b
UNIVERSITÄT
BERN

Faculty of Business, Economics and
Social Sciences

Department of Social Sciences

University of Bern Social Sciences Working Paper No. 32

Influence functions for linear regression (with an application to regression adjustment)

Ben Jann

Current version: March 30, 2019
First version: March 27, 2019

http://ideas.repec.org/p/bss/wpaper/32.html
http://econpapers.repec.org/paper/bsswpaper/32.htm

University of Bern
Department of Social Sciences
Fabrikstrasse 8
CH-3012 Bern

Tel. +41 (0)31 631 48 11
Fax +41 (0)31 631 48 17
info@sowi.unibe.ch
www.sowi.unibe.ch

Influence functions for linear regression (with an application to regression adjustment)

Ben Jann

March 30, 2019

Abstract

Influence functions are useful, for example, because they provide an easy and flexible way to estimate standard errors. This paper contains a brief overview of influence functions in the context of linear regression and illustrates how their empirical counterparts can be computed in Stata, both for unweighted data and for weighted data. Influence functions for regression-adjustment estimators of average treatment effects are also covered.

Keywords: Influence function, sampling variance, sampling weights, standard error, linear regression, mean difference, regression adjustment, average treatment effect, causal inference

Contents

1 Influence functions and standard errors

Given some distribution F, the influence function of statistic $\hat{\theta}(F)$ is defined as

$$IF_{\hat{\theta}}(x) = \lim_{\epsilon \to 0} \frac{\hat{\theta}(F_\epsilon(x)) - \hat{\theta}(F)}{\epsilon}$$

with $F_\epsilon(x) = (1-\epsilon)F + \epsilon\delta_x$, where δ_x is a probability distribution with all its mass at point x. The influence function thus quantifies how statistic $\hat{\theta}(F)$ changes if distribution F is contaminated by a small amount if data mass at point x. Stated differently, it quantifies the *influence* of data point x on $\hat{\theta}$.

Influence functions have many uses. For example, they are an important tool in robust statistics to describe the robustness of an estimator (robust estimators have a bounded influence function). Another use is in estimating standard errors. It can be shown that, asymptotically, the sampling variance of $\hat{\theta}$ is equal to

$$V_{\hat{\theta}} = \frac{E_F[IF_{\hat{\theta}}(x)^2]}{N}$$

where N is the sample size. Since the expectation of $IF_{\hat{\theta}}$ is zero, that is $E_F[IF_{\hat{\theta}}(x)] = 0$, we can also simply use the variance of $IF_{\hat{\theta}}$ in the numerator. The idea now is to use the relationship between variance and influence function to estimate the standard error of a statistic in a given sample: once you have worked out the influence function for a statistic, you can estimate its sampling variance by evaluating the influence function for each observation in the data (replacing theoretical quantities by their sample counterparts) and then computing the standard error of the mean of these values.[1] The standard error obtained from the influence function provides a consistent estimate for the standard error of your statistic. Furthermore, if you are computing multiple estimates (in any combination: a particular type of statistic for different variables or different subpopulations, a series of different types of statistics for a single variable, multiple types of statistics for multiple variables and multiple subpopulations), the full matrix of sampling variances and covariances among these estimates can easily be obtained by generating a series of influence function variables (one for each estimate) and then computing the sampling variance matrix of the means of these variables using standard techniques. This makes the influence-function approach very flexible and powerful.

The only difficulty, of course, is to derive the appropriate influence function in a given situation. This can be easy, as in the case of the mean, but it can also be quite complicated. I have to admit that I never fully understood how influence functions are derived in practice (my math is too poor; I have no clue, for example, what a Gâteaux derivative is). However, I managed to piece things together for linear regression, with intuition, some trial and error, and a little help from Kahn (2015) and jayk (2015).

[1] The property that the mean of the influence function is zero also holds for the empirical counterpart in the sample. If the mean is not zero (apart from roundoff error), then you know that you made a mistake.

One property of influence functions I will make use of below is the *chain rule*. It states that if your statistic of interest is a function of other statistics, that is, if $\hat{\theta} = T(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_k)$, then

$$IF_{\hat{\theta}}(x) = \frac{\partial T}{\partial \hat{\theta}_1} IF_{\hat{\theta}_1}(x) + \frac{\partial T}{\partial \hat{\theta}_2} IF_{\hat{\theta}_2}(x) + \cdots + \frac{\partial T}{\partial \hat{\theta}_k} IF_{\hat{\theta}_k}(x)$$

2 How to handle weights

There are two ways to deal with sampling weights. A common approach is to use weighted estimates for all influence-function components that need to be estimated, but not include the weights themselves in the influence function. The weights are then taken into account in a second step when estimating the sampling variance from the influence function (i.e., using `pweights`). Another approach is to fully integrate the weights into the influence function (i.e., derive the influence function for a weighted statistic) and then estimate the sampling variance ignoring the weights. I follow this second approach because in the application I have in mind I want to be able to estimate the covariance between weighted and unweighted statistics, which is easier if the weights are integrated into the influence function. This is just a matter of convenience, as both approaches lead to the same results.

3 Example data

The examples in this paper use the LaLonde (1986) data, as provided by Dehejia and Wahba (1999) at http://www.nber.org/~rdehejia/nswdata.html. The following code combines the treatment group from the NSW training program with one of the PSID comparison groups.

```
. infile treat age education black hispanic married nodegree ///
>     re74 re75 re78 using nswre74_treated.txt, clear
(185 observations read)

. save tmp.dta, replace
(note: file tmp.dta not found)
file tmp.dta saved

. infile treat age education black hispanic married nodegree ///
>     re74 re75 re78 using psid2_controls.txt, clear
(253 observations read)

. append using tmp.dta

. erase tmp.dta
```

For purpose of illustration, I generate some sampling weights:

```
. set seed 3423098

. generate double w = rnormal(1000, 200)

. summarize w

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
```

```
           w |        438    998.6502    195.1196   390.8385   1650.427
```

4 Influence functions for various statistics

4.1 The mean

Given a sample y_i, $i = 1, \ldots, N$, the values of the influence function for the mean

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

can be computed as

$$\lambda_i^{\bar{y}} = y_i - \bar{y}$$

where I use $\lambda_i^{\bar{y}}$ as shorthand notation for $\widehat{IF}_{\bar{y}}(y_i)$. Hence, an estimate of the sampling variance of $\bar{Y}$ can be computed as

$$\hat{V}_{\bar{y}} = \frac{1}{N}\left(\frac{1}{N-1}\sum_{i=1}^{N}(\lambda_i^{\bar{y}})^2\right) = \frac{1}{N}\left(\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2\right)$$

The above results are valid for an unweighted mean in a simple random sample. In an unequal probability sample, the weighted mean

$$\bar{y} = \frac{1}{W}\sum_{i=1}^{N} w_i y_i$$

provides a consistent estimate of the population average, where w_i, $i = 1, \ldots, N$, are the sampling weights (inverse of the sampling probabilities) and $W = \sum w_i$ is the sum of the weights (the population size). In this case, one way to compute the values of the influence function in the sample is

$$\lambda_i^{\bar{y}} = w_i \frac{N}{W}(y_i - \bar{y})$$

such that an estimate of the sampling variance is given as

$$\hat{V}_{\bar{y}} = \frac{1}{N}\left(\frac{1}{N-1}\sum_{i=1}^{N}\left(w_i \frac{N}{W}(y_i - \bar{y})\right)^2\right)$$

As mentioned above, an alternative approach would be to omit $w_i \frac{N}{W}$ from the influence function, but then take account of the weights when estimating the sampling variance.

▷ **Example**

Here is an example that illustrates the computation of the influence function for a weighted mean. We are interested in the mean of `re78` (earnings in 1978). Assuming the weights to be fixed, we can estimate the mean and its standard error based on the influence function as follows:

```
. mata:
------------------------------------------------- mata (type end to exit) -------
: y = st_data(., "re78"); N = rows(y)
: w = st_data(., "w");    W = sum(w)
: ybar = sum(w :* y) / W  // or simply type mean(y, w)
: IF = w*(N/W) :* (y :- ybar)
: ybar, sqrt(variance(IF)/N)
                 1              2
    +-------------------------------+
  1 |  8343.541322   484.2000874  |
    +-------------------------------+

: end
---------------------------------------------------------------------------------
```

The influence-function based standard error is identical to the default standard error as computed by Stata's `mean` command if `pweight`s are applied:

```
. mean re78 [pweight = w]
Mean estimation                   Number of obs   =        438
```

	Mean	Std. Err.	[95% Conf.	Interval]
re78	8343.541	484.2001	7391.891	9295.192

◁

4.2 Regression coefficients

Given is a sample $(y_i, x_{i1}, x_{i2}, \ldots, x_{ik})$, $i = 1, \ldots, N$, with $x_{i1} = 1$ for all i (the constant). Let x_i denote the vector $(x_{i1}, x_{i2}, \ldots, x_{ik})$. Furthermore, let w_i, $i = 1, \ldots, N$, be weights with $W = \sum w_i$ (for unweighted data, set $w_i = 1$ for all i). We are interested in the influence function for the least-squares estimate of coefficient vector $\beta = (\beta_1, \beta_2, \ldots, \beta_k)'$ in the linear regression

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_k + \upsilon_i = x_i\beta + \upsilon_i$$

The $k \times 1$ vector of observation i's influence values for $\hat{\beta}$ can be computed as

$$\lambda_i^{\hat{\beta}} = w_i \frac{N}{W} \mathbf{Q} x_i'(y_i - x_i\hat{\beta})\sqrt{\frac{N-1}{N-k}}$$

with

$$\mathbf{Q} = \left(\widehat{E}[X'X]\right)^{-1} = \left(\frac{X'\omega X}{W}\right)^{-1} = \left(\frac{1}{W}\sum_{i=1}^{N} w_i x_i' x_i\right)^{-1}$$

where $X = (x_1', x_2', \ldots, x_n)'$ is the $N \times k$ data matrix of covariate values, ω is a $N \times N$ diagonal matrix with $(w_1, w_2, \ldots, w_N)$ on the diagonal, and $\sqrt{(N-1)(N-k)}$ is a correction for the degrees of freedom taking into account the number of estimated coefficients (so that variances can be estimated using the default $N-1$ denominator). If the model only has a constant, the regression influence function is equivalent to the influence function for the mean above. In this case, $X'\omega X = W$, $x_i = 1$, $k = 1$ and $\hat{\beta} = \bar{y}$, such that

$$\left(\frac{X'\omega X}{W}\right)^{-1} x_i'(y_i - x_i\hat{\beta})\sqrt{\frac{N-1}{N-k}} = \left(\frac{W}{W}\right)^{-1} 1(y_i - 1\hat{\beta})\sqrt{\frac{N-1}{N-1}} = (y_i - \hat{\beta}) = (y_i - \bar{y})$$

▷ **Example**

I first illustrate how to estimate the standard errors of the coefficients of a linear regression without weights. I slightly rearrange the formula for the influence function so that a loop over i can be avoided.

```
. regress re78 education re74, noheader
------------------------------------------------------------------------------
        re78 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   education |   836.1177   152.3052     5.49   0.000     536.7721    1135.463
        re74 |   .4437065   .0425331    10.43   0.000     .3601106    .5273024
       _cons |  -3617.041   1625.905    -2.22   0.027    -6812.648   -421.4332
------------------------------------------------------------------------------

. mata:
------------------------------------------------- mata (type end to exit) -----
: b = st_matrix("e(b)")'

: y = st_data(., "re78"); N = rows(y)

: X = st_data(., "education re74"), J(N, 1, 1); k = cols(X)

: Q = invsym(cross(X, X) / N)

: IF = (y :- X*b) :* X * Q * sqrt((N-1)/(N-k))

: V = variance(IF) / N; SE = diagonal(sqrt(V))

: SE
                 1
    +---------------+
  1 |  140.7161657  |
  2 |  .0844806401  |
  3 |  1534.108815  |
    +---------------+

: end
-------------------------------------------------------------------------------
```

The influence-function based standard errors are quite different from the default standard errors computed by `regress`. The reason is that no homoscedasticity assumption (constant error variance) is made. Hence, the influence-function based standard errors are "robust" and we need to compare them to standard errors from `regress` with option `robust` (Huber–White standard errors).

```
. regress re78 education re74, robust noheader
------------------------------------------------------------------------------
             |               Robust
```

```
      re78 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------+----------------------------------------------------------------
 education |   836.1177   140.7162     5.94   0.000     559.5496    1112.686
      re74 |   .4437065   .0844806     5.25   0.000     .2776655    .6097474
     _cons |  -3617.041   1534.109    -2.36   0.019    -6632.228   -601.8533
----------------------------------------------------------------------------
```

We see that the influence-function based standard errors are identical to the Huber–White standard errors.

In the above code I computed the full variance-covariance matrix of the coefficient estimates. Comparing the result to `e(V)` from `regress`, we see that the influence-function based covariances are identical to the covariances computed by `regress`. Hence, results for linear combinations (see `lincom`) and Wald tests (see `test`) will also be valid if computed from the influence-function based variance matrix.

```
. mata: V
[symmetric]
                  1              2              3
    +----------------------------------------------+
  1 |   19801.03928                                |
  2 |   .5917524484    .0071369786                 |
  3 |  -199009.3606   -45.42224456    2353489.855  |
    +----------------------------------------------+

. matrix list e(V)

symmetric e(V)[3,3]
            education        re74       _cons
education   19801.039
     re74   .59175245   .00713698
    _cons  -199009.36  -45.422245   2353489.9
```

We can now replicate the exercise including weights. Also here we see that the influence-function based results are identical to the robust standard errors from `regress`.

```
. regress re78 education re74 [pweight = w], noheader
(sum of wgt is 437,408.772201053)
----------------------------------------------------------------------------
           |               Robust
      re78 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------+----------------------------------------------------------------
 education |     830.35   142.3868     5.83   0.000     550.4983    1110.202
      re74 |   .4491048   .0796519     5.64   0.000     .2925544    .6056552
     _cons |  -3660.089   1529.306    -2.39   0.017    -6665.836   -654.3408
----------------------------------------------------------------------------

. matrix list e(V)

symmetric e(V)[3,3]
            education        re74       _cons
education   20273.998
     re74   .25365329   .00634442
    _cons     -202077  -36.757687   2338777.2

. mata:
---------------------------------------------------- mata (type end to exit) ------
```

```
: b = st_matrix("e(b)")'
: y = st_data(., "re78"); N = rows(y)
: X = st_data(., "education re74"), J(N, 1, 1); k = cols(X)
: w = st_data(., "w"); W = sum(w)
: Q = invsym(cross(X, w, X) / W)
: IF = w*(N/W) :* ((y :- X*b) :* X * Q) * sqrt((N-1)/(N-k))
: V = variance(IF) / N
: SE = diagonal(sqrt(V))
: SE
                 1
    +---------------+
  1 |   142.386789  |
  2 |  .0796518895  |
  3 |  1529.306106  |
    +---------------+

: V
[symmetric]
                  1               2               3
    +-----------------------------------------------+
  1 |   20273.99769                                 |
  2 |   .2536532859     .0063444235                 |
  3 |  -202076.9955    -36.75768671     2338777.165 |
    +-----------------------------------------------+

: end
```

◁

4.3 Predictions

Consider the statistic $\hat{\theta} = \frac{1}{W}\sum_{i=1}^{N} w_i x_i \hat{\beta}$, that is, the average prediction from the linear regression across the sample.[2] We can rewrite the statistic as follows:

$$\hat{\theta} = \frac{1}{W}\sum_{i=1}^{N} w_i x_i \hat{\beta} = \frac{1}{W}\sum_{i=1}^{N} w_i x_{i1} \hat{\beta}_1 + \frac{1}{W}\sum_{i=1}^{N} w_i x_{i2} \hat{\beta}_2 \cdots + \frac{1}{W}\sum_{i=1}^{N} w_i x_{ik} \hat{\beta}_k$$

$$= \hat{\beta}_1 \frac{1}{W}\sum_{i=1}^{N} w_i x_{i1} + \hat{\beta}_2 \frac{1}{N}\sum_{i=1}^{N} w_i x_{i2} \cdots + \hat{\beta}_k \frac{1}{W}\sum_{i=1}^{N} w_i x_{ik} = \hat{\beta}_1 \bar{x}_1 + \hat{\beta}_2 \bar{x}_2 \cdots + \hat{\beta}_k \bar{x}_k = \bar{x}\hat{\beta}$$

with $\bar{x}_j = \frac{1}{W}\sum_{i=1}^{N} w_i x_{ij}$ and $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k)$ (vector of means of the X variables). Using the chain rule for influence statistics (and ignoring the degrees-of-freedom correction), we

[2] Of course, because in linear regression the error term sums to zero, this is equal to the average of Y, that is, $\frac{1}{N}\sum_{i=1}^{N} w_i x_i \hat{\beta} = \frac{1}{W}\sum_{i=1}^{N} w_i y_i$. Hence, the influence function for the average prediction is simply the influence function for the mean, so that working out the details seems like a waste of time. However, these details will be useful for the more general case including out-of-sample predictions.

thus get

$$\lambda_i^{\hat{\theta}} = \bar{x}\left(w_i\frac{N}{W}\mathbf{Q}x_i'(y_i - x_i\hat{\beta})\right) + \left(w_i\frac{N}{W}(x_i - \bar{x})\right)\hat{\beta} = w_i\frac{N}{W}\left(\bar{x}\mathbf{Q}x_i'(y_i - x_i\hat{\beta}) + (x_i - \bar{x})\hat{\beta}\right)$$

The result happens to be exactly the same as

$$\lambda_i^{\bar{y}} = w_i\frac{N}{W}(y_i - \bar{y})$$

where $\bar{y} = \frac{1}{W}\sum_{i=1}^{N} w_i y_i$ (see footnote 2).

▷ Example

```
. regress re78 education re74 [iweight = w]
  (output omitted)
. mata:
------------------------------------------------- mata (type end to exit) ------
: b = st_matrix("e(b)")'
: y = st_data(., "re78"); N = rows(y)
: X = st_data(., "education re74"), J(N, 1, 1); k = cols(X)
: w = st_data(., "w"); W = sum(w)
: xbar = mean(X, w)
: Q = invsym(cross(X, w, X) / W)
: IF = w*(N/W) :* ((y :- X*b) :* X * Q * xbar' + (X :- xbar) * b)
: mean(X*b, w), sqrt(variance(IF)/N)
                 1             2
    +-----------------------------+
  1 |  8343.541322   484.2000874  |
    +-----------------------------+

: end
--------------------------------------------------------------------------------
```

The result is the same as the standard error of the mean of the dependent variable:

```
. mata: mean(y, w), sqrt(variance(w*(N/W) :* (y :- mean(y, w)))/N)
                 1             2
    +-----------------------------+
  1 |  8343.541322   484.2000874  |
    +-----------------------------+

. mean re78 [pweight = w]
Mean estimation                    Number of obs   =        438
```

	Mean	Std. Err.	[95% Conf.	Interval]
re78	8343.541	484.2001	7391.891	9295.192

◁

4.3.1 Degrees-of-freedom correction

Above I ignored the degrees-of-freedom correction for the number of coefficients in the regression model. If we include the correction, the influence function becomes

$$\lambda_i^{\hat{\theta}} = w_i \frac{N}{W} \left(\bar{x}\mathbf{Q}x_i'(y_i - x_i\hat{\beta}) + (x_i - \bar{x})\hat{\beta} \right) \sqrt{\frac{N-1}{N-k}}$$

In this case, results will no longer be exactly equal to the results for a simple mean estimate. However, the results will be consistent with how Stata's `margins` computes standard errors.

▷ **Example**

```
. mata:
──────────────────────────────────────────────── mata (type end to exit) ──────
: IF = IF * sqrt((N-1)/(N-k))

: mean(X*b, w), sqrt(variance(IF)/N)
                  1             2
    ┌─────────────────────────────┐
  1 │  8343.541322   485.3119145  │
    └─────────────────────────────┘

: end
───────────────────────────────────────────────────────────────────────────────

. regress re78 education re74 [pweight = w]
  (output omitted)
. margins, vce(unconditional)

Predictive margins                              Number of obs     =        438
Expression   : Linear prediction, predict()

──────────────────────────────────────────────────────────────────────────────
             │            Unconditional
             │     Margin   Std. Err.      t    P>|t|     [95% Conf. Interval]
─────────────┼────────────────────────────────────────────────────────────────
       _cons │   8343.541   485.3119    17.19   0.000     7389.694    9297.389
──────────────────────────────────────────────────────────────────────────────
```

Option `vce(unconditional)` has been added to prevent `margins` from assuming the X variables to be fixed.

◁

4.3.2 Out-of-sample predictions

Now consider the more interesting case where the predictions are averaged over a subpopulation or where out-of-sample predictions are made. The difference to the above is that in this case, two different subpopulations are used to estimate the regression coefficients and to estimates the means. Let R_i be an indicator for whether observation i is from the subpopulation that was used to identify the coefficients; likewise, let P_i be an indicator for whether observation i is from the subpopulation over which the predictions are averaged. Let

$W_R = \sum R_i w_i$ and $W_P = \sum P_i w_i$ be the sizes (sum of weights) of the two subpopulations; the subpopulations may overlap, that is, $W_R + W_P \geq W$. The statistic is defined as

$$\hat{\theta} = \frac{1}{W_P} \sum_{i=1}^{N} P_i w_i x_i \hat{\beta}_R = \bar{x}_P \hat{\beta}_R$$

and the influence function can then be written as

$$\begin{aligned} \lambda_i^{\hat{\theta}} &= w_i \frac{N}{W} \left(R_i \frac{W}{W_R} \bar{x}_P \mathbf{Q}_R x_i' (y_i - x_i \hat{\beta}_R) + P_i \frac{W}{W_P} (x_i - \bar{x}_P) \hat{\beta}_R \right) \\ &= w_i \left(R_i \frac{N}{W_R} \bar{x}_P \mathbf{Q}_R x_i' (y_i - x_i \hat{\beta}_R) + P_i \frac{N}{W_P} (x_i - \bar{x}_P) \hat{\beta}_R \right) \end{aligned}$$

where $\bar{x}_P$ is the mean vector of the X variables from subpopulation P, and $\hat{\beta}_R$ and $\mathbf{Q}_R$ are from subpopulation R. As above, we can further apply a degrees-of-freedom correction for the number of regression coefficients by multiplying by $\sqrt{(N-1)/(N-k)}$.

▷ Example

I fit a regression model to the control group (`treat==0`) and then estimate the mean of the out-of-sample predictions from this model in the treatment group (`treat==1`).

```
. regress re78 education re74 [iweight = w] if treat==0
  (output omitted)
. mata:
------------------------------------------------- mata (type end to exit) ------
: b = st_matrix("e(b)")'
: y = st_data(., "re78"); N = rows(y)
: X = st_data(., "education re74"), J(N, 1, 1); k = cols(X)
: w = st_data(., "w"); W = sum(w)
: P = st_data(., "treat"); R = (1:-P)
: XP = select(X, P); XR = select(X, R)
: wP = select(w, P); WP = sum(wP)
: wR = select(w, R); WR = sum(wR)
: xbarP = mean(XP, wP)
: Q = invsym(cross(XR, wR, XR) / WR)
: IF = w :* (R*(N/WR) :* ((y :- X*b) :* X * Q * xbarP') +
>            P*(N/WP) :* ((X :- xbarP) * b)) * sqrt((N-1)/(N-k))
: mean(XP * b, wP), sqrt(variance(IF)/N)
                  1              2
    +-------------------------------+
  1 |  4908.611896   856.5191672    |
    +-------------------------------+

: end
--------------------------------------------------------------------------------
```

The analysis can be replicated by `margins` as follows:

```
. qui regress re78 i.treat##c.(education re74) [pweight = w]

. margins, at(treat==0) subpop(treat) vce(unconditional)

Predictive margins                              Number of obs     =        438
                                                Subpop. no. obs   =        185

Expression   : Linear prediction, predict()
at           : treat           =           0

------------------------------------------------------------------------------
             |             Unconditional
             |     Margin   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   4908.612    859.488     5.71   0.000     3219.313     6597.91
------------------------------------------------------------------------------
```

The results are not identical because margins bases its degrees of freedom on a fully interacted model with $2k$ parameters (fitting a model without interactions to the control group and then making out-of-sample predictions for the treatment group would, in principle, be supported by `margins` through the `noesample` option; however, this only works without `vce(unconditional)`, which would assume the X variables to be fixed). We can exactly replicate `margins`'s standard error by adapting the degrees-of-freedom correction in the influence function:

```
. mata:
------------------------------------------------- mata (type end to exit) ------
: IF = IF / sqrt((N-1)/(N-k)) * sqrt((N-1)/(N-2*k))

: mean(XP * b, wP), sqrt(variance(IF)/N)
                  1              2
    +-------------------------------+
  1 |  4908.611896   859.4880467    |
    +-------------------------------+

: end
--------------------------------------------------------------------------------
```

◁

4.4 Mean difference

Before turning to regression adjustment, I take a brief look at the influence function for a mean difference between two subpopulations. Again, let P_i and R_i be indicators for two (possibly overlapping) subpopulations. We are interested in the influence function for the difference in means

$$\hat{\delta} = \bar{y}_P - \bar{y}_R = \frac{1}{W_P}\sum_{i=1}^{N} P_i w_i y_i - \frac{1}{W_R}\sum_{i=1}^{N} R_i w_i y_i$$

Using the chain rule we can write the influence function as follows:

$$\lambda_i^{\hat{\delta}} = w_i\left(P_i\frac{N}{W_P}(y_i - \bar{y}_P) - R_i\frac{N}{W_R}(y_i - \bar{y}_R)\right)$$

▷ **Example**

```
. mata:
─────────────────────────────────────────────── mata (type end to exit) ──────
: y = st_data(., "re78"); N = rows(y)
: w = st_data(., "w");    W = sum(w)
: P = st_data(., "treat"); R = (1:-P)
: wP = select(w, P); WP = sum(wP)
: wR = select(w, R); WR = sum(wR)
: ybarP = mean(select(y, P), wP)
: ybarR = mean(select(y, R), wR)
: IF = w :* (P*(N/WP) :* (y :- ybarP) - R*(N/WR) :* (y :- ybarR))
: ybarR-ybarP, sqrt(variance(IF)/N)
                 1             2
    +-----------------------------+
  1 |  3657.484528   916.4322928  |
    +-----------------------------+

: end
───────────────────────────────────────────────────────────────────────────────
```

The result is the same as obtained by `mean` followed by `lincom`:

```
. mean re78 [pweight = w], over(treat)
  (output omitted)
. lincom _b[0]-_b[1]
 ( 1)  [re78]0 - [re78]1 = 0
```

Mean	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	3657.485	916.4323	3.99	0.000	1856.322	5458.647

◁

4.5 Regression adjustment

In regression adjustment, a regression model fit to the control group is used to impute potential outcomes without treatment (Y^0) in the treatment group and a regression model fit to the treatment group is used to impute potential outcomes with treatment (Y^1) in the control group. Let D_i, $i = 1, \ldots, N$, be a treatment indicator, such that $D_i = 1$ in the treatment group and $D_i = 0$ in the control group. Furthermore, let $\hat{\beta}_0$ be the regression coefficients from the control group and $\hat{\beta}_1$ be the coefficients from the treatment group; likewise $W_0 = \sum_i (1 - D_i) w_i$ and $W_1 = \sum_i D_i w_i$ are the group sizes (sum of weights). Estimators for the expected values of potential outcomes Y^1 and Y^0 in the treatment group, in the control group, and in the combined group are defined as

$$
\begin{aligned}
\hat\eta_1^1 &= \widehat{E}[Y^1|D=1] = \frac{1}{W_1}\sum_{i=1}^N w_i D_i x_i \hat\beta_1 = \frac{1}{W_1}\sum_{i=1}^N w_i D_i y_i && = \bar y_1 \\
\hat\eta_0^1 &= \widehat{E}[Y^1|D=0] = \frac{1}{W_0}\sum_{i=1}^N w_i (1-D_i) x_i \hat\beta_1 && = \bar x_0 \hat\beta_1 \\
\hat\eta^1 &= \widehat{E}[Y^1] \qquad\quad\;\; = \frac{1}{W}\sum_{i=1}^N w_i x_i \hat\beta_1 && = \bar x \hat\beta_1 \\
\hat\eta_1^0 &= \widehat{E}[Y^0|D=1] = \frac{1}{W_1}\sum_{i=1}^N w_i D_i x_i \hat\beta_0 && = \bar x_1 \hat\beta_0 \\
\hat\eta_0^0 &= \widehat{E}[Y^0|D=0] = \frac{1}{W_0}\sum_{i=1}^N w_i (1-D_i) x_i \hat\beta_0 = \frac{1}{W_0}\sum_{i=1}^N w_i (1-D_i) y_i && = \bar y_0 \\
\hat\eta^0 &= \widehat{E}[Y^0] \qquad\quad\;\; = \frac{1}{W}\sum_{i=1}^N w_i x_i \hat\beta_0 && = \bar x \hat\beta_0
\end{aligned}
$$

where $\bar y_1$ (mean of Y) and $\bar x_1$ (row vector of means of X variables) are from the treatment group and $\bar y_0$ and $\bar x_0$ are from the control group. Using results from previous sections, the influence functions for these quantities can be written as

$$
\begin{aligned}
\lambda_i^{\hat\eta_1^1} &= w_i D_i \frac{N}{W_1}(y_i - \bar y_1) \\
\lambda_i^{\hat\eta_0^1} &= w_i \left(D_i \frac{N}{W_1} \bar x_0 \mathbf{Q}_1 x_i' (y_i - x_i \hat\beta_1) + (1-D_i)\frac{N}{W_0}(x_i - \bar x_0)\hat\beta_1 \right) \\
\lambda_i^{\hat\eta^1} &= w_i \left(D_i \frac{N}{W_1} \bar x \mathbf{Q}_1 x_i' (y_i - x_i \hat\beta_1) + \frac{N}{W}(x_i - \bar x)\hat\beta_1 \right) \\
\lambda_i^{\hat\eta_1^0} &= w_i \left((1-D_i) \frac{N}{W_0} \bar x_1 \mathbf{Q}_0 x_i' (y_i - x_i \hat\beta_0) + D_i\frac{N}{W_1}(x_i - \bar x_1)\hat\beta_0 \right) \\
\lambda_i^{\hat\eta_0^0} &= w_i (1-D_i) \frac{N}{W_0}(y_i - \bar y_0) \\
\lambda_i^{\hat\eta^0} &= w_i \left((1-D_i) \frac{N}{W_0} \bar x \mathbf{Q}_0 x_i' (y_i - x_i \hat\beta_0) + \frac{N}{W}(x_i - \bar x)\hat\beta_0 \right)
\end{aligned}
$$

where $\mathbf{Q}_1$ is from the treatment group and $\mathbf{Q}_0$ is from the control group. As above, it may make sense to additionally apply a degrees-of-freedom correction for the number of regression coefficients by multiplying the influence functions by $\sqrt{(N-1)/(N-k)}$.

Based on the potential outcome means, the estimators for the average treatment effect on the treated (ATT), the average treatment effect on the untreated (ATC), and the average treatment effect (ATE) are defined as:

$$
ATT = \hat\eta_1^1 - \hat\eta_1^0 \qquad\qquad ATC = \hat\eta_0^1 - \hat\eta_0^0 \qquad\qquad ATE = \hat\eta^1 - \hat\eta^0
$$

Influence functions for these statistics can be obtained as follows:

$$\lambda_i^{ATT} = \lambda_i^{\hat{\eta}_1^1} - \lambda_i^{\hat{\eta}_1^0} \qquad \lambda_i^{ATC} = \lambda_i^{\hat{\eta}_0^1} - \lambda_i^{\hat{\eta}_0^0} \qquad \lambda_i^{ATE} = \lambda_i^{\hat{\eta}^1} - \lambda_i^{\hat{\eta}^0}$$

▷ **Example**

Here is an example computing all of the above quantities using weighted data:

```
. regress re78 education re74 if treat==0 [iweight = w]
  (output omitted)
. mata: b0 = st_matrix("e(b)")'
. regress re78 education re74 if treat==1 [iweight = w]
  (output omitted)
. mata:
------------------------------------------------- mata (type end to exit) ------
: b1 = st_matrix("e(b)")'
: y  = st_data(., "re78"); N = rows(y)
: w  = st_data(., "w"); W = sum(w)
: X  = st_data(., "education re74"), J(N, 1, 1); k = cols(X)
: D  = st_data(., "treat")
: w0 = select(w, (1:-D)); w1 = select(w, D)
: X0 = select(X, (1:-D)); X1 = select(X, D)
: W0 = sum(w0); W1 = sum(w1)
: ybar0 = mean(select(y, (1:-D)), w0); ybar1 = mean(select(y, D), w1)
: xbar0 = mean(X0, w0);  xbar1 = mean(X1, w1); xbar = mean(X, w)
: Q0 = invsym(cross(X0, w0, X0) / W0)
: Q1 = invsym(cross(X1, w1, X1) / W1)
: // compute IF for E11, E10, E1, E01, E00, E0
: F0 = (y :- X * b0) :* X * Q0
: F1 = (y :- X * b1) :* X * Q1
: IF = w :* (D*(N/W1) :* (y :- ybar1 ),
>             D*(N/W1) :* (F1 * xbar0') + (1:-D)*(N/W0) :* ((X :- xbar0) * b1),
>             D*(N/W1) :* (F1 * xbar' ) +        (N/W)    *  (X :- xbar)  * b1 ,
>        (1:-D)*(N/W0) :* (F0 * xbar1') +      D*(N/W1) :* ((X :- xbar1) * b0),
>        (1:-D)*(N/W0) :* (y :- ybar0 ),
>        (1:-D)*(N/W0) :* (F0 * xbar' ) +        (N/W)    *  (X :- xbar)  * b0)
: // fillin IF for ATT, ATC, ATE
: IF = IF, IF[,1] - IF[,4], IF[,2] - IF[,5], IF[,3] - IF[,6]
: // confirm that the means of the IFs are zero
: assert(all(mean(IF) :< 1e-10))
: // compute variance-covariance matrix
: V = variance(IF) / N
: end
--------------------------------------------------------------------------------
```

We can now compare the resulting standard errors to the values obtained by `margins`. For the average treatment effect on the treated (ATT) the results are as follows:

```
. mata: sqrt(diagonal(V)[(4,1,7)])'
                 1             2             3
    +-------------------------------------------+
  1 |  854.5569214   585.0051281   1022.799096  |
    +-------------------------------------------+

. regress re78 i.treat##c.(education re74) [pweight = w]
  (output omitted)
. margins, at(treat==(0 1)) subpop(treat) vce(unconditional) post
  (output omitted)
. nlcom (E01:_b[1._at]) (E11:_b[2._at]) (ATT:_b[2._at]-_b[1._at]), noheader
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
E01	4908.612	859.488	5.71	0.000	3224.046	6593.178
E11	6238.712	588.3808	10.60	0.000	5085.507	7391.917
ATT	1330.1	1028.701	1.29	0.196	-686.1167	3346.317

The standard errors from the influence functions are slightly different from the standard errors computed by `margins`. This is because `margins` is based on a model with $2k$ coefficients, whereas Mata's `variance()` function divides the sum of squares by $N-1$. We can make the results comparable by rescaling the influence-function standard errors:

```
. mata: sqrt(diagonal(V)[(4,1,7)] * (N-1)/(N-2*k))'
                 1             2             3
    +-------------------------------------------+
  1 |  859.4880467   588.3808348   1028.701044  |
    +-------------------------------------------+
```

We can also compare our results to the values computed by `teffects ra`:

```
. mata: sqrt(diagonal(V)[(7,4)])'
                 1             2
    +-----------------------------+
  1 |  1022.799096   854.5569214  |
    +-----------------------------+

. teffects ra (re78 education re74) (treat) [pweight = w], nolog atet
Treatment-effects estimation                    Number of obs     =        438
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

re78	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
ATET						
treat						

(1 vs 0)	1330.1	1021.631	1.30	0.193	-672.2593	3332.46
POmean						
treat						
0	4908.612	853.5808	5.75	0.000	3235.624	6581.6

Again, the results are slightly different because the number of estimated parameters is taken into account differently. In particular, `teffects ra` completely ignores this information and returns results that are equivalent to dividing the sum of squares by N:

```
. mata: sqrt(diagonal(V)[(7,4)] * (N-1)/N)'
                 1             2
    +-----------------------------+
  1 |   1021.63085   853.5808423  |
    +-----------------------------+
```

For sake of completeness, here are also the results for the average treatment effect on the untreated (ATC) ...

```
. mata: sqrt(diagonal(V)[(5,2,8)] * (N-1)/(N-2*k))'
                 1             2             3
    +-------------------------------------------+
  1 |  709.4903841   1714.034492   1830.694215  |
    +-------------------------------------------+

. regress re78 i.treat##c.(education re74) [pweight = w]
  (output omitted)
. margins, at(treat==(0 1)) subpop(if treat==0) vce(unconditional) post
  (output omitted)
. nlcom (E00:_b[1._at]) (E10:_b[2._at]) (ATC:_b[2._at]-_b[1._at]), noheader
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
E00	9896.197	709.4904	13.95	0.000	8505.621	11286.77
E10	6926.207	1714.034	4.04	0.000	3566.761	10285.65
ATC	-2969.99	1830.694	-1.62	0.105	-6558.085	618.1047

... and the average treatment effect (ATE):

```
. mata: sqrt(diagonal(V)[(6,3,9)] * (N-1)/(N-2*k))'
                 1             2             3
    +-------------------------------------------+
  1 |  616.0128253   1102.705473   1232.698695  |
    +-------------------------------------------+

. regress re78 i.treat##c.(education re74) [pweight = w]
  (output omitted)
. margins, at(treat==(0 1)) vce(unconditional) post
  (output omitted)
```

```
. nlcom (E0:_b[1._at]) (E1:_b[2._at]) (ATE:_b[2._at]-_b[1._at]), noheader

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          E0 |   7778.894   616.0128    12.63   0.000     6571.531    8986.257
          E1 |   6634.355   1102.705     6.02   0.000     4473.092    8795.618
         ATE |  -1144.539   1232.699    -0.93   0.353    -3560.584    1271.506
------------------------------------------------------------------------------

. mata: sqrt(diagonal(V)[(9,6)] * (N-1)/N)'
                 1             2
    +-----------------------------+
  1 |  1224.226438   611.7790101  |
    +-----------------------------+

. teffects ra (re78 education re74) (treat) [pweight = w], nolog

Treatment-effects estimation                    Number of obs     =        438
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
------------------------------------------------------------------------------
             |               Robust
        re78 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ATE          |
       treat |
   (1 vs 0)  |  -1144.539   1224.226    -0.93   0.350    -3543.979    1254.901
-------------+----------------------------------------------------------------
POmean       |
       treat |
          0  |   7778.894    611.779    12.72   0.000      6579.83    8977.959
------------------------------------------------------------------------------
```

◁

4.6 Reweighted regression adjustment

Assume that there are additional weights that are used to estimate the regression adjustment models based on which the potential outcomes are imputed, but that are ignored when taking averages. Let $\tilde{w}_i$, $i = 1, \dots, N$, be these weights. The corresponding coefficients are denoted by $\tilde{\beta}_0$ and $\tilde{\beta}_1$. Just like the sampling weights w_i, I will assume these weights to be fixed. The quantities of interest are then defined as

$$\hat{\eta}_1^1 = \widehat{E}[Y^1|D=1] = \frac{1}{W_1}\sum_{i=1}^N w_i D_i y_i = \bar{y}_1$$

$$\hat{\eta}_0^1 = \widehat{E}[Y^1|D=0] = \frac{1}{W_0}\sum_{i=1}^N w_i (1-D_i) x_i \tilde{\beta}_1 = \bar{x}_0\tilde{\beta}_1$$

$$\hat{\eta}^1 = \widehat{E}[Y^1] = \frac{\hat{p}}{W_1}\sum_{i=1}^N w_i D_i y_i + \frac{1-\hat{p}}{W_0}\sum_{i=1}^N w_i (1-D_i) x_i \tilde{\beta}_1 = \hat{p}\bar{y}_1 + (1-\hat{p})\bar{x}_0\tilde{\beta}_1$$

$$\hat\eta_1^0 = \widehat{E}[Y^0|D=1] = \frac{1}{W_1}\sum_{i=1}^N w_i D_i x_i \tilde\beta_0 \qquad = \bar x_1 \tilde\beta_0$$

$$\hat\eta_0^0 = \widehat{E}[Y^0|D=0] = \frac{1}{W_0}\sum_{i=1}^N w_i (1-D_i) y_i \qquad = \bar y_0$$

$$\hat\eta^0 = \widehat{E}[Y^0] \qquad = \frac{\hat p}{W_1}\sum_{i=1}^N w_i D_i x_i \tilde\beta_0 + \frac{1-\hat p}{W_0}\sum_{i=1}^N w_i (1-D_i) y_i = \hat p \bar x_1 \tilde\beta_0 + (1-\hat p)\bar y_0$$

where $\hat p = W_1/W$ is the relative size of the treatment group. The influence functions can be written as follows:

$$\begin{aligned}
\lambda_i^{\hat\eta_1^1} &= w_i D_i \frac{N}{W_1}(y_i - \bar y_1) \\
\lambda_i^{\hat\eta_0^1} &= \tilde w_i D_i \frac{N}{\widetilde W_1} \bar x_0 \tilde{\mathbf{Q}}_1 x_i'(y_i - x_i\tilde\beta_1) + w_i(1-D_i)\frac{N}{W_0}(x_i - \bar x_0)\tilde\beta_1 \\
\lambda_i^{\hat\eta^1} &= \tilde w_i D_i \frac{N}{\widetilde W_1}(1-\hat p)\bar x_0 \tilde{\mathbf{Q}}_1 x_i'(y_i - x_i\tilde\beta_1) + w_i D_i \frac{N}{W_1}\hat p(y_i - \bar y_1) \\
&\quad + w_i(1-D_i)\frac{N}{W_0}(1-\hat p)(x_i - \bar x_0)\tilde\beta_1 + w_i\frac{N}{W}(\bar y_1 - \bar x_0\tilde\beta_1)(D_i - \hat p) \\
\lambda_i^{\hat\eta_1^0} &= \tilde w_i(1-D_i)\frac{N}{\widetilde W_0}\bar x_1 \tilde{\mathbf{Q}}_0 x_i'(y_i - x_i\tilde\beta_0) + w_i D_i\frac{N}{W_1}(x_i - \bar x_1)\tilde\beta_0 \\
\lambda_i^{\hat\eta_0^0} &= w_i(1-D_i)\frac{N}{W_0}(y_i - \bar y_0) \\
\lambda_i^{\hat\eta^0} &= \tilde w_i(1-D_i)\frac{N}{\widetilde W_0}\hat p \bar x_1 \tilde{\mathbf{Q}}_0 x_i'(y_i - x_i\tilde\beta_0) + w_i(1-D_i)\frac{N}{W_0}(1-\hat p)(y_i - \bar y_0) \\
&\quad + w_i D_i \frac{N}{W_1}\hat p(x_i - \bar x_1)\tilde\beta_0 + w_i\frac{N}{W}(\bar x_1\tilde\beta_0 - \bar y_0)(D_i - \hat p)
\end{aligned}$$

If $\tilde w_i = w_i$ for all i, and thus $\tilde\beta = \hat\beta$, $\widetilde W = W$, and $\tilde{\mathbf{Q}} = \mathbf{Q}$, these formulas are equivalent to the formulas for non-reweighted regression adjustment above. As usual, it may make sense to include a $\sqrt{(N-1)/(N-k)}$ degrees-of-freedom correction.

The influence functions for the ATT, the ATC, and the ATE can again be obtained as:

$$\lambda_i^{ATT} = \lambda_i^{\hat\eta_1^1} - \lambda_i^{\hat\eta_1^0} \qquad \lambda_i^{ATC} = \lambda_i^{\hat\eta_0^1} - \lambda_i^{\hat\eta_0^0} \qquad \lambda_i^{ATE} = \lambda_i^{\hat\eta^1} - \lambda_i^{\hat\eta^0}$$

▷ Example

I first generate some propensity-score weights (IPW). The weights are constructed such that the reweighted control group looks approximately like the (non-reweighted) treatment group and vice versa. As before, I also include the sampling weights.

```
. logit treat age education nodegree black hispanic re74 re75 [pw = w]
```

(*output omitted*)

```
. predict ps, pr
. generate ipw = w * cond(treat==1, (1-ps)/ps, ps/(1-ps))
. summarize ipw
```

Variable	Obs	Mean	Std. Dev.	Min	Max
ipw	438	917.0698	2864.617	.0003952	35386.44

Here is how the propensity-score weights adjust the control group to the treatment group:

```
. tabstat age education nodegree black hispanic re74 re75 [aw=w] if treat==1
```

stats	age	educat~n	nodegree	black	hispanic	re74	re75
mean	25.64152	10.37802	.704659	.8521518	.0605938	2038.417	1484.186

```
. tabstat age education nodegree black hispanic re74 re75 [aw=ipw] if treat==0
```

stats	age	educat~n	nodegree	black	hispanic	re74	re75
mean	25.44958	10.59501	.6636382	.8872232	.0308121	3010.308	2086.715

And here the reverse:

```
. tabstat age education nodegree black hispanic re74 re75 [aw=w] if treat==0
```

stats	age	educat~n	nodegree	black	hispanic	re74	re75
mean	36.07722	10.77328	.4826845	.3928743	.0665231	10867.49	7407.348

```
. tabstat age education nodegree black hispanic re74 re75 [aw=ipw] if treat==1
```

stats	age	educat~n	nodegree	black	hispanic	re74	re75
mean	30.80647	11.28833	.4670723	.7605381	.0218143	12643	8743.437

The weights make the groups quite similar in terms of their covariate means, but the match is far from perfect, especially when trying to adjust the treatment group to the control group. Instead of computing raw IPW estimates based on these propensity-score weights, the idea now is to refine the estimates using regression adjustment as outlined above. The computation of the influence functions is as follows:

```
. regress re78 age education nodegree black hispanic re74 re75 ///
>     if treat==0 [iweight = ipw]
```

(*output omitted*)

```
. mata: b0 = st_matrix("e(b)")'
. regress re78 age education nodegree black hispanic re74 re75 ///
>     if treat==1 [iweight = ipw]
```

(*output omitted*)

```
. mata:
------------------------------------------------- mata (type end to exit) -------
: b1   = st_matrix("e(b)")'

: y    = st_data(., "re78"); N = rows(y)

: w    = st_data(., "w"); W = sum(w)

: ipw  = st_data(., "ipw")

: X    = st_data(., "age education nodegree black hispanic re74 re75"),
>        J(N, 1, 1)

: k    = cols(X)

: D    = st_data(., "treat")

: w0   = select(w, (1:-D)); w1 = select(w, D)

: W0   = sum(w0); W1 = sum(w1)

: ipw0 = select(ipw, (1:-D)); ipw1 = select(ipw, D)

: IPW0 = sum(ipw0); IPW1 = sum(ipw1)

: X0  = select(X, (1:-D)); X1 = select(X, D)

: ybar0 = mean(select(y, (1:-D)), w0); ybar1 = mean(select(y, D), w1)

: xbar0 = mean(X0, w0);  xbar1 = mean(X1, w1); xbar = mean(X, w)

: Q0 = invsym(cross(X0, ipw0, X0) / IPW0)

: Q1 = invsym(cross(X1, ipw1, X1) / IPW1)

: p = W1/W

: // compute estimates
: E = ybar1, xbar0*b1, p*ybar1 + (1-p)*xbar0*b1,
>     xbar1*b0, ybar0, p*xbar1*b0 + (1-p)*ybar0

: E = E, E[1]-E[4], E[2]-E[5], E[3]-E[6]

: // compute IF for E11, E10, E1, E01, E00, E0
: F0 = (y :- X * b0) :* X * Q0

: F1 = (y :- X * b1) :* X * Q1

: IF = ( w :* D*(N/W1) :* (y :- ybar1 ),
>       ipw :* D*(N/IPW1) :* (F1 * xbar0')
>               +  w:* (1:-D)*(N/W0) :* ((X :- xbar0) * b1),
>       ipw :* D*(N/IPW1) * (1-p) :* (F1 * xbar0')
>               +  w :* (D*(N/W1) * p :* (y :- ybar1)
>                   +  (1:-D)*(N/W0) * (1-p) :* ((X :- xbar0) * b1)
>                   +  (N/W) * (ybar1 :- xbar0*b1) :* (D :- p)),
>       ipw :* (1:-D)*(N/IPW0) :* (F0 * xbar1')
>               +  w :* D*(N/W1) :* ((X :- xbar1) * b0),
>         w :* (1:-D)*(N/W0) :* (y :- ybar0 ),
>       ipw :* (1:-D)*(N/IPW0) * p :* (F0 * xbar1')
>               +  w :* ((1:-D)*(N/W0) * (1-p) :* (y :- ybar0)
>                   +  D*(N/W1) * p :* ((X :- xbar1) * b0)
>                   +  (N/W) * (xbar1*b0 :- ybar0) :* (D :- p)))

: // fillin IF for ATT, ATC, ATE
: IF = IF, IF[,1] - IF[,4], IF[,2] - IF[,5], IF[,3] - IF[,6]

: // confirm that the means of the IFs are zero
```

```
: assert(all(mean(IF) :< 1e-10))

: // compute variance-covariance matrix
: V = variance(IF) / N

: end
```

Again, we can compare the results with what is computed by `margins`. For the ATT this is as follows:

```
.  mata: E[(4,1,7)]' , sqrt(diagonal(V)[(4,1,7)] * (N-1)/(N-2*k))
                 1             2

  1    3736.776151   714.9015005
  2    6238.712211   595.3113546
  3     2501.93606   918.3565952

. regress re78 i.treat##c.(age education nodegree black hispanic re74 re75) ///
>     [pweight = cond(treat==1, w, ipw)]
  (output omitted)
. margins, at(treat==(0 1)) subpop(treat) vce(unconditional) post
  (output omitted)
. nlcom (E01:_b[1._at]) (E11:_b[2._at]) (ATT:_b[2._at]-_b[1._at]), noheader
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
E01	3736.776	714.9015	5.23	0.000	2335.595	5137.957
E11	6238.712	595.3114	10.48	0.000	5071.923	7405.501
ATT	2501.936	918.3566	2.72	0.006	701.9902	4301.882

We see that results are identical. For the ATC we get:

```
.  mata: E[(5,2,8)]' , sqrt(diagonal(V)[(5,2,8)] * (N-1)/(N-2*k))
                 1             2

  1    9896.196739   717.8474495
  2    10093.49645   1801.711451
  3    197.2997068   1846.070032

. regress re78 i.treat##c.(age education nodegree black hispanic re74 re75) ///
>     [pweight = cond(treat==1, ipw, w)]
  (output omitted)
. margins, at(treat==(0 1)) subpop(if treat==0) vce(unconditional) post
  (output omitted)
. nlcom (E00:_b[1._at]) (E10:_b[2._at]) (ATC:_b[2._at]-_b[1._at]), noheader
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
E00	9896.197	717.8474	13.79	0.000	8489.242	11303.15

```
         E10 |    10093.5   1801.711     5.60   0.000     6562.207    13624.79
         ATC |   197.2997    1846.07     0.11   0.915    -3420.931     3815.53
-------------------------------------------------------------------------------
```

I did not find a way to produce an estimate of the *ATE* using `margins`. Anyhow, here are the results from the influence-function approach:

```
. mata: R = E[(6,3,9)]', sqrt(diagonal(V)[(6,3,9)] * (N-1)/(N-2*k))

. mata: ("E0", "E1", "ATE")', strofreal(R)
               1          2          3
    +------------------------------------+
  1 |         E0   7281.433   578.8638   |
  2 |         E1   8457.084   1102.122   |
  3 |        ATE   1175.651   1188.438   |
    +------------------------------------+
```

◁

4.6.1 Comparison to `teffetcs ipwra`

The above analysis could be replicated by `teffects ipwra`. Unfortunately, however, the command does not converge in this example. I therefore make a comparison to `teffects ipwra` using a different example.

▷ **Example**

The following example uses the `cattaneo2` dataset from the Stata manual to fit an inverse-probability-weighted regression adjustment estimator. I first generate some sampling weights, just for illustration.

```
. webuse cattaneo2, clear
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

. set seed 9879876

. generate double w = rnormal(1000, 200)
```

I now estimate the inverse-probability weights based on a `probit` model, estimate the regression equations, and calculate the influence functions (*ATT* and *ATE* only):

```
. probit mbsmoke mmarried mage fbaby medu [pweight = w]
  (output omitted)

. predict ps, pr

. generate ipw = w * cond(mbsmoke==1, (1-ps)/ps, ps/(1-ps))

. regress bweight prenatal1 mmarried mage fbaby if mbsmoke==0 [iweight = ipw]
  (output omitted)

. mata: b0 = st_matrix("e(b)")'

. regress bweight prenatal1 mmarried mage fbaby if mbsmoke==1 [iweight = ipw]
  (output omitted)
```

```
. mata:
------------------------------------------------- mata (type end to exit) ------
: b1   = st_matrix("e(b)")'
: y    = st_data(., "bweight"); N = rows(y)
: w    = st_data(., "w"); W = sum(w)
: ipw  = st_data(., "ipw")
: X    = st_data(., "prenatal1 mmarried mage fbaby"), J(N, 1, 1); k = cols(X)
: D    = st_data(., "mbsmoke")
: w0   = select(w, (1:-D)); w1 = select(w, D)
: W0   = sum(w0); W1 = sum(w1)
: ipw0 = select(ipw, (1:-D)); ipw1 = select(ipw, D)
: IPW0 = sum(ipw0); IPW1 = sum(ipw1)
: X0  = select(X, (1:-D)); X1 = select(X, D)
: ybar0 = mean(select(y, (1:-D)), w0); ybar1 = mean(select(y, D), w1)
: xbar0 = mean(X0, w0);  xbar1 = mean(X1, w1); xbar = mean(X, w)
: Q0 = invsym(cross(X0, ipw0, X0) / IPW0)
: Q1 = invsym(cross(X1, ipw1, X1) / IPW1)
: p = W1/W
: // compute estimates
: E = ybar1, p*ybar1 + (1-p)*xbar0*b1, xbar1*b0, p*xbar1*b0 + (1-p)*ybar0
: E = E, E[1]-E[3], E[2]-E[4]
: // compute IF for E11, E10, E1, E01, E00, E0
: F0 = (y :- X * b0) :* X * Q0
: F1 = (y :- X * b1) :* X * Q1
: IF = ( w :* D*(N/W1) :* (y :- ybar1 ),
>      ipw :* D*(N/IPW1) * (1-p) :* (F1 * xbar0')
>              +  w :* (D*(N/W1) * p :* (y :- ybar1)
>                  +  (1:-D)*(N/W0) * (1-p) :* ((X :- xbar0) * b1)
>                  +  (N/W) * (ybar1 :- xbar0*b1) :* (D :- p)),
>      ipw :* (1:-D)*(N/IPW0) :* (F0 * xbar1')
>              +  w :* D*(N/W1) :* ((X :- xbar1) * b0),
>      ipw :* (1:-D)*(N/IPW0) * p :* (F0 * xbar1')
>              +  w :* ((1:-D)*(N/W0) * (1-p) :* (y :- ybar0)
>                  +  D*(N/W1) * p :* ((X :- xbar1) * b0)
>                  +  (N/W) * (xbar1*b0 :- ybar0) :* (D :- p)))
: // fillin IF for ATT and ATE
: IF = IF, IF[,1] - IF[,3], IF[,2] - IF[,4]
: // confirm that the means of the IFs are zero
: assert(all(mean(IF) :< 1e-10))
: // compute variance-covariance matrix
: V = variance(IF) / N
: end
--------------------------------------------------------------------------------
```

Now the results can be compare to `teffects ipwra`. First the ATT:

```
. mata: R = E[(3,1,5)]', sqrt(diagonal(V)[(3,1,5)] * (N-1)/(N))

. mata: ("E01", "E11", "ATT")', strofreal(R)
              1           2           3
    +-------------------------------------+
  1 |       E01    3349.252    14.15778   |
  2 |       E11     3135.68    19.75995   |
  3 |       ATT   -213.5718    24.20196   |
    +-------------------------------------+

. teffects ipwra (bweight prenatal1 mmarried mage fbaby) ///
>                (mbsmoke mmarried mage fbaby medu, probit) ///
>                [pweight = w], atet nolog nofvlabel

Treatment-effects estimation                    Number of obs     =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit
------------------------------------------------------------------------------
             |               Robust
     bweight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ATET         |
     mbsmoke |
   (1 vs 0)  |  -213.5718   24.18786    -8.83   0.000    -260.9791   -166.1644
-------------+----------------------------------------------------------------
POmean       |
     mbsmoke |
          0  |   3349.252   14.14895   236.71   0.000      3321.52    3376.983
------------------------------------------------------------------------------
```

As expected, the point estimates are identical. Furthermore, although, although not being exactly the same, also the standard errors are very close. This is somewhat surprising because `teffects ipwra` estimates the standard errors based on a joint GMM model that includes both the propensity-score model and the regression-adjustment equations, whereas my implementation of the influence-function approach assumes the propensity-score weights to be fixed. For the *ATE* the results are as follows:

```
. mata: R = E[(4,2,6)]', sqrt(diagonal(V)[(4,2,6)] * (N-1)/(N))

. mata: ("E0", "E1", "ATE")', strofreal(R)
              1           2           3
    +-------------------------------------+
  1 |        E0    3397.716    9.827174   |
  2 |        E1    3165.207    24.38111   |
  3 |       ATE   -232.5094    26.27203   |
    +-------------------------------------+

. teffects ipwra (bweight prenatal1 mmarried mage fbaby) ///
>                (mbsmoke mmarried mage fbaby medu, probit) ///
>                [pweight = w], nolog nofvlabel

Treatment-effects estimation                    Number of obs     =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit
------------------------------------------------------------------------------
             |               Robust
```

bweight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
ATE						
mbsmoke						
(1 vs 0)	-232.5311	26.14074	-8.90	0.000	-283.766	-181.2962
POmean						
mbsmoke						
0	3397.687	9.826318	345.77	0.000	3378.428	3416.947

In this case also the point estimates are not exactly the same; it appears that `teffects ipwra` uses a slightly different method to determine the ATE (the definition I use is such that $ATE = \hat{p}ATC + (1 - \hat{p})ATC$, where $\hat{p}$ is the proportion of the treatment group in the sample; for estimates from `teffects ipwra` this relation does not exactly hold, at least not in the above example). Nonetheless, the standard errors are again very close.

◁

References

Dehejia, Rajeev H. and Sadek Wahba. 1999. "Causal Effects in Non-Experimental Studies: Reevaluating the Evaluation of Training Programs." *Journal of the American Statistical Association* 94:1053–1062.

jayk. 2015. "Influence functions and OLS." Cross Validated. Available from https://stats.stackexchange.com/q/139506 (version: 2015-02-26).

Kahn, Jay. 2015. "Influence Functions for Fun and Profit." Ross School of Business, University of Michigan. Available from http://j-kahn.com/files/influencefunctions.pdf (version: July 10, 2015).

LaLonde, Robert J. 1986. "Evaluating the Econometric Evaluations of Training Programs with Experimental Data." *The American Economic Review* 76:604–620.