Econ 675: HW 4

Erin Markiewitz

November 12, 2018

Contents

1 Estimating Equations **3**
1.1 3
1.2 5
1.3 6

2 **8**

3 **11**
3.1 11
3.2 12

4 Code Appendix **13**

1 Estimating Equations

This question considers identification under selection-on-observables for the generaic class of parameters

$$\theta_t(g) = \mathbb{E}[g(Y_i(t))],\ g \in G,\ t \in T$$

where G denotes a class of functions (e.g. $G = \{\mathbf{1}(. \leq y) : y \in \mathbb{R})\}$). Define the regression functions:

$$p_t(\mathbf{X_i} = \P[T_i = t| \mathbf{X_i}]), e_t(g; \mathbf{X_i}) = \mathbb{E}[g(Y_i(t))|\mathbf{X_i}] == \mathbb{E}[g(Y_i(t))|\mathbf{X_i}, T_i = 1],\ g \in G,\ t \in T$$

Assume Ignorability: $Y_i(t) \perp D_i(t)|\mathbf{X_i}$ and $0 < x < p_t(\mathbf{X}_i)$, for all $t \in T$ and for some fixed positive constant c.

1.1

In the following section, we prove the validity of three moment cofitions for the generic class of parameters. The first moment condition is the Inverse Probabiltiy Weighting (IPW) moment condition.

$$\psi_{IPW,t}(\mathbf{Z}_i; \theta_t(g)) = \frac{D_i(t) \cdot g(Y_i(t))}{p_t(\mathbf{X}_i)} - \theta_t(g)$$

To begin, take the expectation of the moment condition.

$$\mathbb{E}\left[\frac{D_i(t) \cdot g(Y_i(t))}{p_t(\mathbf{X}_i)}\right] - \theta_t(g)$$

By the law of iterative expectations

$$\mathbb{E}\left[\mathbb{E}\left[\frac{D_i(t) \cdot g(Y_i(t))}{p_t(\mathbf{X}_i)}|\mathbf{X}_i\right]\right] - \theta_t(g)$$

$$\mathbb{E}\left[\frac{1}{p_t(\mathbf{X}_i)}\mathbb{E}\left[D_i(t) \cdot g(Y_i(t))|\mathbf{X}_i\right]\right] - \theta_t(g)$$

$$\mathbb{E}\left[\frac{1}{p_t(\mathbf{X}_i)}\mathbb{E}\left[D_i(t)|\mathbf{X}_i\right] \cdot \mathbb{E}\left[g(Y_i(t))|\mathbf{X}_i\right]\right] - \theta_t(g)$$

As $\mathbb{E}\left[D_i(t)|\mathbf{X}_i\right] = \Pr\left[D_i(t) = 1|\mathbf{X}_i\right] = \Pr\left[T_i = t|\mathbf{X}_i\right] = p_t(\mathbf{X}_i)$

$$\mathbb{E}\left[\frac{p_t(\mathbf{X}_i)}{p_t(\mathbf{X}_i)} \cdot \mathbb{E}\left[g(Y_i(t))|\mathbf{X}_i\right]\right] - \theta_t(g)$$

which gives us our result,

$$\mathbb{E}\left[\mathbb{E}\left[g(Y_i(t))|\mathbf{X}_i\right]\right] - \theta_t(g) = \mathbb{E}\left[g(Y_i(t))\right] - \theta_t(g) = \theta_t(g) - \theta_t(g) = 0$$

The second moment condition of this exercise is the Regression Imputation (1) moment condition:

$$\psi_{RI1,t}(\mathbf{Z}_i; \theta_t(g)) = e_t(g; \mathbf{X}_i) - \theta_t(g)$$

Take the expectation

$$\mathbb{E}\left[e_t(g; \mathbf{X}_i)\right] - \theta_t(g)$$

$$\mathbb{E}\left[\mathbb{E}\left[g(Y_i(t))|\mathbf{X}_i\right]\right] - \theta_t(g)$$

$$\mathbb{E}\left[g(Y_i(t))\right] - \theta_t(g) = \theta_t(g) - \theta_t(g) = 0$$

The second moment condition of this exercise is the Regression Imputation (2) moment condition, which includes inverse probabilty weighting:

$$\psi_{RI2,t}(\mathbf{Z}_i; \theta_t(g)) = \frac{D_i(t) \cdot e_t(g; \mathbf{X}_i)}{p_t(\mathbf{X}_i)} - \theta_t(g)$$

Take the expectation

$$\mathbb{E}\left[\frac{D_i(t) \cdot e_t(g; \mathbf{X}_i)}{p_t(\mathbf{X}_i)}\right] - \theta_t(g)$$

iterate the expectation a bit

$$\mathbb{E}\left[\frac{1}{p_t(\mathbf{X}_i)} p_t(\mathbf{X}_i) \cdot \mathbb{E}\left[g(Y_i(t))|\mathbf{X}_i\right]\right] - \theta_t(g)$$

which gives the result

$$\mathbb{E}\left[g(Y_i(t))\right] - \theta_t(g) = \theta_t(g) - \theta_t(g) = 0$$

Last, we consider the doubly robust estimator's moment condition:

$$\psi_{DR,t}(\mathbf{Z}_i; \theta_t(g)) = \frac{D_i(t) \cdot g(Y_i(t))}{p_t(\mathbf{X}_i)} - \theta_t(g) - \frac{e_t(g; \mathbf{X}_i)}{p_t(\mathbf{X}_i)} \cdot (D_i(t) - p_t(\mathbf{X}_i))$$

Take expectations

$$\mathbb{E}\left[\frac{D_i(t) \cdot g(Y_i(t))}{p_t(\mathbf{X}_i)}\right] - \theta_t(g) - \mathbb{E}\left[\frac{e_t(g; \mathbf{X}_i)}{p_t(\mathbf{X}_i)} \cdot (D_i(t) - p_t(\mathbf{X}_i))\right]$$

From previous results the first two terms cancel,

$$-\mathbb{E}\left[\frac{D_i(t) \cdot e_t(g; \mathbf{X}_i)}{p_t(\mathbf{X}_i)}\right] + \mathbb{E}\left[e_t(g; \mathbf{X}_i)\right]$$

The result follows from the law of iterated expectations.

1.2

The IPW plug-in estimator:

$$\hat{\psi}_{IPW,t}(\mathbf{Z}_i; \theta_t(g)) = \frac{1}{n} \sum_{i=1}^{n} \frac{D_i(t) \cdot g(Y_i)}{\hat{p}_t(\mathbf{X}_i)}$$

Where $\hat{p}_t(\mathbf{X}_i)$ is the estimated propensity score from the first-stage regression of the treatment on the covariates.

To write down the RI1 plug-in estimator, start by putting a hat on it:

$$\hat{\psi}_{RI1,t}(\mathbf{Z}_i) = \frac{1}{n} \sum_{i=1}^{n} \hat{e}_t(\mathbf{X}_i)$$

where $\hat{e}_t(X_i) = \mathbb{E}[g(Y_i(t))|\mathbf{X}_i, T_i = t]$, the conditional expectation of the class of regression functions specified by G. We can rewrite the estimator above as:

$$\hat{\psi}_{RI1,t}(\mathbf{Z}_i) = \frac{1}{n} \sum_{i=1}^{n} \frac{D_i(t) \cdot \hat{e}_t(\mathbf{X}_i)}{\hat{p}_t(\mathbf{X}_i)}$$

To write down the RI1 plug-in estimator, just reweight using the estimated propensity score:

$$\hat{\psi}_{RI2,t}(\mathbf{Z}_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{D_i(t)\cdot \hat{e}_t(\mathbf{X}_i)}{\hat{p}_t(\mathbf{X}_i)}$$

And the double robust plug in estimator

$$\hat{\psi}_{DR,t}(\mathbf{Z}_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{D_i(t)\cdot g(Y_i)}{\hat{p}_t(\mathbf{X}_i)} - \frac{1}{n}\sum_{i=1}^{n}\frac{\hat{e}_t(\mathbf{X}_i)}{\hat{p}_t(\mathbf{X}_i)}(D_i(t)-\hat{p}_t(\mathbf{X}_i))$$

The relative performance of the estimators depends on the data generating process. As the IPW and R2 plug in estimators use the estimated propensity score reweight the treatment effects, both estimators will be inconsistent in finite samples when the propensity score is very close to either one or zero. (If you are only estimating the treatement effect on the treated, it is sufficient that the propensity score is not degenerative with respect to 1.) The double robust estimators includes further safeguards against bias induced by misspecification but at the cost of imposing additional specification choices.

Then again, transparency is a key feature of an estimator - especially in policy analysis. Conditioning on covariates allow for specification of the propensity score without prior knowledge of the outcome variable/equation.

1.3

(Just a hunch) The estimating equations in section 1.1 can be used to estimate the variance of the potential outcome variables. First, we specify the function

$$g(x) = (x - \mathbb{E}[x])^2,\ x \in \mathbb{R}$$

and its finite sample analogue

$$\hat{g}(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \frac{1}{n}\sum_{i=1}^{n}x_i)^2,\ x_i \in \mathbf{X} \in \mathbb{R}^n$$

The validity of these moment conditions is established in section 1.1 more generally. Under this specification of g, $\theta_t(g) = \mathbb{V}[Y_i(t)]$ and $e_t(g;\mathbf{X}_i) = \mathbb{V}[Y_i(t)|\mathbf{X}_i]$, the unconditional and conditional variance of the potential outcome of treatement t, respectively. This gives us the moment conditions:

$$\psi_{IPW,t}(\mathbf{Z}_i;\sigma_t^2) = \frac{D_i(t)\cdot(Y_i(t)-\mathbb{E}[Y_i(t)])^2}{p_t(\mathbf{X}_i)} - \sigma_t^2$$

$$\psi_{RI1,t}(\mathbf{Z}_i;\sigma_t^2) = \mathbb{E}[(Y_i(t)-\mathbb{E}[Y_i(t)])^2|\mathbf{X}_i] - \sigma_t^2$$

$$\psi_{RI2,t}(\mathbf{Z}_i;\sigma_t^2) = \frac{D_i(t)\cdot\mathbb{E}[(Y_i(t)-\mathbb{E}[Y_i(t)])^2|\mathbf{X}_i]}{p_t(\mathbf{X}_i)} - \sigma_t^2$$

$$\psi_{DR,t}(\mathbf{Z}_i;\sigma_t^2) = \frac{D_i(t)\cdot(Y_i(t)-\mathbb{E}[Y_i(t)])^2}{p_t(\mathbf{X}_i)} - \sigma_t^2 - \frac{\mathbb{E}[(Y_i(t)-\mathbb{E}[Y_i(t)])^2|\mathbf{X}_i]}{p_t(\mathbf{X}_i)}\cdot(D_i(t)-p_t(\mathbf{X}_i))$$

Now in order to conduct the hypothesis test of $\mathbf{H}_0 : \sigma_t^2 = \sigma^2$ we need to use the finite sample analogue of the g function specified above. The variance of our moment conditions will be estimated using a simple GMM procedure. In this case, we use a two step procedure for the moment conditions that use IPW. In the first step, we estimate the propensity score and drop any observations with propensity scores sufficiently close to zero or one. For a given moment condition $M \in \{IPW, RI1, RI2, DR\}$ and treatment t we define the finite sample analogue as $\hat{\psi}_{M,t}$

So GMM is

$$\hat{\mathbf{\Omega}}_{M,t} = \frac{1}{n}\sum_{i=1}^{n}\hat{\psi}_{M,t}\hat{\psi}'_{M,t}$$

From Theorem 12.7.1 in Hansens's Econometrics text,

$$\hat{\mathbf{V}}_{\psi,M,t} = (\hat{\psi}_{M,t}\hat{\mathbf{\Omega}}^{-1}_{M,t}\hat{\psi}'_{M,t})^{-1}$$

And so in order to conduct the hypothesis test we simply reject the null if $\hat{\sigma}_t^2$ is outside of the following confidence interval:

$$\mathbf{CI}_\alpha(\hat{\sigma}_t^2) = \left[\sigma^2 - \mathbf{\Phi}^{-1}(\frac{\alpha}{2})\sqrt{\frac{\hat{\mathbf{V}}_{\psi,M,t}}{n}}, \sigma^2 + \mathbf{\Phi}^{-1}(\frac{\alpha}{2})\sqrt{\frac{\hat{\mathbf{V}}_{\psi,M,t}}{n}}\right]$$

2

In the following tables I present the ATE and ATT estimated using the Lalonde and PSID data. I am presenting only the Stata results. I have run the other results in R, although I was unable to get certain models to converge or even run in R. In discussion of the results, you see relative significant results with reasonable maginitude that appears to tell the story that there is significant returns to the NSW program. That said, there was significant problems getting the PSID IPW models to converge. This is an interesting situation that I do not have a good sense of what happened. Also matching across nearest neighbor and propensity score returns the same result for all of the specifications, which seems odd in general.

Table 1: Average Treatment Effects

	Experimental Data					PSID Control		
	$\hat{\tau}$	s.e.	C.I.			$\hat{\tau}$	s.e.	C.I.
Mean Diff.								
	1794.3424	670.99654	479.18915	3109.4956	−15204.777	657.07631	−16492.647	−13916.908
OLS								
a	1582.1667	659.2457	290.04507	2874.2882	6302.3954	1212.4566	3925.9805	8678.8104
b	1506.9012	657.31475	218.56428	2795.2381	4699.259	1031.6669	2677.1918	6721.3262
c	1501.3732	662.43532	202.99999	2799.7464	4284.342	1037.3931	2251.0516	6317.6324
Reg. Impute								
a	1462.2693	642.24087	203.4772	2721.0614	−11195.037	1741.3261	−14608.036	−7782.0374
b	1454.1282	643.48562	192.89638	2715.36	−10398.22	3293.3996	−16853.283	−3943.1565
c	1427.5263	642.95044	167.34339	2687.7091	−11920.18	3834.631	−19436.057	−4404.3033
IPW								
a	1537.3978	646.6316	269.99986	2804.7957	−13507.18	2800.1988	−18995.569	−8018.79
b	1469.6152	647.10472	201.28993	2737.9404	−6028.4906	3819.791	−13515.281	1458.2998
c	1468.1014	646.91873	200.14072	2736.0622	−6626.6908	0	−6626.6908	−6626.6908
D. Robust								
a	1537.3978	646.6316	269.99986	2804.7957	−13507.18	2800.1988	−18995.569	−8018.79
b	1469.6152	647.10472	201.28993	2737.9404	−6028.4906	3819.791	−13515.281	1458.2998
c	1468.1014	646.9188	200.1406	2736.0623	−6626.6908	0	−6626.6908	−6626.6908
N1 Match								
a	1829.7958	779.59802	301.78369	3357.8079	−15619.49	1153.3158	−17879.989	−13358.991
b	1875.8281	734.95291	435.32036	3316.3358	−9349.564	3974.7701	−17140.113	−1559.0146
c	1671.7386	726.06099	248.6591	3094.8182	−9561.9662	4033.5414	−17467.707	−1656.225
p Match								
a	10421.27	4318.4516	1957.1047	18885.435	−10421.27	4318.4516	−18885.435	−1957.1047
b	10421.27	4318.4516	1957.1047	18885.435	−10421.27	4318.4516	−18885.435	−1957.1047
c	10421.27	4318.4516	1957.1047	18885.435	−10421.27	4318.4516	−18885.435	−1957.1047

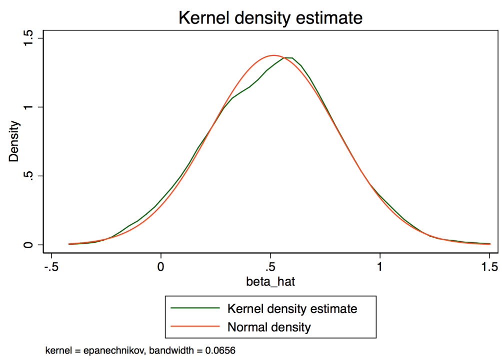
Table 2: Average Treatment Effects on Treated

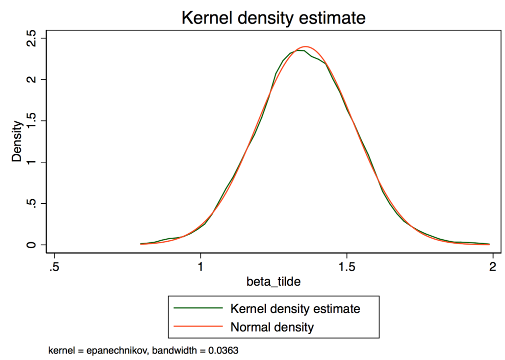
	Experimental Data				**PSID Control**			
	$\hat{\tau}$	s.e.	C.I.		$\hat{\tau}$	s.e.	C.I.	
Mean Diff.								
	1794.3424	670.99654	479.18915	3109.4956	−15204.777	657.07631	−16492.647	−13916.908
OLS								
a	1582.1667	659.2457	290.04507	2874.2882	6302.3954	1212.4566	3925.9805	8678.8104
b	1506.9012	657.31475	218.56428	2795.2381	4699.259	1031.6669	2677.1918	6721.3262
c	1501.3732	662.43532	202.99999	2799.7464	4284.342	1037.3931	2251.0516	6317.6324
Reg. Impute								
a	1726.6021	688.76383	376.62496	3076.5792	−12661.529	1852.7548	−16292.929	−9030.1299
b	1809.6967	693.86739	449.71661	3169.6768	−11537.261	3539.2681	−18474.226	−4600.295
c	1844.6059	694.8217	482.75536	3206.4564	−13218.766	4119.5046	−21292.995	−5144.5369
IPW								
a	1765.8615	698.04009	397.70292	3134.0201	−15249.872	3117.0044	−21359.201	−9140.5436
b	1741.4891	701.89105	365.78265	3117.1956	−7712.6195	4223.102	−15989.899	564.66034
c	1774.86	702.34115	398.27133	3151.4486	−8139.2549	0	−8139.2549	−8139.2549
D. Robust								
a	1765.8615	698.04009	397.70292	3134.0201	−15249.872	3117.0044	−21359.201	−9140.5436
b	1741.4891	701.89105	365.78266	3117.1956	−7712.6195	4223.102	−15989.899	564.66034
c	1774.86	702.34104	398.27155	3151.4484	−8139.2549	0	−8139.2549	−8139.2549
N1 Match								
a	1558.1563	776.73016	35.765204	3080.5474	−16904.15	1217.6947	−19290.831	−14517.468
b	1731.6091	732.36262	296.17836	3167.0398	−10196.024	4254.9459	−18535.718	−1856.3305
c	1137.4252	813.43663	−456.91054	2731.761	−10421.27	4318.4516	−18885.435	−1957.1047
p Match								
a	10421.27	4318.4516	1957.1047	18885.435	−10421.27	4318.4516	−18885.435	−1957.1047
b	10421.27	4318.4516	1957.1047	18885.435	−10421.27	4318.4516	−18885.435	−1957.1047
c	10421.27	4318.4516	1957.1047	18885.435	−10421.27	4318.4516	−18885.435	−1957.1047

3

3.1

While I only include the graphs from the my stata run, I was able to run the exercise in R and provide the table below. The results are quite similar in terms of magnitude. In particular, I found that excluding z_i from the regression led to a biased estimate of the coefficient on x_i which affects the estimate coefficient in the model selection process.





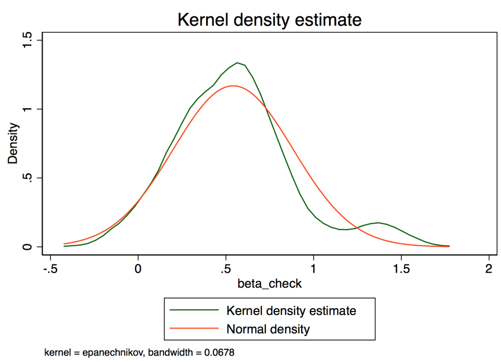


Table 3: Stata Output:

	mean	sd	min	max
$\hat{\beta}$	.5153	.2900	-.3544	1.436
$\check{\beta}$	.5415	.3413	-.3544	1.705
$\tilde{\beta}$	1.358	.1663	.8311	1.950

Table 4: R Output:

	Mean	Min.	Max.
$\hat{\beta}$	0.49	0.49	1.43
$\tilde{\beta}$	1.42	1.43	1.86
$\check{\beta}$	0.49	0.51	1.53

3.2

The model selection procedure drastically reduces the coverage rate of the estimator As the process shifts the location of the point estimate, it significantly reduces the coverage rate of the confidence interval. I think my coverage rate for beta hat is incorrectly estimated, it is far too low.

Stata output:

	Coverage Rate
$\hat{\beta}$	0.201
$\check{\beta}$	0.244
$\tilde{\beta}$	0

R output:

	Coverage Rate
$\hat{\beta}$	0.74
$\tilde{\beta}$	0.00
$\check{\beta}$	0.00

4 Code Appendix

Stata

```
// Erin Markiewitz
// ECON 675 Assignment 4
*******************************************************************************
clear all
set more off, perm
set seed 12345
global dir "/Users/erinmarkiewitz/Dropbox/Phd_Coursework/Econ675/hw4"
global datadir $dir\data
global resdir $dir\results
cd $dir
cap log close
log using $resdir\pset4_stata.smcl, replace


****
* Question 2
****
clear all
set seed 12345
scalar n1 = 185
scalar n0 = 260
scalar n2 = 2490
import delimited using LaLonde_all.csv, clear
gen log_re74 = log(re74 + 1)
gen log_re75 = log(re75 + 1)
gen age2 = age^2
gen age3 = age^3
gen educ2 = educ^2
gen black_u74 = black*u74
gen educ_log_re74 = educ*log_re74

local covars_a = "age educ black hisp  married nodegr log_re74 log_re75"
local covars_b = "age educ black hisp  married nodegr log_re74 log_re75 age2 educ2 u74 u75"
local covars_c = "age educ black hisp  married nodegr log_re74 log_re75 age2 educ2 u74 u75 age3 black_u74 educ_log_re74"


*initialize matricies (0) Experimental data, (2) PSID Control

mat ate0 = J(19,4,.)
mat att0 = J(19,4,.)
mat ate2 = J(19,4,.)
mat att2 = J(19,4,.)

*diff in means
reg re78 treat if treat ==1 | treat == 0, hc2
mat ate0[1,1] = _b[treat]
mat ate0[1,2] = _se[treat]
mat ate0[1,3] = ate0[1,1] - _se[treat] * 1.96
mat ate0[1,4] = ate0[1,1] + _se[treat] * 1.96

mat att0[1,1] = _b[treat]
mat att0[1,2] = _se[treat]
mat att0[1,3] = att0[1,1] - _se[treat] * 1.96
mat att0[1,4] = att0[1,1] + _se[treat] * 1.96

reg re78 treat if treat ==1 | treat == 2, hc2

mat ate2[1,1] = - _b[treat]
mat ate2[1,2] = _se[treat]
mat ate2[1,3] = ate2[1,1] - _se[treat] * 1.96
mat ate2[1,4] = ate2[1,1] + _se[treat] * 1.96

mat att2[1,1] = - _b[treat]
mat att2[1,2] = _se[treat]
mat att2[1,3] = att2[1,1] - _se[treat] * 1.96
mat att2[1,4] = att2[1,1] + _se[treat] * 1.96




*OLS
local base_count = 2
foreach num of numlist 0 2 {
local count = `base_count'

foreach cv in a b c {
di "`covars_`cv''"
di "`num'"
reg re78 treat `covars_`cv'' if treat ==1 | treat == `num', hc2
mat ate`num'[`count',1] = _b[treat] * (1 - `num') // stata thinks 2 is treatment
mat ate`num'[`count',2] = _se[treat]
mat ate`num'[`count',3] = ate`num'[`count',1] - _se[treat] * 1.96
mat ate`num'[`count',4] = ate`num'[`count',1] + _se[treat] * 1.96
```

```
mat att`num'[`count',1] = _b[treat] * (1 - `num') // stata thinks 2 is treatment
mat att`num'[`count',2] = _se[treat]
mat att`num'[`count',3] = att`num'[`count',1] - _se[treat] * 1.96
mat att`num'[`count',4] = att`num'[`count',1] + _se[treat] * 1.96

local ++count
}
}


*Reg. Impute
local base_count = `count'
foreach num of numlist 0 2 {
local count = `base_count'

foreach cv in a b c {
di "`covars_`cv''"
di "`num'"
teffects ra (re78 `covars_`cv'') (treat) if treat ==1 | treat == `num', ate
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of `colnmsb'
local colv: word 1 of `colnmsv'
mat ate`num'[`count',1] = _b[`colb'] * (1 - `num') // stata thinks 2 is treatment
mat ate`num'[`count',2] = _se[`colv']
mat ate`num'[`count',3] = ate`num'[`count',1] - ate`num'[`count',2] * 1.96
mat ate`num'[`count',4] = ate`num'[`count',1] + ate`num'[`count',2] * 1.96

teffects ra (re78 `covars_`cv'') (treat) if treat ==1 | treat == `num', atet
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of `colnmsb'
local colv: word 1 of `colnmsv'
mat att`num'[`count',1] = _b[`colb'] * (1 - `num') // stata thinks 2 is treatment
mat att`num'[`count',2] = _se[`colv']
mat att`num'[`count',3] = att`num'[`count',1] - att`num'[`count',2] * 1.96
mat att`num'[`count',4] = att`num'[`count',1] + att`num'[`count',2] * 1.96

local ++count
}
}


*IPW
local base_count = `count'
foreach num of numlist 0 2 {
local count = `base_count'

foreach cv in a b c {
di "`covars_`cv''"
di "`num'"
capture teffects ipw (re78) (treat `covars_`cv'', logit) if treat ==1 | treat == `num', ate osample(otest) iter(50)
 if _rc==498 {
                display "Overlap Assumption Violated"
                        teffects ipw (re78) (treat `covars_`cv'', logit) if (treat ==1 | treat == `num') & otest ==0, at
}
drop otest
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of `colnmsb'
local colv: word 1 of `colnmsv'
mat ate`num'[`count',1] = _b[`colb'] * (1 - `num') // stata thinks 2 is treatment
mat ate`num'[`count',2] = _se[`colv']
mat ate`num'[`count',3] = ate`num'[`count',1] - ate`num'[`count',2] * 1.96
mat ate`num'[`count',4] = ate`num'[`count',1] + ate`num'[`count',2] * 1.96

capture teffects ipw (re78) (treat `covars_`cv'', logit) if treat ==1 | treat == `num', atet osample(otest) iter(50)
 if _rc==498 {
                display "Overlap Assumption Violated"
                         teffects ipw (re78) (treat `covars_`cv'', logit) if (treat ==1 | treat == `num') & otest ==0, a
}
drop otest
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of `colnmsb'
local colv: word 1 of `colnmsv'
mat att`num'[`count',1] = _b[`colb'] * (1 - `num') // stata thinks 2 is treatment
mat att`num'[`count',2] = _se[`colv']
mat att`num'[`count',3] = att`num'[`count',1] - att`num'[`count',2] * 1.96
mat att`num'[`count',4] = att`num'[`count',1] + att`num'[`count',2] * 1.96


local ++count
}
}


*DR
local base_count = `count'
foreach num of numlist 0 2 {
local count = `base_count'
```

```
foreach cv in a b c {
di "‘covars_‘cv’’"
di "‘num’"
capture teffects ipwra (re78) (treat ‘covars_‘cv’’, logit) if treat ==1 | treat == ‘num’, ate osample(otest) iter(50)
 if _rc==498 {
             display "Overlap Assumption Violated"
                        teffects ipw (re78) (treat ‘covars_‘cv’’, logit) if (treat ==1 | treat == ‘num’) & otest ==0, at
}
drop otest
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of ‘colnmsb’
local colv: word 1 of ‘colnmsv’
mat ate‘num’[‘count’,1] = _b[‘colb’] * (1 − ‘num’) // stata thinks 2 is treatment
mat ate‘num’[‘count’,2] = _se[‘colv’]
mat ate‘num’[‘count’,3] = ate‘num’[‘count’,1] − ate‘num’[‘count’,2] * 1.96
mat ate‘num’[‘count’,4] = ate‘num’[‘count’,1] + ate‘num’[‘count’,2] * 1.96

capture teffects ipwra (re78) (treat ‘covars_‘cv’’, logit) if treat ==1 | treat == ‘num’, atet osample(otest) iter(50)
 if _rc==498 {
             display "Overlap Assumption Violated"
                         teffects ipw (re78) (treat ‘covars_‘cv’’, logit) if (treat ==1 | treat == ‘num’) & otest ==0, a
}
drop otest
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of ‘colnmsb’
local colv: word 1 of ‘colnmsv’
mat att‘num’[‘count’,1] = _b[‘colb’] * (1 − ‘num’) // stata thinks 2 is treatment
mat att‘num’[‘count’,2] = _se[‘colv’]
mat att‘num’[‘count’,3] = att‘num’[‘count’,1] − att‘num’[‘count’,2] * 1.96
mat att‘num’[‘count’,4] = att‘num’[‘count’,1] + att‘num’[‘count’,2] * 1.96


local ++count
}
}


*Reg. Impute
local base_count = ‘count’
foreach num of numlist 0 2 {
local count = ‘base_count’

foreach cv in a b c {
di "‘covars_‘cv’’"
di "‘num’"
teffects nnmatch (re78 ‘covars_‘cv’’) (treat) if treat ==1 | treat == ‘num’, ate nneighbor(1) metric(maha)
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of ‘colnmsb’
local colv: word 1 of ‘colnmsv’
mat ate‘num’[‘count’,1] = _b[‘colb’] * (1 − ‘num’) // stata thinks 2 is treatment
mat ate‘num’[‘count’,2] = _se[‘colv’]
mat ate‘num’[‘count’,3] = ate‘num’[‘count’,1] − ate‘num’[‘count’,2] * 1.96
mat ate‘num’[‘count’,4] = ate‘num’[‘count’,1] + ate‘num’[‘count’,2] * 1.96

teffects nnmatch (re78 ‘covars_‘cv’’) (treat) if treat ==1 | treat == ‘num’, atet nneighbor(1) metric(maha)
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of ‘colnmsb’
local colv: word 1 of ‘colnmsv’
mat att‘num’[‘count’,1] = _b[‘colb’] * (1 − ‘num’) // stata thinks 2 is treatment
mat att‘num’[‘count’,2] = _se[‘colv’]
mat att‘num’[‘count’,3] = att‘num’[‘count’,1] − att‘num’[‘count’,2] * 1.96
mat att‘num’[‘count’,4] = att‘num’[‘count’,1] + att‘num’[‘count’,2] * 1.96

local ++count
}
}

*PS Matching
local base_count = ‘count’
foreach num of numlist 0 2 {
local count = ‘base_count’

foreach cv in a b c {
di "‘covars_‘cv’’"
di "‘num’"
capture teffects psmatch (re78) (treat ‘covars_‘cv’’, logit) if treat ==1 | treat == ‘num’, ate osample(otest) iter(50)
 if _rc==498 {
             display "Overlap Assumption Violated"
                        teffects psmatch (re78) (treat ‘covars_‘cv’’, logit) if (treat ==1 | treat == ‘num’) & otest ==0
}
cap drop otest
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of ‘colnmsb’
local colv: word 1 of ‘colnmsv’
mat ate‘num’[‘count’,1] = _b[‘colb’] * (1 − ‘num’) // stata thinks 2 is treatment
mat ate‘num’[‘count’,2] = _se[‘colv’]
```

```
mat ate`num'[`count',3] = ate`num'[`count',1] - ate`num'[`count',2] * 1.96
mat ate`num'[`count',4] = ate`num'[`count',1] + ate`num'[`count',2] * 1.96

capture teffects psmatch (re78) (treat `covars_`cv'', logit) if treat ==1 | treat == `num', atet osample(otest) iter(50)
 if _rc==498 {
                display "Overlap Assumption Violated"
                        teffects psmatch (re78) (treat `covars_`cv'', logit) if (treat ==1 | treat == `num') & otest ==
}
cap drop otest
local colnmsb: coln e(b)
local colnmsv: coln e(V)
local colb: word 1 of `colnmsb'
local colv: word 1 of `colnmsv'
mat att`num'[`count',1] = _b[`colb'] * (1 - `num') // stata thinks 2 is treatment
mat att`num'[`count',2] = _se[`colv']
mat att`num'[`count',3] = att`num'[`count',1] - att`num'[`count',2] * 1.96
mat att`num'[`count',4] = att`num'[`count',1] + att`num'[`count',2] * 1.96


local ++count
}
}



mat li ate0
mat li ate2

mat li att0
mat li att2
**TODO: put into charts

/*


****
* Question 3
****

clear all
set seed 12345

*construct dgp variance covariance matrix
matrix P = (1,.85 \.85, 1)
mat A = cholesky(P)

program modelsim, rclass
        args A
        drop _all
        set obs 50

        *generate component normal variables
        gen c1= invnorm(uniform())
        gen c2= invnorm(uniform())

        *use cholesky decomp to back out x,z
        gen x = `A'[1,1] * c1 + `A'[1,2] * c2
        gen z = `A'[2,1] * c1 + `A'[2,2] * c2

        *general epsilon and outcome variable
        gen e= invnorm(uniform())
        gen y = 0.5*x + z + e


        *simulate model selection process (flip order of regs for speed)
        reg y x
        scalar beta_tilde = _b[x]

        reg y x z
        scalar beta_hat = _b[x]


        if abs(_b[z]/_se[z])>=1.96 {
                scalar beta_check = beta_hat
        }
        else{
                scalar beta_check = beta_tilde
        }

end

simulate beta_hat = beta_hat beta_check=beta_check beta_tilde=beta_tilde, ///
seed(1234) reps(1000): modelsim A

**TODO: empircal coverage rate

**

sum *
estpost summarize *
estout using hw4_q3_1_stata.tex, cells("mean sd min max") style(tex)  replace
```

```
kdensity beta_hat, normal name(beta_hat,replace)
gr export hw4_q3_bhat_stata.png ,replace
kdensity beta_check, normal name(beta_check,replace)
gr export hw4_q3_bcheck_stata.png ,replace
kdensity beta_tilde,normal name(beta_tilde,replace)
gr export hw4_q3_btilde_stata.png ,replace

sum beta_hat
gen cov_uppers_hat = beta_hat + 1.96*r(sd)/sqrt(50)
gen cov_lowers_hat = beta_hat - 1.96*r(sd)/sqrt(50)
gen covrate_hat =  cond(0.5<= cov_uppers_hat & 0.5>= cov_lowers_hat,1,0)

sum beta_tilde
gen cov_uppers_tilde = beta_tilde + 1.96*r(sd)/sqrt(50)
gen cov_lowers_tilde = beta_tilde - 1.96*r(sd)/sqrt(50)
gen covrate_tilde =  cond(0.5<= cov_uppers_tilde & 0.5>= cov_lowers_tilde,1,0)

sum beta_check
gen cov_uppers_check = beta_check + 1.96*r(sd)/sqrt(50)
gen cov_lowers_check = beta_check - 1.96*r(sd)/sqrt(50)
gen covrate_check =  cond(0.5<= cov_uppers_check & 0.5>= cov_lowers_check,1,0)

sum covrate*
estout using hw4_q3_2_stata.tex, cells("mean ") style(tex)  replace








cap log close
```

R

Question 2

```
####################################################################
# ECON 675, Assignment 4
# Erin Markiewitz
# Fall 2018
# University of Michigan
# Latest update: Nov 9, 2018
####################################################################
rm(list = ls())              #clear workspace
library(foreach)             #for looping
library(data.table)          #for data manipulation
library(Matrix)              #fast matrix calcs
library(ggplot2)             #for pretty plots
library(sandwich)            #for variance-covariance estimation
library(xtable)              #for latex tables
library(boot)                #for bootstrapping
library(CausalGAM)
options(scipen = 999)        #forces R to use normal numbers instead of scientific notation

#######################################################################
# Input data, subset/organize data
#######################################################################
setwd("/Users/erinmarkiewitz/Dropbox/Phd_Coursework/Econ675/hw4")
data <- as.data.table(read.csv('LaLonde_all.csv'))

data = data[,log.re74:=log(re74+1)]
data = data[,log.re75:=log(re75+1)]
data = data[,age.sq:=age^2]
data = data[,educ.sq:=educ^2]
data = data[,age.cu:=age^3]
data = data[,black.u74:=black*u74]
data = data[,educ.logre74:=educ*log.re74]

#subset the data into subsets for LaLonde and PSID controls
X.l = data[treat ==1 | treat ==0]
Y.l = data[treat ==1 | treat ==0,.(re78)]

X.p = data[treat ==1 | treat ==2]
Y.p = data[treat ==1 | treat ==2,.(re78)]

#recode psid treatment indiator
X.p = X.p[,treat:=as.numeric(treat==1)]

#specify covariate sets
X.l.0  = X.l[,.(treat)]
X.p.0  = X.p[,.(treat)]

X.l.a  = X.l[,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75")]
X.p.a  = X.p[,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75")]
```

```
X.l.b  =  X.l[,-c("age.cu","black.u74","educ.logre74","re78","re74","re75")]
X.p.b  =  X.p[,-c("age.cu","black.u74","educ.logre74","re78","re74","re75")]

X.l.c  =  X.l[,-c("re78","re74","re75")]
X.p.c  =  X.p[,-c("re78","re74","re75")]

######################################################################
# 1) difference in means
######################################################################
# lalonde
dm.l    = lm(as.matrix(Y.l)~as.matrix(X.l.0))
dm.l.se = sqrt(diag(vcovHC(dm.l, type = "HC1")))
dm.l.ciu = dm.l$coefficients + 1.96*dm.l.se
dm.l.cil = dm.l$coefficients - 1.96*dm.l.se
dm.l.out = cbind(dm.l$coefficients,dm.l.se,dm.l.cil,dm.l.ciu)

#psid
dm.p = lm(as.matrix(Y.p)~as.matrix(X.p.0))
dm.p.se = sqrt(diag(vcovHC(dm.p, type = "HC1")))
dm.p.ciu = dm.p$coefficients + 1.96*dm.p.se
dm.p.cil = dm.p$coefficients - 1.96*dm.p.se
dm.p.out = cbind(dm.p$coefficients,dm.p.se,dm.p.cil,dm.p.ciu)

######################################################################
# 2) OLS
######################################################################
# lalonde
ols.l.a    = lm(as.matrix(Y.l)~as.matrix(X.l.a))
ols.l.se.a = sqrt(diag(vcovHC(ols.l.a, type = "HC1")))
ols.l.ciu.a = ols.l.a$coefficients + 1.96*ols.l.se.a
ols.l.cil.a = ols.l.a$coefficients - 1.96*ols.l.se.a

ols.l.b    = lm(as.matrix(Y.l)~as.matrix(X.l.b))
ols.l.se.b = sqrt(diag(vcovHC(ols.l.b, type = "HC1")))
ols.l.ciu.b = ols.l.b$coefficients + 1.96*ols.l.se.b
ols.l.cil.b = ols.l.b$coefficients - 1.96*ols.l.se.b

ols.l.c    = lm(as.matrix(Y.l)~as.matrix(X.l.c))
ols.l.se.c = sqrt(diag(vcovHC(ols.l.c, type = "HC1")))
ols.l.ciu.c = ols.l.c$coefficients + 1.96*ols.l.se.c
ols.l.cil.c = ols.l.c$coefficients - 1.96*ols.l.se.c

# PSID
ols.p.a    = lm(as.matrix(Y.p)~as.matrix(X.p.a))
ols.p.se.a = sqrt(diag(vcovHC(ols.p.a, type = "HC1")))
ols.p.ciu.a = ols.p.a$coefficients + 1.96*ols.p.se.a
ols.p.cil.a = ols.p.a$coefficients - 1.96*ols.p.se.a

ols.p.b    = lm(as.matrix(Y.p)~as.matrix(X.p.b))
ols.p.se.b = sqrt(diag(vcovHC(ols.p.b, type = "HC1")))
ols.p.ciu.b = ols.p.b$coefficients + 1.96*ols.p.se.b
ols.p.cil.b = ols.p.b$coefficients - 1.96*ols.p.se.b

ols.p.c    = lm(as.matrix(Y.p)~as.matrix(X.p.c))
ols.p.se.c = sqrt(diag(vcovHC(ols.p.c, type = "HC1")))
ols.p.ciu.c = ols.p.c$coefficients + 1.96*ols.p.se.c
ols.p.cil.c = ols.p.c$coefficients - 1.96*ols.p.se.c

#out
ols.l.out = cbind(c(ols.l.a$coefficients[2],ols.l.b$coefficients[2],ols.l.c$coefficients[2]),c(ols.l.se.a[2],ols.l.se.b[2],
ols.p.out = cbind(c(ols.p.a$coefficients[2],ols.p.b$coefficients[2],ols.p.c$coefficients[2]),c(ols.p.se.a[2],ols.p.se.b[2],

######################################################################
# 3) RI
######################################################################

################
# covariates a #
################
# subset the outcome and coveriate series
Y.treat       = data[treat==1,.(re78)]
Y.control.l   = data[treat==0,.(re78)]
Y.control.p   = data[treat==2,.(re78)]

#subset covariates for imputation
X.treat.a      = data[treat==1,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","
X.control.l.a  = data[treat==0,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","
X.control.p.a  = data[treat==2,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","

#estimate ols coefficients for imputation
ols.treat.a         = lm(as.matrix(Y.treat)~as.matrix(X.treat.a))
ols.control.l.a     = lm(as.matrix(Y.control.l)~as.matrix(X.control.l.a))
ols.control.p.a     = lm(as.matrix(Y.control.p)~as.matrix(X.control.p.a))

#insert constants
X.treat.a[,const:=1]
setcolorder(X.treat.a,c("const"))
X.control.l.a[,const:=1]
setcolorder(X.control.l.a,c("const"))
X.control.p.a[,const:=1]
setcolorder(X.control.p.a,c("const"))
```

```
#impute individual treatment effects
tvec.ri.treat.l.a      = as.matrix(X.treat.a)%*%(as.vector(ols.treat.a$coefficients)-as.vector(ols.control.l.a$coefficients
tvec.ri.treat.p.a      = as.matrix(X.treat.a)%*%(as.vector(ols.treat.a$coefficients)-as.vector(ols.control.p.a$coefficients

tvec.ri.control.l.a    = as.matrix(X.control.l.a)%*%(as.vector(ols.treat.a$coefficients)-as.vector(ols.control.l.a$coefficie
tvec.ri.control.p.a    = as.matrix(X.control.p.a)%*%(as.vector(ols.treat.a$coefficients)-as.vector(ols.control.p.a$coefficie


#ATE
ate.ri.l.a       = mean(c(tvec.ri.treat.l.a,tvec.ri.control.l.a))
ate.ri.p.a       = mean(c(tvec.ri.treat.p.a,tvec.ri.control.p.a))


#ATT
att.ri.a          = mean(tvec.ri.treat.l.a)


################
# covariates b #
################
Y.treat       = data[treat==1,.(re78)]
Y.control.l   = data[treat==0,.(re78)]
Y.control.p   = data[treat==2,.(re78)]

#subset covariates for imputation
X.treat.b      = data[treat==1,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","
X.control.l.b  = data[treat==0,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","
X.control.p.b  = data[treat==2,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","

#estimate ols coefficients for imputation
ols.treat.b         = lm(as.matrix(Y.treat)~as.matrix(X.treat.b))
ols.control.l.b     = lm(as.matrix(Y.control.l)~as.matrix(X.control.l.b))
ols.control.p.b     = lm(as.matrix(Y.control.p)~as.matrix(X.control.p.b))

#insert constants
X.treat.b[,const:=1]
setcolorder(X.treat.b,c("const"))
X.control.l.b[,const:=1]
setcolorder(X.control.l.b,c("const"))
X.control.p.b[,const:=1]
setcolorder(X.control.p.b,c("const"))

#impute individual treatment effects
tvec.ri.treat.l.b      = as.matrix(X.treat.b)%*%(as.vector(ols.treat.b$coefficients)-as.vector(ols.control.l.b$coefficients
tvec.ri.treat.p.b      = as.matrix(X.treat.b)%*%(as.vector(ols.treat.b$coefficients)-as.vector(ols.control.p.b$coefficients

tvec.ri.control.l.b    = as.matrix(X.control.l.b)%*%(as.vector(ols.treat.b$coefficients)-as.vector(ols.control.l.b$coefficie
tvec.ri.control.p.b    = as.matrix(X.control.p.b)%*%(as.vector(ols.treat.b$coefficients)-as.vector(ols.control.p.b$coefficie


#ATE
ate.ri.l.b       = mean(c(tvec.ri.treat.l.b,tvec.ri.control.l.b))
ate.ri.p.b       = mean(c(tvec.ri.treat.p.b,tvec.ri.control.p.b))


#ATT
att.ri.b          = mean(tvec.ri.treat.l.b)

################
# covariates c #
################

Y.treat       = data[treat==1,.(re78)]
Y.control.l   = data[treat==0,.(re78)]
Y.control.p   = data[treat==2,.(re78)]

#subset covariates for imputation
X.treat.c      = data[treat==1,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","
X.control.l.c  = data[treat==0,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","
X.control.p.c  = data[treat==2,-c("age.sq","educ.sq","age.cu","black.u74","educ.logre74","u74","u75","re78","re74","re75","

#estimate ols coefficients for imputation
ols.treat.c         = lm(as.matrix(Y.treat)~as.matrix(X.treat.c))
ols.control.l.c     = lm(as.matrix(Y.control.l)~as.matrix(X.control.l.c))
ols.control.p.c     = lm(as.matrix(Y.control.p)~as.matrix(X.control.p.c))

#insert constants
X.treat.c[,const:=1]
setcolorder(X.treat.c,c("const"))
X.control.l.c[,const:=1]
setcolorder(X.control.l.c,c("const"))
X.control.p.c[,const:=1]
setcolorder(X.control.p.c,c("const"))

#impute individual treatment effects
tvec.ri.treat.l.c      = as.matrix(X.treat.c)%*%(as.vector(ols.treat.c$coefficients)-as.vector(ols.control.l.c$coefficients
tvec.ri.treat.p.c      = as.matrix(X.treat.c)%*%(as.vector(ols.treat.c$coefficients)-as.vector(ols.control.p.c$coefficients

tvec.ri.control.l.c    = as.matrix(X.control.l.c)%*%(as.vector(ols.treat.c$coefficients)-as.vector(ols.control.l.c$coefficie
tvec.ri.control.p.c    = as.matrix(X.control.p.c)%*%(as.vector(ols.treat.c$coefficients)-as.vector(ols.control.p.c$coefficie
```

```
#ATE
ate.ri.l.c       = mean(c(tvec.ri.treat.l.c,tvec.ri.control.l.c))
ate.ri.p.c       = mean(c(tvec.ri.treat.p.c,tvec.ri.control.p.c))

#ATT
att.ri.c          = mean(tvec.ri.treat.l.c)

#####################################################################
# IPW and Doubly Robust  (ATT)
#####################################################################

# Generate treatment outcome variables
T.l = data[treat==1|treat==0,.(treat)]
T.p = data[treat==1|treat==2,.(treat)]

#Recode 2's to 0's for PSID sample
T.p = T.p[,treat:=as.numeric(treat==1)]

# Get propensity scores using logit regression
prop.l.a = glm(as.matrix(T.l) ~ as.matrix(X.l.a[,-c("treat")]),family = "binomial")
prop.l.b = glm(as.matrix(T.l) ~ as.matrix(X.l.b[,-c("treat")]),family = "binomial")
prop.l.c = glm(as.matrix(T.l) ~ as.matrix(X.l.c[,-c("treat")]),family = "binomial")

prop.p.a = glm(as.matrix(T.p) ~ as.matrix(X.p.a[,-c("treat")]),family = "binomial")
prop.p.b = glm(as.matrix(T.p) ~ as.matrix(X.p.b[,-c("treat")]),family = "binomial")
prop.p.c = glm(as.matrix(T.p) ~ as.matrix(X.p.c[,-c("treat")]),family = "binomial")

# Add prop scores to the data matrices for easy computing of treatment effects
X.l.ipw = X.l
X.l.ipw[,ps.a:=prop.l.a$fitted.values]
X.l.ipw[,ps.b:=prop.l.b$fitted.values]
X.l.ipw[,ps.c:=prop.l.c$fitted.values]

X.p.ipw = X.p
X.p.ipw[,ps.a:=prop.p.a$fitted.values]
X.p.ipw[,ps.b:=prop.p.b$fitted.values]
X.p.ipw[,ps.c:=prop.p.c$fitted.values]

# Create variables for computing ATEs
X.l.ipw[,t1.a:=treat*re78/ps.a]
X.l.ipw[,t0.a:=(1-treat)*re78/(1-ps.a)]
X.l.ipw[,t1.b:=treat*re78/ps.b]
X.l.ipw[,t0.b:=(1-treat)*re78/(1-ps.b)]
X.l.ipw[,t1.c:=treat*re78/ps.c]
X.l.ipw[,t0.c:=(1-treat)*re78/(1-ps.c)]

# Compute proportion of treated respondents
p.l          = mean(X.l[,treat])

# Create additional variables for computing ATTs
X.l.ipw[,t1.att:=treat*re78/p.l]
X.l.ipw[,t0.a2:=(1-treat)*re78/(1-ps.a)*(ps.a/p.l)]
X.l.ipw[,t0.b2:=(1-treat)*re78/(1-ps.b)*(ps.b/p.l)]
X.l.ipw[,t0.c2:=(1-treat)*re78/(1-ps.c)*(ps.c/p.l)]

# Compute ATTs
att.ipw.l.a  = mean(X.l.ipw[,t1.att])-mean(X.l.ipw[,t0.a2])
att.ipw.l.b  = mean(X.l.ipw[,t1.att])-mean(X.l.ipw[,t0.b2])
att.ipw.l.c  = mean(X.l.ipw[,t1.att])-mean(X.l.ipw[,t0.c2])



#####################################################################
# [4.b] Inverse Probability Weighting, PSID control
#####################################################################

# Create variables for computing ATEs
X.p.ipw[,t1.a:=treat*re78/ps.a]
X.p.ipw[,t0.a:=(1-treat)*re78/(1-ps.a)]
X.p.ipw[,t1.b:=treat*re78/ps.b]
X.p.ipw[,t0.b:=(1-treat)*re78/(1-ps.b)]
X.p.ipw[,t1.c:=treat*re78/ps.c]
X.p.ipw[,t0.c:=(1-treat)*re78/(1-ps.c)]

# Compute proportion of treated respondents
p.p          = mean(X.p[,treat])

# Create additional variables for computing ATTs
X.p.ipw[,t1.att:=treat*re78/p.p]
X.p.ipw[,t0.a2:=(1-treat)*re78/(1-ps.a)*(ps.a/p.p)]
X.p.ipw[,t0.b2:=(1-treat)*re78/(1-ps.b)*(ps.b/p.p)]
X.p.ipw[,t0.c2:=(1-treat)*re78/(1-ps.c)*(ps.c/p.p)]

# Compute ATTs
att.ipw.p.a  = mean(X.p.ipw[,t1.att])-mean(X.p.ipw[,t0.a2])
att.ipw.p.b  = mean(X.p.ipw[,t1.att])-mean(X.p.ipw[,t0.b2])
att.ipw.p.c  = mean(X.p.ipw[,t1.att])-mean(X.p.ipw[,t0.c2])
```

```
####################################################################
# IPW and Doubly Robust  (ATE)
####################################################################

# Covariates A
ATE.l.a <- estimate.ATE(pscore.formula = treat ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75,
                         pscore.family = binomial,
                         outcome.formula.t = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75,
                         outcome.formula.c = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75,
                         outcome.family = gaussian,
                         treatment.var = "treat",
                         data=as.data.frame(X.l),
                         divby0.action="t",
                         divby0.tol=0.001,
                         var.gam.plot=FALSE,
                         nboot=0
)

# Covariates B
ATE.l.b <- estimate.ATE(pscore.formula = treat ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + age.s
                         pscore.family = binomial,
                         outcome.formula.t = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.formula.c = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.family = gaussian,
                         treatment.var = "treat",
                         data=as.data.frame(X.l),
                         divby0.action="t",
                         divby0.tol=0.001,
                         var.gam.plot=FALSE,
                         nboot=0
)

# Covariates C
ATE.l.c <- estimate.ATE(pscore.formula = treat ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + age.s
                         pscore.family = binomial,
                         outcome.formula.t = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.formula.c = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.family = gaussian,
                         treatment.var = "treat",
                         data=as.data.frame(X.l),
                         divby0.action="t",
                         divby0.tol=0.001,
                         var.gam.plot=FALSE,
                         nboot=0
)

 #Covariates A, PSID control #can't calculate standard error
 ATE.p.a <- estimate.ATE(pscore.formula = treat ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75,
                          pscore.family = binomial,
                          outcome.formula.t = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75,
                          outcome.formula.c = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75,
                          outcome.family = gaussian,
                          treatment.var = "treat",
                          data=as.data.frame(X.p),
                          divby0.action="t",
                          divby0.tol=0.001,
                          var.gam.plot=FALSE,
                          nboot=0,
                          variance.smooth.span = 0
 )

# Covariates B, PSID control
ATE.p.b <- estimate.ATE(pscore.formula = treat ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + age.s
                         pscore.family = binomial,
                         outcome.formula.t = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.formula.c = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.family = gaussian,
                         treatment.var = "treat",
                         data=as.data.frame(X.p),
                         divby0.action="t",
                         divby0.tol=0.001,
                         var.gam.plot=FALSE,
                         nboot=0
)

# Covariates C, PSID control
ATE.p.c <- estimate.ATE(pscore.formula = treat ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + age.s
                         pscore.family = binomial,
                         outcome.formula.t = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.formula.c = re78 ~ age + educ + black + hisp + married + nodegr + log.re74 + log.re75 + ag
                         outcome.family = gaussian,
                         treatment.var = "treat",
                         data=as.data.frame(X.p),
                         divby0.action="t",
                         divby0.tol=0.001,
                         var.gam.plot=FALSE,
                         nboot=0
)

####################################################################
#  construct table
####################################################################
```

```
## lalonde controls
# Mean Diff + OLS results
a  =    rbind(dm.l.out[2,],ols.l.out)

# Reg imputation results
b1  =   c(ATE.l.a$ATE.reg.hat,ATE.l.a$ATE.reg.asymp.SE,ATE.l.a$ATE.reg.hat-1.96*ATE.l.a$ATE.reg.asymp.SE,ATE.l.a$ATE.reg.ha
b2  =   c(ATE.l.b$ATE.reg.hat,ATE.l.b$ATE.reg.asymp.SE,ATE.l.b$ATE.reg.hat-1.96*ATE.l.b$ATE.reg.asymp.SE,ATE.l.b$ATE.reg.ha
b3  =   c(ATE.l.c$ATE.reg.hat,ATE.l.c$ATE.reg.asymp.SE,ATE.l.c$ATE.reg.hat-1.96*ATE.l.c$ATE.reg.asymp.SE,ATE.l.c$ATE.reg.ha

# IPW results
c1  =   c(ATE.l.a$ATE.IPW.hat,ATE.l.a$ATE.IPW.asymp.SE,ATE.l.a$ATE.IPW.hat-1.96*ATE.l.a$ATE.IPW.asymp.SE,ATE.l.a$ATE.IPW.ha
c2  =   c(ATE.l.b$ATE.IPW.hat,ATE.l.b$ATE.IPW.asymp.SE,ATE.l.b$ATE.IPW.hat-1.96*ATE.l.b$ATE.IPW.asymp.SE,ATE.l.b$ATE.IPW.ha
c3  =   c(ATE.l.c$ATE.IPW.hat,ATE.l.c$ATE.IPW.asymp.SE,ATE.l.c$ATE.IPW.hat-1.96*ATE.l.c$ATE.IPW.asymp.SE,ATE.l.c$ATE.IPW.ha

# Doubly robust results
d1  =   c(ATE.l.a$ATE.AIPW.hat,ATE.l.a$ATE.AIPW.asymp.SE,ATE.l.a$ATE.AIPW.hat-1.96*ATE.l.a$ATE.AIPW.asymp.SE,ATE.l.a$ATE.AIP
d2  =   c(ATE.l.b$ATE.AIPW.hat,ATE.l.b$ATE.AIPW.asymp.SE,ATE.l.b$ATE.AIPW.hat-1.96*ATE.l.b$ATE.AIPW.asymp.SE,ATE.l.b$ATE.AIP
d3  =   c(ATE.l.c$ATE.AIPW.hat,ATE.l.c$ATE.AIPW.asymp.SE,ATE.l.c$ATE.AIPW.hat-1.96*ATE.l.c$ATE.AIPW.asymp.SE,ATE.l.c$ATE.AIP

## PSID control

# Mean Diff + OLS results
e  =    rbind(dm.p.out[2,],ols.p.out)

# Reg imputation results
f1  =   c(ATE.p.a$ATE.reg.hat,0,0,0)
f2  =   c(ATE.p.b$ATE.reg.hat,ATE.p.b$ATE.reg.asymp.SE,ATE.p.b$ATE.reg.hat-1.96*ATE.p.b$ATE.reg.asymp.SE,ATE.p.b$ATE.reg.ha
f3  =   c(ATE.p.c$ATE.reg.hat,ATE.p.c$ATE.reg.asymp.SE,ATE.p.c$ATE.reg.hat-1.96*ATE.p.c$ATE.reg.asymp.SE,ATE.p.c$ATE.reg.ha

# IPW results
g1  =   c(ATE.p.a$ATE.IPW.hat,0,0,0)
g2  =   c(ATE.p.b$ATE.IPW.hat,ATE.p.b$ATE.IPW.asymp.SE,ATE.p.b$ATE.IPW.hat-1.96*ATE.p.b$ATE.IPW.asymp.SE,ATE.p.b$ATE.IPW.ha
g3  =   c(ATE.p.c$ATE.IPW.hat,ATE.p.c$ATE.IPW.asymp.SE,ATE.p.c$ATE.IPW.hat-1.96*ATE.p.c$ATE.IPW.asymp.SE,ATE.p.c$ATE.IPW.ha

# Doubly robust results
h1  =   c(ATE.p.a$ATE.AIPW.hat,0,0,0)
h2  =   c(ATE.p.b$ATE.AIPW.hat,ATE.p.b$ATE.AIPW.asymp.SE,ATE.p.b$ATE.AIPW.hat-1.96*ATE.p.b$ATE.AIPW.asymp.SE,ATE.p.b$ATE.AIP
h3  =   c(ATE.p.c$ATE.AIPW.hat,ATE.p.c$ATE.AIPW.asymp.SE,ATE.p.c$ATE.AIPW.hat-1.96*ATE.p.c$ATE.AIPW.asymp.SE,ATE.p.c$ATE.AIP


## PUT RESULTS TOGETHER
l.out  = rbind(a,b1,b2,b3,c1,c2,c3,d1,d2,d3)
p.out  = rbind(e,f1,f2,f3,g1,g2,g3,h1,h2,h3)
ate.out = round(cbind(l.out,p.out),2)

# EXPORT RESULTS AS CSV
write.table(ate.out, file = "Table1_ATE_resultq.csv",row.names=FALSE,col.names=FALSE,sep=",")
```

Question 3

```
################################################################
# ECON 675, Assignment 4
# Erin Markiewitz
# Fall 2018
# University of Michigan
# Latest update: Nov 9, 2018
################################################################


rm(list=ls(all=TRUE))
library(foreign); library(MASS);
library(boot)
library(data.table)
library(foreach)
library(data.table)
library(Matrix)
library(ggplot2)
library(sandwich)
library(xtable)
library(mvtnorm)
set.seed(12345)
setwd("/Users/erinmarkiewitz/Dropbox/Phd_Coursework/Econ675/hw4")

####
# Generate data
####

obs = 50
reps = 1000
covmatrix = matrix(c(1,0.85,.85,1),2,2)


#Generate X and Z, e, and y
W = replicate(reps,rmvnorm(obs, mean = c(0,0), sigma = covmatrix, method = "chol"))
e = replicate(reps,rnorm(50))
Y = sapply(1:reps,function(i) rep(1,obs) + W[,,i]%*%c(.5,1)+e[,i])

#estimate beta hat and gamma_tstats
beta_hat = sapply(1:reps, function(i) lm(Y[,i]~W[,,i])$coefficients[2])
```

```
gamma_tstat = sapply(1:reps,function(i) summary(lm(Y[,i]~W[,,i]))[["coefficients"]][, "t value"][3])
beta_hat_se = sapply(1:reps,function(i) summary(lm(Y[,i]~W[,1,i]))[["coefficients"]][, "Std. Error"][2])
lol = mean(beta_hat_se)


#estimate beta tilde
beta_tilde = sapply(1:reps, function(i) lm(Y[,i]~W[,2,i])$coefficients[2])

#estimate beta check
beta_check= ifelse(gamma_tstat>=1.96,beta_hat,beta_tilde)

# Summary statistics
sum_beta = rbind(summary(beta_hat),summary(beta_tilde),summary(beta_check))
print(xtable(sum_beta, type = "latex"), file = "hw4_q3_1_r.tex")

plot_dat = data.frame(beta = c(beta_hat,beta_tilde,beta_check),Estimator=rep(c("hat", "tilde","check"), each = reps))

densplot = ggplot(plot_dat,aes(x=beta,fill=Estimator))+
  geom_density(alpha=0.5, kernel="e",bw="ucv")+
  ggtitle("Kernel Density Plots")+
  xlab("Point Estimator")+
  ylab("Density")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_fill_discrete(
    name="Estimator",
    breaks=c("hat", "tilde", "check"),
    labels=c("hat", "tilde", "check" ))+
  theme(legend.justification = c(0.05, 0.98), legend.position = c(0.05, 0.98)) +  stat_function(fun = dnorm, n = 5000, args
ggsave("hw4_q3_1_r.png")

#Coverage rates

#Compute coverage rate for beta_hat
beta_hat_cis      = cbind(beta_hat-1.96*beta_hat_se,beta_hat+1.96*beta_hat_se)
beta_hat_covdum  = ifelse(0.5>=beta_hat_cis[,1]&0.5<=beta_hat_cis[,2],1,0)
beta_hat_cr         = mean(beta_hat_covdum)

# Compute coverage rate for beta_tilde

beta_tilde_se        = sapply(1:reps,function(i) summary(lm(Y[,i]~W[,1,i]))[["coefficients"]][, "Std. Error"][2])
beta_tilde_cis      = cbind(beta_tilde-1.96*beta_tilde_se,beta_tilde+1.96*beta_tilde_se)
beta_tilde_covdum  = ifelse(0.5>=beta_tilde_cis[,1]&0.5<=beta_tilde_cis[,2],1,0)
beta_tilde_cr         = mean(beta_tilde_covdum)

# Compute coverage rate for beta_check
beta_check_cip     = ifelse(beta_hat==beta_check,beta_hat-1.96*beta_hat_se,beta_tilde-1.96*beta_tilde_se)
beta_check_cil     = ifelse(beta_hat==beta_check,beta_hat+1.96*beta_hat_se,beta_tilde+1.96*beta_tilde_se)
beta_check_cis     = cbind(beta_check_cil,beta_check_cip)
beta_check_covdum = ifelse(0.5>=beta_check_cis[,1]&0.5<=beta_check_cis[,2],1,0)
beta_check_cr      = mean(beta_check_covdum)

# Put results together
cr_table            = rbind(beta_hat_cr,beta_tilde_cr,beta_check_cr)
rownames(cr_table) = c("beta_hat","beta_tilde","beta_check")
colnames(cr_table) = c("Coverage Rate")

print(xtable(cr_table, type = "latex"), file = "hw4_q3_2_r.tex")
```