# Documentation of MATLAB code generated in connection with "Inverse probability tilting for moment condition models with missing data"

*by Bryan Graham, Cristine Pinto, and Daniel Egel*

In connection with our paper we wrote several MATLAB programs which we document here. The first file `NLSY79_Application_Base_Summer2010.m` is the base batch file which reproduces the empirical results in Section 4 of the paper. See

`IPTPaperReplicationInstructions.pdf`

for details on replicating our results. Here we describe the functions this file calls. These functions may be helpful to researchers undertaking original research. Please feel free to modify the code for your own use. We do ask that you acknowledge our work by citing the original paper and explicitly mentioning the use of our code. This code is provided "as is", without warranty explicit or otherwise. Beyond these instructions we are unable to assist with its use. If you do find errors we would welcome you telling us. Please e-mail me at `bryan.graham@nyu.edu`.

**Propensity score estimation files** We generated two functions for propensity score estimation. The first `LOGIT.m` computes conditional maximum likelihood estimates of a binary choice logit model. Our implementation is standard. The second file `IPT_LOGIT.m` estimates a binary choice logit model using the method-of-moments approach introduced in our paper. The algorithm we implement is described in detail in the computational appendix of the paper. This function may be used to estimate the propensity score to impose 'exact balance' in the sense described in the paper. Inverse probability weights constructed using the resulting fitted propensity score can then be used in conjunction with any method of moments or M-estimation software that accepts user-defined weights to implement the second stage of IPT estimation (which is model-specific). A nonparametric bootstrap procedure, which repeats each of the two stages of estimation for each bootstrap draw, can be used to construct asymptotically valid standard errors and confidence intervals. Alternatively, analytic standard errors may be computed as described in the Appendix to the paper.

**IPW, AIPW, and IPT Linear Predictor (LP) estimation with a subset of regressors missing at random** In the empirical application included in the paper the object of interest is a vector of linear predictor (LP) coefficients where one of the regressors is missing at random. The functions `IPW_MIS_REG.m`, `AIPW_MIS_REG.m`, and `IPT_MIS_REG.m` respectively compute IPW, AIPW and IPT estimates of LP coefficients when a subset of regressors are missing (i.e., with item non-response).

**Utility files** The linear predictor estimation functions allow for the computation of standard errors appropriate for settings where the data have a natural groups structure (i.e., they compute 'clustered' standard errors). In order to do this the user must pass in a $G \times 1$ vector of cluster counts.

$G$ is the total number of clusters and the vector of counts includes the number of units in each cluster. The data are assumed to be sorted by cluster/group identification number. The function `[NG] = CountGroupMembers(G_id)` creates the required vector of cluster counts. Let `G_id` be an $N \times 1$ vector with each sampled unit's group identification number, then `NG` is the required vector.

```
[delta_ML VCOV_delta_ML] = LOGIT(D,X,silent,sw);
```

This function computes the ML estimates of the logit binary choice model: $\Pr\left(D=1|\,X=x\right) = \exp\left(x'\delta_0\right)/\left[1+\exp\left(x'\delta_0\right)\right]$

*INPUTS*

`D` : $N \times 1$ vector of binary outcomes

`X` : $X$ is an $N \times K$ matrix of covariates (without a constant)

`silent` : when `silent` $= 1$ optimization output is suppressed and optimization is by Fisher-scoring with lower tolerances. Otherwise optimization starts with a few Fisher-scoring steps and then switches to a quasi-Newton search.

`sw` : $N \times 1$ vector of known sampling weights (assumed to have mean one)

*OUTPUTS*

`gamma_ML` : ML estimates of logit coefficients

`VCOV_delta_ML` : large sample covariance of estimates (= inverse Information)

*FUNCTIONS CALLED*: `LOGIT_LOGL()`

```
[delta_IPT VCOV_delta_IPT HESS_IPT] = IPT_LOGIT(D, h_X, silent, sw);
```

This function computes IPT estimates of the logit binary choice model: $\Pr\left(D=1|\,X=x\right) = \exp\left(x'\delta_0\right)/\left[1+\exp\left(x'\delta_0\right)\right]$. The algorithm is as described in the Appendix of Graham, Pinto and Egel (2010).

*INPUTS*

`D` : $N \times 1$ vector of binary outcomes

h_`X` : $h(X)$ is an $N \times M$ matrix of covariates (assumes a constant is excluded). Note: $t(X)$ in the paper equals $(1, h(X)')'$.

`silent` : when `silent` $= 1$ optimization output is suppressed and optimization is by Newton-Raphson with lower tolerances. Otherwise optimization starts with a few Newton-Raphson steps and then switches to a quasi-Newton search.

`sw` : $N \times 1$ vector of known sampling weights (assumed to have mean one)

*OUTPUTS*

`gamma_IPT` : IPT estimates of logit coefficients

`VCOV_delta_IPT` : large sample covariance of estimates

`HESS_IPT` : Hessian of IPT criterion function (this corresponds to the Jacobian of the IPT sample moments and hence can be useful for constructing standard errors when `IPT_LOGIT()` is the first step of a two-step procedure.

*FUNCTIONS CALLED*: `IPT_LOGIT_CRIT()` and `IPT_LOGIT_PHI()`

```
[gamma_IPW, VCOV_gamma_IPW, delta_ML, VCOV_delta_ML, w_IPW]
                                    = IPW_MIS_REG(D,DY,X1,X2,h_X,NG,sw)
```

This function computes the IPW parameter estimates associated with the best linear predictor of `X2` given `X1` and `Y`:

$$\mathbb{E}^*\left[X2|\,X1,Y\right]=X1'\gamma_1+Y'\gamma_2$$

with $Y$ MAR. This corresponds to Example 2 in paper. The estimator is as described in, for example, Wooldridge (*Journal of Econometrics*, 2007). The propensity score is assumed to take a logit form with an index that is linear in the function of always observed variables `h_X`.The propensity score parameter is estimated by conditional maximum likelihood.

*INPUTS*

`D` : $N\times 1$ vector with $i^{th}$ element equal to 1 if $i^{th}$ unit's $Y$ variable is observed and zero otherwise

`DY` : $D\cdot Y$, with $Y$ the $N\times\ K_1$ matrix of regressors missing at random (note this implies that missing values of the regressors should be set equal to zero)

`X1` : $N\times\ K_2$ matrix of always-observed regressors (should include constant). NOTE: $K=K_1+K_2=\dim(\gamma)$

`X2` : $N\times 1$ vector of always-observed scalar-valued outcome

`h_X` : h(X) $N\times M$ function for propensity score computation (does not include a constant). Note that $t(X)=(1,h(X)')'$. Should include functions of proxies or surrogates for $Y$.

`NG` : $G\times 1$ vector with $g^{th}$ row equal to the number of units in the $g^{th}$ cluster. Data are assumed to be ordered by cluster/group identification number. If the data are iid pass in a $N\times 1$ vector of ones for this parameter.

`sw` : $N\times 1$ vector of known sampling weights

*OUTPUTS*

`gamma_IPW` : IPW estimate of $\gamma$

`VCOV_gamma_IPW` : estimated large sample covariance of $\widehat{\gamma}_{IPW}$ (Huber/Clustered estimate)

`delta_ML` : CMLE estimates of propensity score parameter

`VCOV_delta_ML` : estimated large sample covariance of p-score parameter (Huber/Clustered estimate)

`w_IPW` : IPW probability weights / distribution function estimate (as described in Section 3 of the paper)

*FUNCTIONS CALLED*: `LOGIT()` and `LOGIT_LOGL()`

```
[gamma_AIPW, VCOV_gamma_AIPW, delta_ML, VCOV_delta_ML, w_AIPW, VERSION]
                          = AIPW_MIS_REG(D,DY,X1,X2,h_X,NG,sw)
```

This function computes the AIPW parameter estimates associated with the best linear predictor of `X2` given `X1` and `Y`:

$$\mathbb{E}^*\left[X2|\,X1,Y\right] = X1'\gamma_1 + Y'\gamma_2$$

with $Y$ MAR. This corresponds to Example 2 in paper. The estimator is as described in Section 3 of the paper with all four variants listed in Table 2 available. The propensity score is assumed to take a logit form with an index that is linear in the function of always observed variables `h_X`.The propensity score parameter is estimated by conditional maximum likelihood.

*INPUTS*

`D` : $N \times 1$ vector with $i^{th}$ element equal to 1 if $i^{th}$ unit's $Y$ variable is observed and zero otherwise

`DY` : $D \cdot Y$, with $Y$ the $N \times K_1$ matrix of regressors missing at random (note this implies that missing values of the regressors should be set equal to zero)

`X1` : $N \times K_2$ matrix of always-observed regressors (should include constant). NOTE: $K = K_1 + K_2 = \dim(\gamma)$

`X2` : $N \times 1$ vector of always-observed scalar-valued outcome

`h_X` : h(X) $N \times M$ function for propensity score computation (does not include a constant). Note that $t(X) = (1, h(X)')'$. Should include functions of proxies or surrogates for $Y$.

`NG` : $G \times 1$ vector with $g^{th}$ row equal to the number of units in the $g^{th}$ cluster. Data are assumed to be ordered by cluster/group identification number. If the data are iid pass in a $N \times 1$ vector of ones for this parameter.

`sw` : $N \times 1$ vector of known sampling weights

`VERSION` : 1 = 'Newey (1994)', 2 = 'Hirano and Imbens (2001)/Wooldridge (2007)', 3 = 'Cao, Tsiatis and Davidian (2009)', otherwise = 'Robins, Rotnitzky and Zhao (1994)'. Refers to estimators listed in Table 2 of the paper.

*OUTPUTS*

`gamma_AIPW` : AIPW estimate of $\gamma$

`VCOV_gamma_AIPW` : estimated large sample covariance of $\widehat{\gamma}_{AIPW}$ (Huber/Clustered estimate)

`delta_ML` : CMLE estimates of propensity score parameter

`VCOV_delta_ML` : estimated large sample covariance of p-score parameter (Huber/Clustered estimate)

`w_AIPW` : AIPW probability weights / distribution function estimate (as described in Section 3 of the paper)

*FUNCTIONS CALLED*: `LOGIT()` and `LOGIT_LOGL()`

```
[gamma_IPT, VCOV_gamma_IPT, delta_IPT, VCOV_delta_IPT, w_IPT]
                                  = IPT_MIS_REG(D,DY,X1,X2,h_X,NG,sw)
```

This function computes the IPT parameter estimates associated with the best linear predictor of `X2` given `X1` and `Y`:

$$\mathbb{E}^*\left[X2|\, X1, Y\right] = X1'\gamma_1 + Y'\gamma_2$$

with $Y$ MAR. This corresponds to Example 2 in paper. The estimator is as described in Section 2 of the paper. The propensity score is assumed to take a logit form with an index that is linear in the function of always observed variables `h_X`.The propensity score parameter is estimated by the method of moments procedure described in the paper and the computational appendix.

*INPUTS*

`D` : $N\times 1$ vector with $i^{th}$ element equal to 1 if $i^{th}$ unit's $Y$ variable is observed and zero otherwise

`DY` : $D \cdot Y$, with $Y$ the $N\times\ K_1$ matrix of regressors missing at random (note this implies that missing values of the regressors should be set equal to zero)

`X1` : $N\times\ K_2$ matrix of always-observed regressors (should include constant). NOTE: $K = K_1 + K_2 = \dim(\gamma)$

`X2` : $N \times 1$ vector of always-observed scalar-valued outcome

`h_X` : h(X) $N \times M$ function for propensity score computation (does not include a constant). Note that $t(X) = (1, h(X)')'$. Should include functions of proxies or surrogates for $Y$.

`NG` : $G\times 1$ vector with $g^{th}$ row equal to the number of units in the $g^{th}$ cluster. Data are assumed to be ordered by cluster/group identification number. If the data are iid pass in a $N \times 1$ vector of ones for this parameter.

`sw` : $N \times 1$ vector of known sampling weights

*OUTPUTS*

`gamma_IPT` : IPT estimate of $\gamma$

`VCOV_gamma_IPT` : estimated large sample covariance of $\widehat{\gamma}_{IPT}$ (Huber/Clustered estimate)

`delta_IPT` : IPT estimates of propensity score parameter

`VCOV_delta_IPT` : estimated large sample covariance of p-score parameter (Huber/Clustered estimate)

`w_IPT` : IPT probability weights / distribution function estimate (as described in Sections 2 and 3 of the paper)

*FUNCTIONS CALLED*: `IPT_LOGIT()`, `IPT_LOGIT_CRIT()` and `IPT_LOGIT_PHI()`