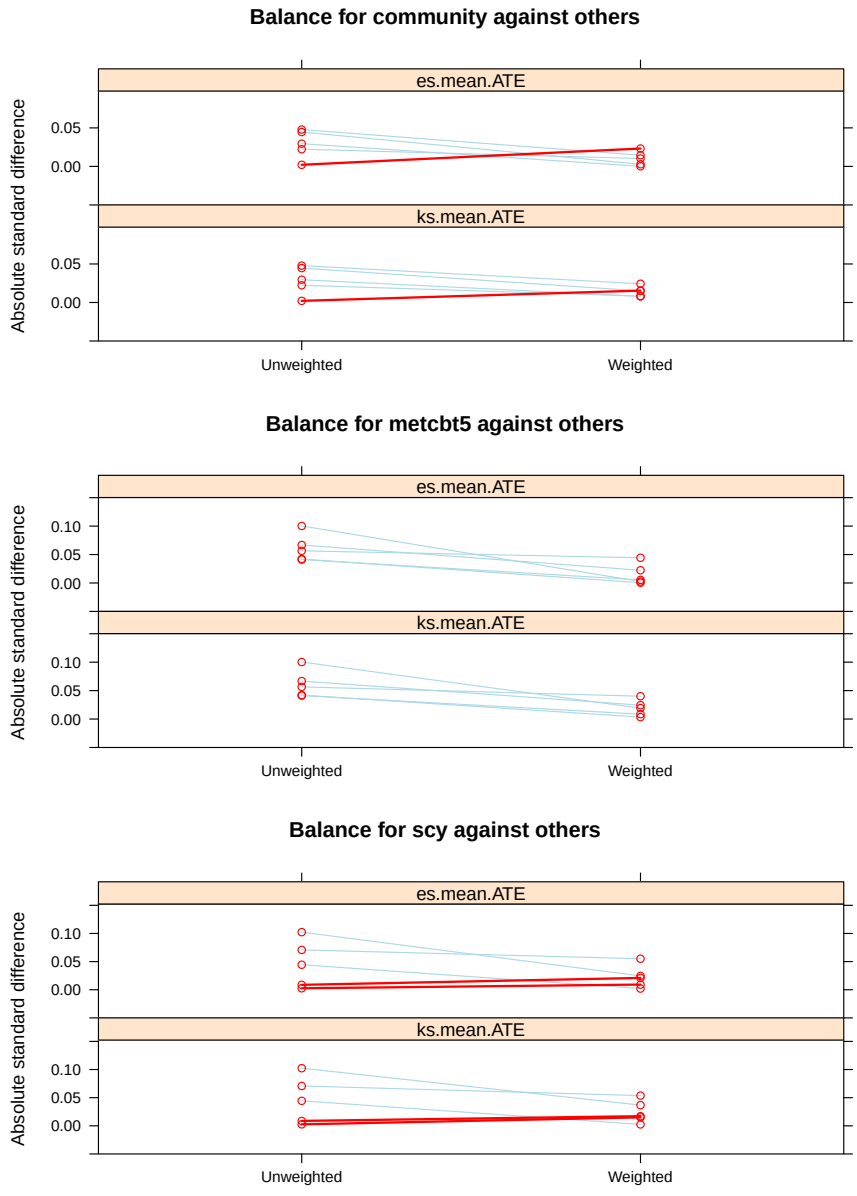
As with the `ps` function for the binary treatment setting, the default plotting function for `mnps`-class objects also displays information on commonly-used balance statistics. In particular, it provides comparisons of the absolute standard differences (setting the `plots` argument equal to 3) and t statistics (with the `plots` argument equal to 4), before and after weighting. However, whereas there is a single plot for these balance diagnostics in the binary treatment setting, in the multiple treatment case, one can either examine a plot for each of the treatment conditions, or summarize the balance statistics in some way, across the treatment conditions. As a default, the `plot` function for an `mnps` object returns the maximum of the balance statistics across treatment groups for each of the covariates:

```
> plot(mnps.AOD, plots = 3)
```



If any of the differences had been statistically significant (before taking the maximum across treatment groups), the corresponding hollow circles in this plot would be solid. One may see the balance plots for the individual fits by setting the `pairwiseMax` argument to `FALSE`.

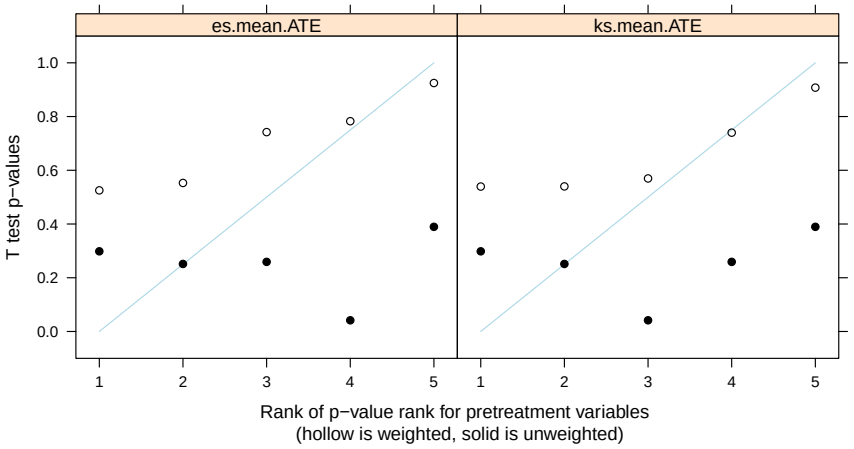
```
> plot(mnps.AOD, plots = 3, pairwiseMax = FALSE, figureRows = 3)
```



The additional `figureRows` argument instructs the function to spread the plots over three rows; by default the plots would be arranged in a single row rather than a column.

Setting the `plots` argument equal to 4 compares weighted and unweighted t-test or χ^2 statistic p-values for differences between each of the individual treatment groups and observations in all other treatment groups.

```
>     plot(mnps.AOD, plots = 4)
```



Beyond graphics, there are several other functions that may be of interest to `mnps` users. The first is `means.table` which provides a simple summary of balance across the groups. When `estimand` is set as `'ATE'`, the table shows the population means for each pretreatment covariate in the first column as well as each treatment group's unweighted and ATE weighted means and corresponding unweighted and weighted population standardized mean differences. As shown in the table below, incorporation of the ATE propensity score weights improves each treatment groups overall balance with the population means for each pretreatment covariate. The function also includes an argument called `includeSD` whose default is `FALSE`; changing it to `TRUE` returns standard deviations for each of the treatment conditions (not shown).

```
> means.table(mnps.AOD, stop.method = "es.mean", digits = 3)
```

```
         pop.mean unwt.community.mean wt.community.mean unwt.community.smd
illact      0.075                0.097             0.085               0.022
crimjust   -0.068               -0.065            -0.092               0.002
subprob    -0.016               -0.060            -0.013              -0.045
subdep      0.015                0.046             0.015               0.030
white       0.178                0.160             0.173              -0.048
         wt.community.smd unwt.metcbt5.mean wt.metcbt5.mean unwt.metcbt5.smd
illact              0.010              0.007           0.052           -0.067
crimjust           -0.023              0.037          -0.065            0.100
subprob             0.003              0.026          -0.016            0.042
subdep              0.000              0.058           0.021            0.041
white              -0.015              0.200           0.195            0.057
         wt.metcbt5.smd unwt.scy.mean wt.scy.mean unwt.scy.smd wt.scy.smd
illact           -0.022         0.120       0.077        0.044      0.002
crimjust          0.003        -0.174      -0.093       -0.102     -0.025
subprob           0.000        -0.013      -0.007        0.003      0.009
subdep            0.005        -0.058      -0.042       -0.071     -0.055
white             0.044         0.175       0.170       -0.009     -0.021
```

More extensive balance information is given by the `bal.table` function. For propensity score analyses with multiple treatments, this function returns a lot of information. The intention with

this function is that its ouput be loaded into a spreadsheet software program. (E.g., one can write the output into a .csv file using the `write.csv` function and open the resulting file using a spreadsheet application.) For each outcome category, and each stopping rule (in addition to the unweighted analysis) the `bal.table` function gives balance statistics such as weighted and unweighted means by treatment group. As of version 1.4 of TWANG, the balance measures are given for all pairwise combinations. (Prior to that version the balance measures were reported for each treatment against all others; we feel that the pairwise comparisons give a fuller accounting of balance in ATE applications.)

```
> bal.table(mnps.AOD, digits = 2)
```

	tmt1	tmt2	var	mean1	mean2	pop.sd	std.eff.sz	p	ks	ks.pval
1	community	metcbt5	illact	0.10	0.01	1.01	0.09	0.38	0.10	0.27
2	community	metcbt5	crimjust	-0.07	0.04	1.04	0.10	0.33	0.10	0.22
3	community	metcbt5	subprob	-0.06	0.03	0.98	0.09	0.39	0.09	0.39
4	community	metcbt5	subdep	0.05	0.06	1.03	0.01	0.91	0.06	0.92
5	community	metcbt5	white	0.16	0.20	0.38	0.10	0.30	0.04	1.00
6	community	scy	illact	0.10	0.12	1.01	0.02	0.82	0.06	0.87
7	community	scy	crimjust	-0.07	-0.17	1.04	0.10	0.30	0.08	0.55
8	community	scy	subprob	-0.06	-0.01	0.98	0.05	0.63	0.09	0.39
9	community	scy	subdep	0.05	-0.06	1.03	0.10	0.31	0.08	0.47
10	community	scy	white	0.16	0.18	0.38	0.04	0.69	0.02	1.00
11	metcbt5	scy	illact	0.01	0.12	1.01	0.11	0.26	0.11	0.18
12	metcbt5	scy	crimjust	0.04	-0.17	1.04	0.20	0.04	0.13	0.07
13	metcbt5	scy	subprob	0.03	-0.01	0.98	0.04	0.70	0.06	0.79
14	metcbt5	scy	subdep	0.06	-0.06	1.03	0.11	0.25	0.09	0.39
15	metcbt5	scy	white	0.20	0.18	0.38	0.07	0.52	0.02	1.00
16	community	metcbt5	illact	0.09	0.05	1.01	0.03	0.74	0.06	0.90
17	community	metcbt5	crimjust	-0.09	-0.06	1.04	0.03	0.79	0.05	0.93
18	community	metcbt5	subprob	-0.01	-0.02	0.98	0.00	0.97	0.06	0.83
19	community	metcbt5	subdep	0.02	0.02	1.03	0.01	0.96	0.05	0.96
20	community	metcbt5	white	0.17	0.20	0.38	0.06	0.58	0.02	1.00
21	community	scy	illact	0.09	0.08	1.01	0.01	0.94	0.05	0.97
22	community	scy	crimjust	-0.09	-0.09	1.04	0.00	0.99	0.04	1.00
23	community	scy	subprob	-0.01	-0.01	0.98	0.01	0.95	0.07	0.77
24	community	scy	subdep	0.02	-0.04	1.03	0.06	0.58	0.05	0.96
25	community	scy	white	0.17	0.17	0.38	0.01	0.95	0.00	1.00
26	metcbt5	scy	illact	0.05	0.08	1.01	0.02	0.81	0.06	0.79
27	metcbt5	scy	crimjust	-0.06	-0.09	1.04	0.03	0.78	0.06	0.90
28	metcbt5	scy	subprob	-0.02	-0.01	0.98	0.01	0.92	0.04	1.00
29	metcbt5	scy	subdep	0.02	-0.04	1.03	0.06	0.55	0.06	0.79
30	metcbt5	scy	white	0.20	0.17	0.38	0.07	0.53	0.03	1.00
31	community	metcbt5	illact	0.08	0.05	1.01	0.03	0.74	0.06	0.84
32	community	metcbt5	crimjust	-0.08	-0.05	1.04	0.03	0.72	0.05	0.95
33	community	metcbt5	subprob	0.00	-0.01	0.98	0.01	0.91	0.05	0.93
34	community	metcbt5	subdep	0.01	0.02	1.03	0.02	0.87	0.05	0.97
35	community	metcbt5	white	0.17	0.19	0.38	0.06	0.54	0.02	1.00
36	community	scy	illact	0.08	0.08	1.01	0.01	0.96	0.05	0.99
37	community	scy	crimjust	-0.08	-0.11	1.04	0.02	0.83	0.04	1.00
38	community	scy	subprob	0.00	0.00	0.98	0.00	1.00	0.06	0.88
39	community	scy	subdep	0.01	-0.04	1.03	0.05	0.65	0.05	0.96

```
40 community     scy    white  0.17  0.17   0.38       0.01 0.94 0.00    1.00
41   metcbt5     scy   illact  0.05  0.08   1.01       0.03 0.79 0.06    0.83
42   metcbt5     scy crimjust -0.05 -0.11   1.04       0.06 0.57 0.06    0.81
43   metcbt5     scy  subprob -0.01  0.00   0.98       0.01 0.91 0.04    1.00
44   metcbt5     scy   subdep  0.02 -0.04   1.03       0.06 0.54 0.06    0.80
45   metcbt5     scy    white  0.19  0.17   0.38       0.06 0.58 0.02    1.00
   stop.method
1          unw
2          unw
3          unw
4          unw
5          unw
6          unw
7          unw
8          unw
9          unw
10         unw
11         unw
12         unw
13         unw
14         unw
15         unw
16     es.mean
17     es.mean
18     es.mean
19     es.mean
20     es.mean
21     es.mean
22     es.mean
23     es.mean
24     es.mean
25     es.mean
26     es.mean
27     es.mean
28     es.mean
29     es.mean
30     es.mean
31     ks.mean
32     ks.mean
33     ks.mean
34     ks.mean
35     ks.mean
36     ks.mean
37     ks.mean
38     ks.mean
39     ks.mean
40     ks.mean
41     ks.mean
42     ks.mean
43     ks.mean
```

```
44     ks.mean
45     ks.mean
```

More parsimonious versions of the summaries are available using the `collapse.to` argument. Setting `collapse.to = 'covariate'` gives maximum pairwise difference measures for each pretreatment covariate and stopping rule. Setting `collapse.to = 'stop.method'` further collapses, giving only the maximum pairwise differences (and minimum p-values) across covariates.

```
> bal.table(mnps.AOD, collapse.to = "covariate", digits = 4)

        var max.std.eff.sz  min.p max.ks min.ks.pval stop.method
1    illact         0.1112 0.2591 0.1100      0.1779         unw
2  crimjust         0.2027 0.0416 0.1300      0.0680         unw
3   subprob         0.0867 0.3896 0.0900      0.3935         unw
4    subdep         0.1120 0.2514 0.0900      0.3935         unw
5     white         0.1044 0.2984 0.0400      0.9973         unw
6    illact         0.0326 0.7421 0.0647      0.7934     es.mean
7  crimjust         0.0272 0.7827 0.0568      0.8964     es.mean
8   subprob         0.0097 0.9248 0.0666      0.7664     es.mean
9    subdep         0.0605 0.5529 0.0646      0.7944     es.mean
10    white         0.0653 0.5253 0.0250      1.0000     es.mean
11   illact         0.0327 0.7401 0.0627      0.8278     ks.mean
12 crimjust         0.0560 0.5696 0.0640      0.8072     ks.mean
13  subprob         0.0116 0.9077 0.0583      0.8799     ks.mean
14   subdep         0.0623 0.5400 0.0646      0.7980     ks.mean
15    white         0.0645 0.5395 0.0247      1.0000     ks.mean

> bal.table(mnps.AOD, collapse.to = "stop.method", digits = 4)

  max.std.eff.sz  min.p max.ks min.ks.pval stop.method
1         0.2027 0.0416 0.1300      0.0680         unw
2         0.0653 0.5253 0.0666      0.7664     es.mean
3         0.0645 0.5395 0.0646      0.7980     ks.mean
```

Finally, there is also `summary` method for `mnps` objects which gives some information on balance measures as well as the effective sample sizes for each treatment group under each stopping rule.

```
> summary(mnps.AOD)

Summary of pairwise comparisons:
  max.std.eff.sz      min.p     max.ks min.ks.pval stop.method
1     0.20266446 0.04161562 0.13000000   0.0680192         unw
2     0.06529298 0.52525235 0.06661985   0.7663959     es.mean
3     0.06448455 0.53947426 0.06460328   0.7979986     ks.mean

Sample sizes and effective sample sizes:
  treatment   n ESS.es.mean ESS:ks.mean
1 community 200    184.5124    187.4713
2   metcbt5 200    186.1874    183.3987
3       scy 200    189.5017    185.7158
```

After examining the graphical and tabular diagnostics provided by `twang`, we can analyze the outcome variable using the propensity scores generated by the `mnps` function. Although two stop methods were specified initially (`es.mean` and `ks.mean`), at this point we have to commit to a single set of weights. From the `bal.table` call above, we see that the balance properties are very similar for the two stopping rules, and from the `summary` statement, we see that the effective sample sizes (ESS) are similar as well. Hence, we expect the two stop methods to give similar results; we choose to analyze the data with the `es.mean` weights.

In order to analyze the data using the weights, it is recommended that one use the `survey` package, which performs weighted analyses. We can add the weights to the dataset using the `get.weights` function and specify the survey design as follows:

```
> require(survey)
> AOD$w <- get.weights(mnps.AOD, stop.method = "es.mean")
> design.mnps <- svydesign(ids=~1, weights=~w, data=AOD)
```

As shown in the `ps` vignette, we can then perform the propensity score-adjusted regression using the `svyglm` function:

```
> glm1 <- svyglm(suf12 ~ as.factor(treat), design = design.mnps)
> summary(glm1)

Call:
svyglm(formula = suf12 ~ as.factor(treat), design = design.mnps)

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

Coefficients:
                        Estimate Std. Error t value
(Intercept)             -0.09913    0.06736  -1.472
as.factor(treat)metcbt5  0.14858    0.10502   1.415
as.factor(treat)scy      0.06464    0.09998   0.647
                        Pr(>|t|)
(Intercept)                0.142
as.factor(treat)metcbt5    0.158
as.factor(treat)scy        0.518

(Dispersion parameter for gaussian family taken to be 1.002082)

Number of Fisher Scoring iterations: 2
```

Using this small subset of the data, we are unable to detect differences in the treatment group means. However, the coefficient for the metcbt5 term represents the causal effect of metcbt5 vs. community and the coefficient for the scy term represents the causal effect of scy vs. community assuming there are no unobserved confounders. In the context of this application, the signs of the estimates correspond to higher substance use frequency for youths exposed to metcbt5 or scy relative to community. More details on how to obtain all relevant pairwise differences can be found in McCaffrey et al. (2013).

3 An ATT example

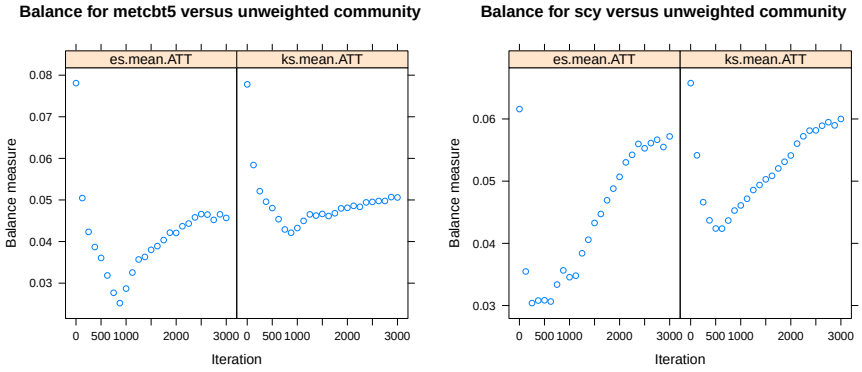
It is also possible to explore treatment effects on the treated (ATTs) using the `mnps` function. A key difference in the multiple treatment setting is that we must be clear as to which treatment

condition "the treated" refers to. This is done through the `treatATT` argument. Here, we define the treatment group of interest to be the community group; thus, we are trying to draw inferences about the relative effectiveness of the three treatment groups for individuals like those who were enrolled in the community program.

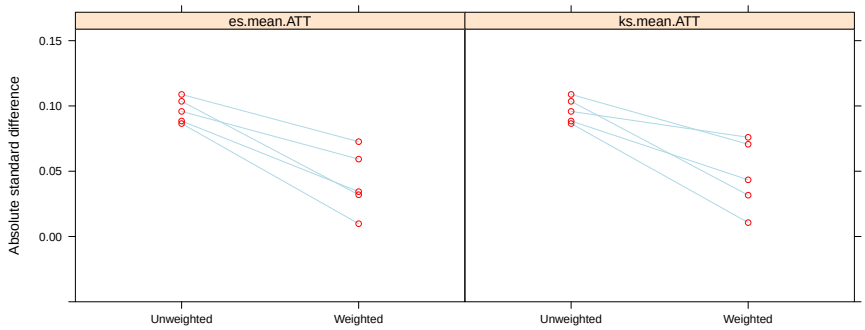
```
> mnps.AOD.ATT <- mnps(treat ~ illact + crimjust + subprob + subdep + white,
+                      data = AOD, estimand = "ATT", treatATT = "community",
+                      verbose = FALSE, n.trees = 3000,
+                      stop.method = c("es.mean", "ks.mean"))
```

The same array of visual and numerical summaries are available as they were in the ATE analysis.

```
>     plot(mnps.AOD.ATT, plots = 1)
```



```
>     plot(mnps.AOD.ATT, plots = 3)
```



Although the same basic graphical descriptions are available as in the ATE case, note that the population means above are replaced with the means of the `treatATT` category in the `means.table` call.

```
> means.table(mnps.AOD.ATT, digits = 3)
```

```
         community.mean unwt.metcbt5.mean wt.metcbt5.mean unwt.metcbt5.smd
illact            0.097             0.007           0.087            0.087
crimjust         -0.065             0.037          -0.032           -0.097
subprob          -0.060             0.026          -0.062           -0.088
subdep            0.046             0.058           0.058           -0.011
white             0.160             0.200           0.187           -0.109
         wt.metcbt5.smd unwt.scy.mean wt.scy.mean unwt.scy.smd wt.scy.smd
illact            0.010         0.120       0.100       -0.021     -0.002
crimjust         -0.032        -0.174      -0.064        0.104     -0.002
subprob           0.003        -0.013      -0.027       -0.048     -0.034
subdep           -0.012        -0.058      -0.018        0.096      0.059
white            -0.073         0.175       0.176       -0.041     -0.045
```

The `bal.table` output is similar to the ATE case. However, for ATT, we only report pairwise comparisons that include the `treatATT` category.

```
> bal.table(mnps.AOD.ATT, digits = 2)
```

```
Note that `tx` refers to the category specified as the treatATT, community.

        var tx.mn tx.sd ct.mn ct.sd std.eff.sz  stat    p   ks ks.pval control
1    illact  0.10  1.04  0.01  1.03       0.09  0.87 0.38 0.10    0.27 metcbt5
2  crimjust -0.07  1.05  0.04  1.04      -0.10 -0.98 0.33 0.10    0.22 metcbt5
3   subprob -0.06  0.97  0.03  1.02      -0.09 -0.86 0.39 0.09    0.39 metcbt5
4    subdep  0.05  1.08  0.06  1.05      -0.01 -0.11 0.91 0.06    0.92 metcbt5
5     white  0.16  0.37  0.20  0.40      -0.11 -1.04 0.30 0.04    1.00 metcbt5
6    illact  0.10  1.04  0.12  0.96      -0.02 -0.22 0.82 0.06    0.87     scy
7  crimjust -0.07  1.05 -0.17  1.03       0.10  1.05 0.30 0.08    0.55     scy
8   subprob -0.06  0.97 -0.01  0.97      -0.05 -0.48 0.63 0.09    0.39     scy
9    subdep  0.05  1.08 -0.06  0.96       0.10  1.01 0.31 0.08    0.47     scy
10    white  0.16  0.37  0.18  0.38      -0.04 -0.40 0.69 0.02    1.00     scy
```

```
11   illact  0.10  1.04  0.09  1.02       0.01  0.09 0.93 0.04    1.00 metcbt5
12 crimjust -0.07  1.05 -0.03  1.00      -0.03 -0.32 0.75 0.05    0.96 metcbt5
13  subprob -0.06  0.97 -0.06  0.99       0.00  0.02 0.98 0.04    1.00 metcbt5
14   subdep  0.05  1.08  0.06  1.05      -0.01 -0.11 0.91 0.05    0.96 metcbt5
15    white  0.16  0.37  0.19  0.39      -0.07 -0.68 0.50 0.03    1.00 metcbt5
16   illact  0.10  1.04  0.10  1.01       0.00 -0.02 0.98 0.06    0.90     scy
17 crimjust -0.07  1.05 -0.06  1.00       0.00 -0.02 0.99 0.05    0.94     scy
18  subprob -0.06  0.97 -0.03  0.97      -0.03 -0.34 0.74 0.06    0.90     scy
19   subdep  0.05  1.08 -0.02  0.99       0.06  0.60 0.55 0.07    0.71     scy
20    white  0.16  0.37  0.18  0.38      -0.04 -0.43 0.67 0.02    1.00     scy
21   illact  0.10  1.04  0.09  1.02       0.01  0.10 0.92 0.04    1.00 metcbt5
22 crimjust -0.07  1.05 -0.03  1.00      -0.03 -0.31 0.75 0.05    0.96 metcbt5
23  subprob -0.06  0.97 -0.06  0.99       0.00  0.02 0.99 0.04    1.00 metcbt5
24   subdep  0.05  1.08  0.06  1.05      -0.01 -0.10 0.92 0.05    0.96 metcbt5
25    white  0.16  0.37  0.19  0.39      -0.07 -0.66 0.51 0.03    1.00 metcbt5
26   illact  0.10  1.04  0.10  1.04       0.00 -0.01 1.00 0.05    0.96     scy
27 crimjust -0.07  1.05 -0.04  0.97      -0.02 -0.23 0.81 0.04    1.00     scy
28  subprob -0.06  0.97 -0.02  0.98      -0.04 -0.40 0.69 0.04    0.99     scy
29   subdep  0.05  1.08 -0.04  0.99       0.08  0.74 0.46 0.07    0.66     scy
30    white  0.16  0.37  0.16  0.37      -0.01 -0.08 0.94 0.00    1.00     scy
   stop.method
1          unw
2          unw
3          unw
4          unw
5          unw
6          unw
7          unw
8          unw
9          unw
10         unw
11 es.mean.ATT
12 es.mean.ATT
13 es.mean.ATT
14 es.mean.ATT
15 es.mean.ATT
16 es.mean.ATT
17 es.mean.ATT
18 es.mean.ATT
19 es.mean.ATT
20 es.mean.ATT
21 ks.mean.ATT
22 ks.mean.ATT
23 ks.mean.ATT
24 ks.mean.ATT
25 ks.mean.ATT
26 ks.mean.ATT
27 ks.mean.ATT
28 ks.mean.ATT
29 ks.mean.ATT
```

```
30 ks.mean.ATT
```

The process to analyze the outcome variable is also similar:

```
> require(survey)
> AOD$w.ATT <- get.weights(mnps.AOD.ATT, stop.method = "es.mean")
> design.mnps.ATT <- svydesign(ids=~1, weights=~w.ATT, data=AOD)

> glm1 <- svyglm(suf12 ~ as.factor(treat), design = design.mnps.ATT)
> summary(glm1)

Call:
svyglm(formula = suf12 ~ as.factor(treat), design = design.mnps.ATT)

Survey design:
svydesign(ids = ~1, weights = ~w.ATT, data = AOD)

Coefficients:
                         Estimate Std. Error t value
(Intercept)              -0.10505    0.06383  -1.646
as.factor(treat)metcbt5   0.20071    0.10409   1.928
as.factor(treat)scy       0.08076    0.09901   0.816
                         Pr(>|t|)
(Intercept)                0.1003
as.factor(treat)metcbt5    0.0543 .
as.factor(treat)scy        0.4150
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for gaussian family taken to be 0.9746663)

Number of Fisher Scoring iterations: 2
```

Note in this case that the estimated treatment effect of community on those exposed to the community treatment is slightly stronger than in the ATE case (high numbers are bad for the outcome variable). Although not statistically significant, such differences are compatible with the notion that the youths who actually received the community treatment responded more favorably to it than the "average" youth would have (where the average is taken across the whole collection of youths enrolled in the study).

The discussion in McCaffrey et al. (2013) may be useful for determining whether the ATE or ATT is of greater interest in a particular application.

4 Conclusion

Often, more than two treatments are available to study participants. If the study is not randomized, analysts may be interested in using a propensity score approach. Previously, few tools existed to aide the analysis of such data, perhaps tempting analysts to ignore all but two of the treatment conditions. We hope that this extension to the `twang` package will encourage more appropriate analyses of observational data with more than two treatment conditions.

Acknowledgements

The random subset of data was supported by the Center for Substance Abuse Treatment (CSAT), Substance Abuse and Mental Health Services Administration (SAMHSA) contract # 270-07-0191 using data provided by the following grantees: Adolescent Treatment Model (Study: ATM: CSAT/SAMHSA contracts # 270-98-7047, # 270-97-7011, #2 77-00-6500, # 270-2003-00006 and grantees: TI-11894, TI-11874, TI-11892), the Effective Adolescent Treatment (Study: EAT; CSAT/SAMHSA contract # 270-2003-00006 and grantees: TI-15413, TI-15433, TI-15447, TI-15461, TI-15467, TI-15475, TI-15478, TI-15479, TI-15481, TI-15483, TI-15486, TI-15511, TI15514, TI-15545, TI-15562, TI-15670, TI-15671, TI-15672, TI-15674, TI-15678, TI-15682, TI-15686, TI-15415, TI-15421, TI-15438, TI-15446, TI-15458, TI-15466, TI-15469, TI-15485, TI-15489, TI-15524, TI-15527, TI-15577, TI-15584, TI-15586, TI-15677), and the Strengthening Communities-Youth (Study: SCY; CSAT/SAMHSA contracts # 277-00-6500, # 270-2003-00006 and grantees: TI-13305, TI-13308, TI-13313, TI-13322, TI-13323, TI-13344, TI-13345, TI-13354). The authors thank these grantees and their participants for agreeing to share their data to support the development of the `mnps` functionality.

References

[1] McCaffrey, D.F., B.A. Griffin, D. Almirall, M.E. Slaughter, R. Ramchand, and L.F. Burgette (2013). "A tutorial on propensity score estimation for multiple treatments using generalized boosted models." Forthcoming at *Statistics in Medicine*.

[2] Ridgeway, G., D. McCaffrey, B.A. Griffin, and L. Burgette (2006). "twang: Toolkit for weighting and analysis of non-equivalent groups." Available at http://cran.fyxm.net/web/packages/twang/.