ECON675 – Assignment 3

Anirudh Yadav

October 29, 2018

Contents

1 Non-linear least squares

1.1 Identifiability

This is a standard M-estimation problem. The parameter vector $\boldsymbol{\beta}_0$ is assumed to solve the population problem

$$\boldsymbol{\beta}_0 = \arg\min_{\beta \in \mathbb{R}^d} \mathbb{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2].$$

For $\boldsymbol{\beta}_0$ to be identified, it must be the *unique* solution to the above population problem (i.e. the unique minimizer). In math, this means for all $\epsilon > 0$ and for some $\delta > 0$:

$$\sup_{||\beta - \beta_0|| > \epsilon} M(\boldsymbol{\beta}) \geq M(\boldsymbol{\beta}_0) + \delta$$

where $M(\boldsymbol{\beta}) = \mathbb{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2]$. Of course $\boldsymbol{\beta}_0$ can be written in closed form if $\mu(\cdot)$ is linear. In this case, we know that

$$\boldsymbol{\beta}_0 = \mathbb{E}[\boldsymbol{x}_i\boldsymbol{x}_i']^{-1}\mathbb{E}[\boldsymbol{x}_i y_i].$$

1.2 Asymptotic normality

The M-estimator is asymptotically normal if:

1. $\hat{\boldsymbol{\beta}} \to_p \boldsymbol{\beta}_0$
2. $\boldsymbol{\beta}_0 \in int(B)$ and $m(\boldsymbol{x}_i, \boldsymbol{\beta}) \equiv (y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2$ is 3 times continuously differentiable.
3. $\Sigma_0 = \mathbb{V}[\frac{\partial}{\partial \beta} m(\boldsymbol{x}_i; \beta_0)] < \infty$ and $H_0 = \mathbb{E}[\frac{\partial^2}{\partial \beta \partial \beta'} m(\boldsymbol{x}_i; \beta_0)]$ is full rank (and therefore invertible).

Now, the FOC for the M-estimation problem is

$$0 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}))\boldsymbol{x}_i \tag{1}$$

where $\dot{\mu} = \frac{\partial}{\partial \beta}\mu(\boldsymbol{x}_i'\boldsymbol{\beta})$. So, we've converted the M-estimation problem into a Z-estimation problem. Then we can use the standard asymptotic normality result to arrive at a precise form of the asymptotic variance:

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \to_d \mathcal{N}(0, H_0^{-1}\Sigma_0 H_0^{-1}).$$

Now, taking the second derivative gives the Hessian

$$\begin{aligned}
H_0 &= \mathbb{E}[\frac{\partial^2}{\partial \beta \partial \beta'} m(\boldsymbol{x}_i; \beta_0)] \\
&= \mathbb{E}\left[-\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))\boldsymbol{x}_i\boldsymbol{x}_i' + (y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))\ddot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))\boldsymbol{x}_i\boldsymbol{x}_i'\right] \\
&= -\mathbb{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\boldsymbol{x}_i\boldsymbol{x}_i']
\end{aligned}$$

by LIE. And, the variance of the score is

$$\begin{aligned}\Sigma_0 &= \mathbb{V}[\frac{\partial}{\partial\beta}m(\boldsymbol{x}_i;\beta_0)] \\ &= \mathbb{E}\left[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\boldsymbol{x}_i\boldsymbol{x}_i'\right] \\ &= \mathbb{E}[\sigma^2(\boldsymbol{x}_i)\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\boldsymbol{x}_i\boldsymbol{x}_i']\end{aligned}$$

again by LIE. Then we have the asymptotic variance

$$\boldsymbol{V}_0 = H_0^{-1}\Sigma_0 H_0^{-1}.$$

1.3 Variance estimator under heteroskedasticity

Under heteroskedasticity we can use the sandwich variance estimator

$$\widehat{\boldsymbol{V}}_{HC} = \hat{H}^{-1}\hat{\Sigma}\hat{H}^{-1},$$

where

$$\begin{aligned}\hat{H} &= \frac{1}{n}\sum_{i=1}^{n}\dot{\mu}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}))^2\boldsymbol{x}_i\boldsymbol{x}_i' \\ \hat{\Sigma} &= \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i^2\dot{\mu}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}))^2\boldsymbol{x}_i\boldsymbol{x}_i'\end{aligned}$$

Now, to get an asymptotically valid CI for $||\boldsymbol{\beta}_0||^2$ we need to use the Delta Method. First, note that:

$$\begin{aligned}&||\boldsymbol{\beta}_0||^2 = \boldsymbol{\beta}_0'\boldsymbol{\beta}_0 \\ \implies &\frac{\partial}{\partial\beta}||\boldsymbol{\beta}_0||^2 = 2\boldsymbol{\beta}_0\end{aligned}$$

Then, using the Delta Method

$$\begin{aligned}\sqrt{n}(||\hat{\boldsymbol{\beta}}||^2 - ||\boldsymbol{\beta}_0||^2) &\to_d 2\boldsymbol{\beta}_0\mathcal{N}(0, \boldsymbol{V}_0) \\ &= \mathcal{N}(0, 4\boldsymbol{\beta}_0'\boldsymbol{V}_0\boldsymbol{\beta}_0)\end{aligned}$$

Thus, an asymptotically valid 95% CI for $||\boldsymbol{\beta}_0||^2$ is

$$CI_{95} = \left[\hat{\boldsymbol{\beta}} - 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}'\widehat{\boldsymbol{V}}_{HC}\hat{\boldsymbol{\beta}}}{n}}, \hat{\boldsymbol{\beta}} + 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}'\widehat{\boldsymbol{V}}_{HC}\hat{\boldsymbol{\beta}}}{n}}\right]$$

1.4 Variance estimator under homoskedasticity

Using the above results, under homoskedasticity, the asymptotic variance collapses to

$$\begin{aligned}\boldsymbol{V}_0 &= \mathbb{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\boldsymbol{x}_i\boldsymbol{x}_i']^{-1}\sigma^2\mathbb{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}))^2\boldsymbol{x}_i\boldsymbol{x}_i']\mathbb{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\boldsymbol{x}_i\boldsymbol{x}_i']^{-1}\\ &= \sigma^2\mathbb{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\boldsymbol{x}_i\boldsymbol{x}_i']^{-1}\end{aligned}$$

The variance estimator is now takes a simpler form

$$\widehat{\boldsymbol{V}}_{HO} = \hat{\sigma}^2\hat{H}^{-1}$$

where $\hat{H}$ is the same as above and

$$\hat{\sigma}^2 = \frac{1}{n-d}\sum_{i=1}^n(y_i - \mu(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}))^2$$

Then, as above, the asymptotically valid 95% CI for $||\boldsymbol{\beta}_0||^2$ is

$$CI_{95} = \left[\hat{\boldsymbol{\beta}} - 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}'\widehat{\boldsymbol{V}}_{HO}\hat{\boldsymbol{\beta}}}{n}}, \hat{\boldsymbol{\beta}} + 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}'\widehat{\boldsymbol{V}}_{HO}\hat{\boldsymbol{\beta}}}{n}}\right].$$

1.5 MLE

Given the assumption of a normal DGP we have the conditional density

$$f(y_i|\boldsymbol{x}_i) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\Big(-\frac{1}{2\sigma^2}(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\Big).$$

Then, the sample log-likelihood function is

$$\ell(\boldsymbol{\beta}, \sigma^2; \boldsymbol{X}) = n\log\Big(\frac{1}{\sqrt{2\pi\sigma^2}}\Big) - \frac{1}{2\sigma^2}\sum_{i=1}^n(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2$$

Dividing by n gives

$$\ell(\boldsymbol{\beta}, \sigma^2; \boldsymbol{X}) = \log\Big(\frac{1}{\sqrt{2\pi\sigma^2}}\Big) - \frac{1}{n2\sigma^2}\sum_{i=1}^n(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2$$

The FOC wrt $\boldsymbol{\beta}$ is

$$0 = \frac{1}{n\sigma^2}\sum_{i=1}^n(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}))\boldsymbol{x}_i,$$

which is equivalent to the FOC for the M-estimation problem (1) (since σ^2 just scales the FOC, it does not affect the solution). Thus,

$$\hat{\boldsymbol{\beta}}_{MLE} = \hat{\boldsymbol{\beta}}_{M.est}.$$

Now, the FOC of the log-likelihood wrt σ^2 is

$$0 = -\frac{1}{2}(2\pi\sigma^2)^{-1}2\pi + \frac{1}{2n}(\sigma^2)^{-2}\sum_{i=1}^{n}(y_i - \mu(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}))^2$$

Solving for σ^2 gives the MLE:

$$\hat{\sigma}^2_{ML} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}))^2,$$

which is not the same as the estimator proposed in [4], since it does not adjust for the number of regressors.

1.6 When the link function is unknown

Suppose the link function is unknown, and consider two pairs of true parameters, $(\mu_1, \boldsymbol{\beta}_1)$ and $(\mu_2, \boldsymbol{\beta}_2)$ where $\mu_2(u) = \mu_1(u/c)$ and $\boldsymbol{\beta}_2 = c\boldsymbol{\beta}_1$ for some $c \neq 0$. Then the parameters are clearly different, but $\mu_1(\boldsymbol{x}_i'\boldsymbol{\beta}_1) = \mu_2(\boldsymbol{x}_i'\boldsymbol{\beta}_2)$.

1.7 Logistic link function

The link function is

$$\begin{aligned}
\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) &= \mathbb{E}[y_i|\boldsymbol{x}_i] \\
&= \mathbb{E}[\mathbf{1}(\boldsymbol{x}_i'\boldsymbol{\beta}_0 \geq \epsilon_i)|\boldsymbol{x}_i] \\
&= \Pr[\boldsymbol{x}_i'\boldsymbol{\beta}_0 \geq \epsilon_i|\boldsymbol{x}_i] \\
&= F(\boldsymbol{x}_i'\boldsymbol{\beta}_0) \\
&= \frac{1}{1+\exp(-\boldsymbol{x}_i'\boldsymbol{\beta}_0)}, \text{ if } s_0 = 1.
\end{aligned}$$

The conditional variance of y_i is

$$\sigma^2(\boldsymbol{x}_i)\mathbb{V}[y_i|\boldsymbol{x}_i]$$

Now, note that $y_i|\boldsymbol{x}_i$ is a Bernoulli random variable, with $\Pr[y_i = 1|\boldsymbol{x}_i] = F(\boldsymbol{x}_i'\boldsymbol{\beta}_0)$. Then

$$\begin{aligned}
\sigma^2(\boldsymbol{x}_i) &= F(\boldsymbol{x}_i'\boldsymbol{\beta}_0)(1 - F(\boldsymbol{x}_i'\boldsymbol{\beta}_0)) \\
&= \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0)(1 - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))
\end{aligned}$$

To derive an expression for the asymptotic variance, first note that for the logistic cdf: $\dot{\mu}(u) = (1-\mu(u))\mu(u)$. Then, the asymptotic variance is

$$\boldsymbol{V}_0 = H_0^{-1}\Sigma_0 H_0^{-1}.$$

where

$$H_0 = \mathbb{E}[(1 - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0)^2\boldsymbol{x}_i\boldsymbol{x}_i']$$

and

$$\Sigma_0 = \mathbb{E}[(1 - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^3\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0)^3\boldsymbol{x}_i\boldsymbol{x}_i']$$

1.8 Logistic link function, MLE

MLE gives the same point estimator as NLS (i.e. they have the same FOC; we did this in 672), but MLE is asymptotically efficient, so $\boldsymbol{V}_0^{ML} \leq \boldsymbol{V}_0^{NLS}$.

1.9 Some data work

(a) I estimated the logistic model with robust (HC1) standard errors in both `R` and `Stata`. The results from `R` are presented in Table 1. The standard errors from `Stata` are very slightly different, but I'm not sure why.

Table 1: **Logistic Regression Estimates for** $s = 1-$`dmissing`

	Coef.	Std. Err.	t-stat	p-val	CI.lower	CI.upper
Const.	1.755	0.335	5.245	0.000	1.099	2.411
S_age	1.333	0.123	10.826	0.000	1.092	1.575
S_HHpeople	-0.067	0.023	-2.871	0.004	-0.112	-0.021
$\log(\text{inc}+1)$	-0.119	0.044	-2.707	0.007	-0.205	-0.033

(b) Table 2 presents the 95% confidence interval and p-values for each coefficient derived from 999 bootstrap replications of the t-statistic: $t^* = (\beta^* - \hat{\beta}_{obs})/se^*$. The statistics are very similar to those in Table 1, which rely on large sample approximations.

The idea for computing bootstrapped CIs is simple: for each bootstrap replication, compute t^* for each coefficient; this gives an empirical distribution for t^*; then extract the desired quantiles from the empirical distribution, and compute the confidence intervals as

$$CI_{95}^{boot}(\beta) = \left[\hat{\beta}_{obs} + q^*_{0.025} \times \hat{se}_{obs},\ \hat{\beta}_{obs} + q^*_{0.975} \times \hat{se}_{obs}\right]$$

I computed the bootstrapped p-values as

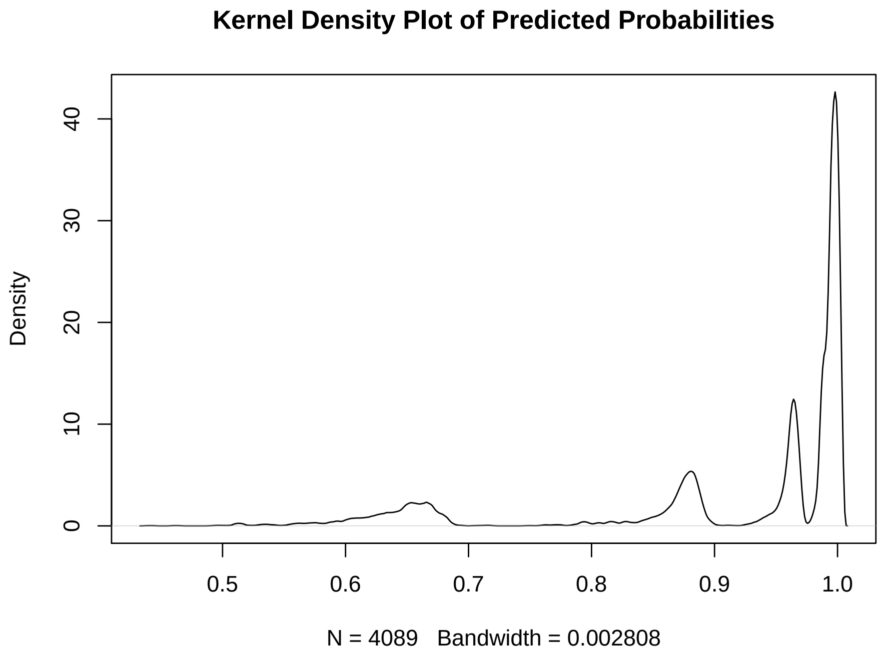
$$p^{boot} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}[t^* \geq t_{obs}]$$

where M is the number of bootstrap replications.

Table 2: **Bootstrap Statistics for the Logistic Model of** $s = 1-$`dmissing`

	Coef.	CI.lower	CI.upper	p-val
Const.	1.755	1.157	2.471	0.000
S_age	1.333	1.142	1.609	0.000
S_HHpeople	-0.067	-0.112	-0.020	0.001
$\log(\text{inc}+1)$	-0.119	-0.216	-0.042	0.001

(c) I plot the kernel density estimate of the predicted probabilities of reporting data, $\hat{\mu}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})$, using an Epanechnikov kernel with `R`'s unbiased cross-validation bandwidth.



2 Semiparametric GMM with missing data

2.1 An optimal instrument

We have the conditional moment condition:

$$\mathbb{E}[m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)|t_i, \boldsymbol{x}_i] = 0$$

By LIE

$$\mathbb{E}\big[\boldsymbol{g}(t_i, \boldsymbol{x}_i)\mathbb{E}[m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)|t_i, \boldsymbol{x}_i]\big] = 0$$

for any $\boldsymbol{g}(\cdot)$. Thus,

$$\begin{aligned} &\mathbb{E}\big[\mathbb{E}[\boldsymbol{g}(t_i, \boldsymbol{x}_i)m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)|t_i, \boldsymbol{x}_i]\big] = 0 \\ \implies &\mathbb{E}[\boldsymbol{g}(t_i, \boldsymbol{x}_i)m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)] = 0 \end{aligned}$$

Now, we want to find the optimal $\boldsymbol{g}$ that minimizes AsyVar$(\hat{\boldsymbol{\beta}})$; call it $\boldsymbol{g}_0$. Let $\boldsymbol{z}_i = (t_i, \boldsymbol{x}_i)$, $\boldsymbol{w}_i = (y_i^*, t_i, \boldsymbol{x}_i)$. The GMM estimator is

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta \in \mathbb{R}^d} \left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta})\right)' \boldsymbol{W} \left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta})\right)$$

The FOC wrt $\boldsymbol{\beta}$ is

$$0 = \left(\frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial \beta}\boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta})\right)' \boldsymbol{W} \left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta})\right)$$

We can write the FOC plugging in $\hat{\boldsymbol{\beta}}$:

$$0 = \left(\frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \beta}\boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \hat{\boldsymbol{\beta}})\right)' \boldsymbol{W} \left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \hat{\boldsymbol{\beta}})\right).$$

Then, if we take a mean value Taylor expansion of the last term in parentheses around the true parameter $\boldsymbol{\beta}_0$ and rearrange in the usual way, we get

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) = -\left[\boldsymbol{G}_n(\hat{\boldsymbol{\beta}})' \boldsymbol{W} \boldsymbol{G}_n(\tilde{\boldsymbol{\beta}})\right]^{-1} \boldsymbol{G}_n(\hat{\boldsymbol{\beta}})' \boldsymbol{W} \left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n} \boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta}_0)\right)$$

where

$$\boldsymbol{G}_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \beta}\boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta})$$

Now we just need to let things converge via LLN and CLT to get

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \to_d \mathcal{N}\left(0, [\boldsymbol{G}(\boldsymbol{\beta}_0)' \boldsymbol{W} \boldsymbol{G}(\boldsymbol{\beta}_0)]^{-1} \boldsymbol{G}(\boldsymbol{\beta}_0)' \boldsymbol{W} \mathbb{V}[\boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta}_0)] \boldsymbol{W} \boldsymbol{G}(\boldsymbol{\beta}_0 [\boldsymbol{G}(\boldsymbol{\beta}_0)' \boldsymbol{W} \boldsymbol{G}(\boldsymbol{\beta}_0)]^{-1}\right)$$

Then with the optimal weight matrix $\boldsymbol{W} = \mathbb{V}[\boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta}_0)]^{-1}$ the asymptotic variance collapses to

$$\boldsymbol{V}_0 = \left[\boldsymbol{G}(\boldsymbol{\beta}_0)' \mathbb{V}[\boldsymbol{g}(\boldsymbol{z}_i)m(\boldsymbol{w}_i, \boldsymbol{\beta}_0)]^{-1} \boldsymbol{G}(\boldsymbol{\beta}_0)\right]^{-1}$$

And, as shown in class, this leads to the optimal instrument

$$\boldsymbol{g}_0(\boldsymbol{z}_i) = \mathbb{E}\left[\frac{\partial}{\partial \beta} m(\boldsymbol{w}_i, \boldsymbol{\beta}) | \boldsymbol{z}_i\right] \mathbb{V}[m(\boldsymbol{w}_i, \boldsymbol{\beta}_0) | \boldsymbol{z}_i]^{-1}$$

Now, in the given case, since y_i is Bernoulli we know that

$$\mathbb{V}[m(\boldsymbol{w}_i, \boldsymbol{\beta}_0) | \boldsymbol{z}_i] = F(t_i\theta_0 + \boldsymbol{x}_i'\boldsymbol{\gamma}_0)[1 - F(t_i\theta_0 + \boldsymbol{x}_i'\boldsymbol{\gamma}_0)]$$

And

$$\mathbb{E}\left[\frac{\partial}{\partial \beta} m(\boldsymbol{w}_i, \boldsymbol{\beta}) | \boldsymbol{z}_i\right] = f(t_i\theta_0 + \boldsymbol{x}_i'\boldsymbol{\gamma}_0)[t_i, \boldsymbol{x}_i]',$$

which gives the desired result. When $F(\cdot)$ is the logistic cdf we know that

$$f(u) = F(u)[1 - F(u)],$$

so that

$$\boldsymbol{g}_0(t_i, \boldsymbol{x}_i) = [t_i, \boldsymbol{x}_i]'.$$

2.2 Missing completely at random

(a) The optimal unconditional moment condition is:

$$\mathbb{E}[\boldsymbol{g}_0(t_i, \boldsymbol{x}_i)m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)] = 0$$

Now, $y_i = s_i y_i^*$. Thus,

$$\mathbb{E}[\boldsymbol{g}_0(t_i, \boldsymbol{x}_i)m(y_i, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)|s_i = 1] = \mathbb{E}[\boldsymbol{g}_0(t_i, \boldsymbol{x}_i)m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)|s_i = 1]$$

And, with the MCAR assumption, the above moment condition is identical to the optimal one. Accordingly, a feasible estimator is simply

$$\hat{\boldsymbol{\beta}}_{\texttt{MCAR,feasible}} = \arg\min_{\theta,\gamma} \frac{1}{n}\sum_{i=1}^{n}[s_i \boldsymbol{g}_0(t_i, \boldsymbol{x}_i)(y_i - F(t_i\theta + \boldsymbol{x}_i'\boldsymbol{\gamma}))].$$

(b) I report the the results from using the feasible estimator below.

Table 3: **Logit Results under MCAR Assumption**

	Estimate	Std.Error	t	p-value	CI.lower	CI.upper
dpisofirme	-0.325	0.105	-3.106	0.032	-0.518	-0.167
S_age	-0.226	0.024	-9.326	0.000	-0.275	-0.177
S_HHpeople	0.027	0.018	1.563	0.140	-0.006	0.061
log_inc	0.023	0.017	1.341	0.184	-0.010	0.060

2.3 Missing at random

(a) Start with the optimal moment condition

$$\mathbb{E}[\boldsymbol{g}_0(t_i, \boldsymbol{x}_i)m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)] = 0.$$

Since $y_i = s_i * y_i^*$ and $s_i \perp y_i^*|(t_i, x_i)$,

$$\mathbb{E}[s_i m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)|t_i, x_i] = \mathbb{E}[\boldsymbol{g}(t_i, \boldsymbol{x}_i)m(y_i^*, t_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)]$$

Thus, $\mathbb{E}[s_i m(y_i^*, t_i, x_i; \beta_0)|t_i, x_i] = 0$ is equivalent to the optimal unconditional moment restriction.

(b) Here the problem is that we do not have a functional form for the propensity score $p(x_i, t_i)$. Accordingly, even if we have consistent estimator of β it might not be efficient. We could provide a functional form for the probability of not missing data $p(x_i, t_i)$ as a probit or logit estimate. Therefore, it would be a two-step approach in with the first step we estimate $p(x_i, t_i)$ and then plug it in our moment conditions. This are to be used to estimate $\hat{\beta}_{\text{MAR}}$ using GMM. This estimator would be consistent but may not be efficient, so $\hat{\beta}_{\text{MAR}}$ and $\tilde{\beta}_{\text{MAR}}$.

(c)

Table 4: **Logit Results under MAR Assumption**

	Estimates	Std. Error	t	p-value	CI.lower	CI.upper
dpisofirme	-0.322	0.096	-3.368	0.020	-0.491	-0.161
S_age	-0.224	0.024	-9.248	0.000	-0.270	-0.174
S_HHpeople	0.029	0.018	1.651	0.108	-0.006	0.061
log_inc	0.021	0.018	1.188	0.241	-0.014	0.056

(d) Trimming does not change the results.

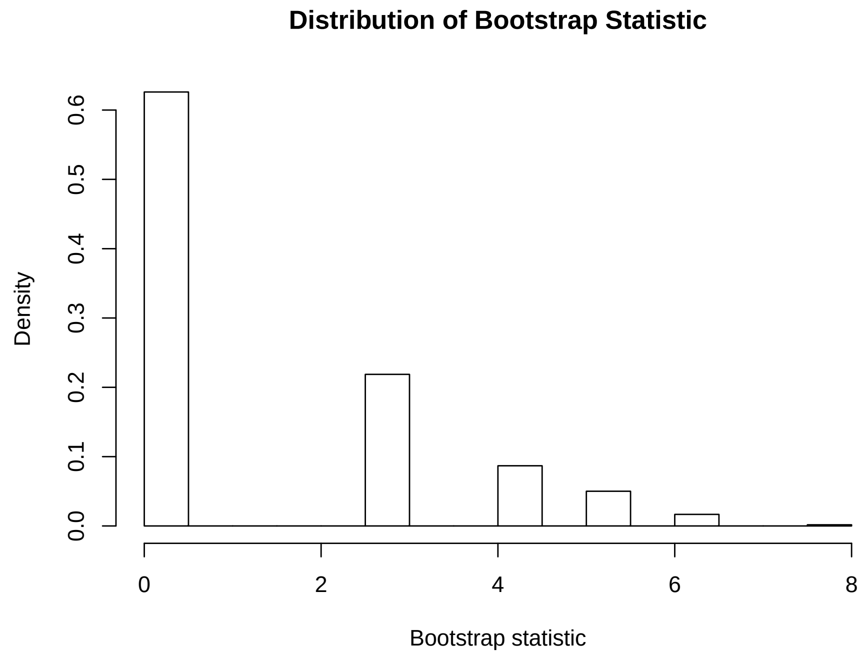
Table 5: **Logit Results under MAR Assumption with Trimming**

	Estimates	Std. Error	t	p-value	CI.lower	CI.upper
dpisofirme	-0.322	0.096	-3.368	0.020	-0.491	-0.161
S_age	-0.224	0.024	-9.248	0.000	-0.270	-0.174
S_HHpeople	0.029	0.018	1.651	0.108	-0.006	0.061
log_inc	0.021	0.018	1.188	0.241	-0.014	0.056

3 When bootstrap fails

3.1 Nonparametric bootstrap fail

I plot the empirical distribution of the bootstrap statistic, $n(\max\{x_i\} - \max\{x_i^*\})$, below. Clearly, the empirical distribution does not coincide with the theoretical Exponential (1) distribution.



3.2 Parametric bootstrap to the rescue

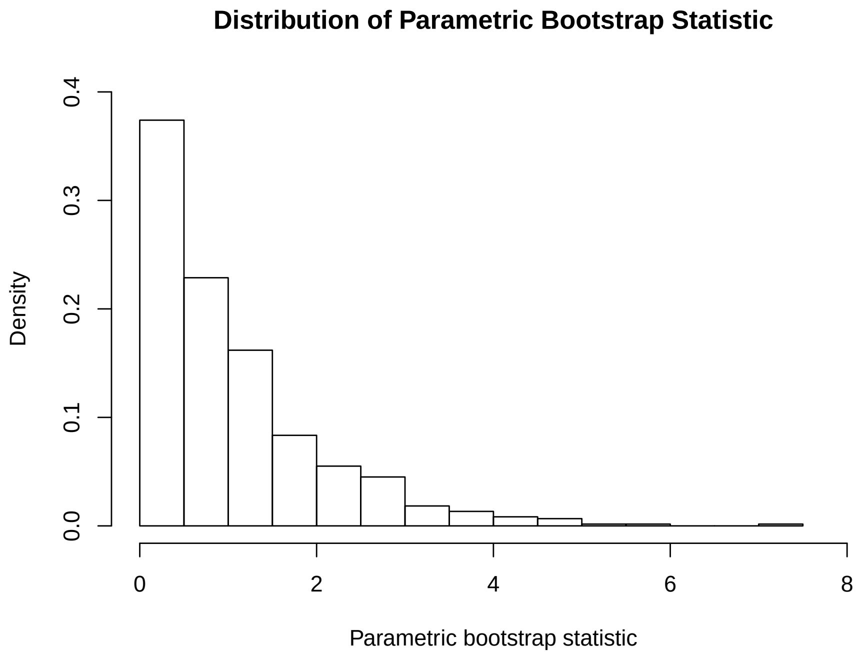
Now, consider the parametric bootstrap statistic, $t_{\mathtt{P}}^* = n(\max\{x_i\} - \max\{x_i^*\})$, where

$$x_i^* \sim_{iid} \texttt{Uniform}[0, \max\{x_i\}].$$

I plot the empirical distribution of $t_{\mathtt{P}}^*$ below. Now, the empirical distribution *does* seem to coincide with the theoretical Exponential (1) distribution.

3.3 Intuition

In the nonparametric case, the bootstrap statistic has a mass point at zero since $\Pr[\max\{x_i\} = \max\{x_i^*\}]$ converges to 1. However, the parametric bootstrap corrects for this "bias": in this case $\Pr[\max\{x_i\} = \max\{x_i^*\}] = 0$, since $x_i^* \sim_{iid} \texttt{Uniform}[0, \max\{x_i\}]$.



4 Appendix

4.1 R code

4.1.1 Question 1

```
## ECON675: ASSIGNMENT 3
## Q1: NLS
## Anirudh Yadav
## 10/24/2018

#######################################################################
# Load packages, clear workspace
#######################################################################
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
options(scipen = 999)       #forces R to use normal numbers instead of scientific notation


#######################################################################
# Input data, create additional covariates
#######################################################################

# Get Piso Firme data
data <- as.data.table(read.csv('PhD_Coursework/ECON675/HW3/pisofirme.csv'))

# Create dependent variable for logistic regression
data[,s:= 1-dmissing]

# Create income regressor
```

```
data[,log_inc:= log(S_incomepc+1)]

######################################################################
# Q9(a): Estimate logistic regression
######################################################################

# Estimate model
mylogit <- glm(s ~ S_age + S_HHpeople + log_inc, data = data, family = "binomial")

b.hat   <- as.data.table(mylogit["coefficients"])

# Get robust standard errors
V.hat       <- vcovHC(mylogit, type = "HC1")
se.hat      <- as.data.table(sqrt(diag(V.hat)))

# Compute t-stats
t.stats       <- b.hat/se.hat

# Compute p-vals
n = nrow(data)
d = 4
p = round(2*pt(abs(t.stats[[1]]),df=n-d,lower.tail=FALSE),3)

# Compute CIs
CI.lower = b.hat - qnorm(0.975)*se.hat
CI.upper = b.hat + qnorm(0.975)*se.hat

# Mash results together
results  = as.data.frame(cbind(b.hat,se.hat,t.stats,p,CI.lower,CI.upper))
colnames(results) = c("Coef.","Std. Err.","t-stat","p-val","CI.lower","CI.upper")
rownames(results) = c("Const.", "S_age","S_HHpeople","log_inc")

# Get latex table output
xtable(results,digits=3)


######################################################################
# Q9(b): Bootstrap statistics
######################################################################

# Define function for bootstrap statistic
boot.logit <- function(data, i){
  logit  <- glm(s ~ S_age + S_HHpeople + log_inc,
             data = data[i, ], family = "binomial")
  V      <- vcovHC(logit, type = "HC1")
  se     <- sqrt(diag(V.hat))
  t.boot <- (coef(logit)-coef(mylogit))/se

  return(t.boot)
}

# Run bootstrap replications
set.seed(123)
boot.results <- boot(data = data, R = 999, statistic = boot.logit)

# Get 0.025/0.975 quantiles from the boot t-distribution
boot.q      <- sapply(1:4, function (i) quantile(boot.results$t[,i], c(0.025, 0.975)))

# Construct 95% CIs using bootstrapped quantiles
boot.ci.lower = b.hat + t(boot.q)[,1]*se.hat
boot.ci.upper = b.hat + t(boot.q)[,2]*se.hat

# Get p-val -- I'm not sure if this is right!!!
boot.p = sapply(1:4,function(i) 1/999*sum(boot.results$t[,i]>=t.stats[i]))

# Tabulate bootstrap results
results.b  = as.data.frame(cbind(b.hat,boot.ci.lower,boot.ci.upper,boot.p))
colnames(results.b) = c("Coef.","CI.lower","CI.upper","p-val")
```

```
rownames(results.b) = c("Const.", "S_age","S_HHpeople","log_inc")

# Get latex table output
xtable(results.b,digits=3)


######################################################################
# Q9(c): Predicted probabilities
######################################################################

b.hat = coef(mylogit)

# Subset data
X     = data[,.(S_age,S_HHpeople,log_inc)]
X[,const:= 1]
setcolorder(X,c("const","S_age","S_HHpeople","log_inc"))

# Define logistic cdf (i.e. mu function)
mu = function(u){(1+exp(-u))^(-1)}

# Construct vector of x_i'*beta.hats
XB = as.matrix(X)%*%b.hat

# Compute predicted probabilities
mu.hat = mu(XB)

X[,mu.hat:=mu.hat]

#Make plot
plot(density(mu.hat,kernel="e",bw="ucv",na.rm=TRUE),main="Kernel Density Plot of Predicted Probabilities")
```

4.1.2 Question 2

```
## ECON675: ASSIGNMENT 3
## Q2: SEMIPARAMETRIC GMM W MISSING DATA
## Anirudh Yadav
## 10/26/2018

######################################################################
# Load packages, clear workspace
######################################################################
rm(list = ls())             #clear workspace
library(foreach)            #for looping
library(data.table)         #for data manipulation
library(dplyr)              #melting for ggplot
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(gmm)
options(scipen = 999)       #forces R to use normal numbers instead of scientific notation

######################################################################
# Input data, create additional covariates
######################################################################

# Get Piso Firme data
pisofirme <- read.csv('PhD_Coursework/ECON675/HW3/pisofirme.csv')
complete  <- complete.cases(pisofirme[, 5:27])
pisofirme <- pisofirme[complete, ]

# s_i: non-missing indicator
pisofirme$nmissing <- 1 - pisofirme$dmissing
```

```
######################################################################
# Q2: Missing completely at random
######################################################################

# GMM moment condition: logistic
g_logistic <- function(theta, data) {
  a <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$dpisofirme
  b <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$S_age
  c <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$S_HHpeople
  d <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    log(1+data$S_incomepc)
  cbind(a, b, c, d)
}

# logistic bootstrap
boot.T_logistic <- function(boot.data, ind) {
  gmm(g_logistic, boot.data[ind, ], t0=c(0,0,0,0), wmatrix="ident", vcov="iid")$coef
}

ptm <- proc.time()
set.seed(123)
temp <- boot(data=pisofirme[pisofirme$nmissing==1, ], R=499, statistic = boot.T_logistic, stype = "i")
proc.time() - ptm
table3 <- matrix(NA, ncol=6, nrow=4)
for (i in 1:4) {
  table3[i, 1] <- temp$t0[i]
  table3[i, 2] <- sd(temp$t[, i])
  table3[i, 3] <- table3[i, 1] / table3[i, 2]
  table3[i, 4] <- 2 * max( mean(temp$t[, i]-temp$t0[i]>=abs(temp$t0[i])), mean(temp$t[, i]-temp$t0[i]<=-1*abs(temp$t0[i])) )
  table3[i, 5] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.975)
  table3[i, 6] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.025)
}

rownames(table3)=c("dpisofirme", "S_age","S_HHpeople","log_inc")
colnames(table3)=c("Estimate", "Std.Error", "t", "p-value", "CI.lower","CI.upper")

######################################################################
# Q3(c): Missing at random
######################################################################
# GMM moment condition
g_MAR <- function(theta, data) {
  data <- data[data$nmissing==1, ]
  a <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$dpisofirme * data$weights
  b <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$S_age * data$weights
  c <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$S_HHpeople * data$weights
  d <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    log(1+data$S_incomepc) * data$weights
  cbind(a, b, c, d)
}

# logistic bootstrap
boot.T_MAR <- function(boot.data, ind) {
  data.temp <- boot.data[ind, ]
  fitted <- glm(nmissing ~ dpisofirme + S_age + S_HHpeople +I(log(S_incomepc+1)) - 1,
                data = data.temp,
                family = binomial(link = "logit"))$fitted
  data.temp$weights <- 1 / fitted
  gmm(g_MAR, data.temp, t0=c(0,0,0,0), wmatrix="ident", vcov="iid")$coef
}
```

```
ptm <- proc.time()
set.seed(123)
temp <- boot(data=pisofirme, R=499, statistic = boot.T_MAR, stype = "i")
proc.time() - ptm
table5 <- matrix(NA, ncol=6, nrow=4)
for (i in 1:4) {
  table5[i, 1] <- temp$t0[i]
  table5[i, 2] <- sd(temp$t[, i])
  table5[i, 3] <- table5[i, 1] / table5[i, 2]
  table5[i, 4] <- 2 * max( mean(temp$t[, i]-temp$t0[i]>=abs(temp$t0[i])), mean(temp$t[, i]-temp$t0[i]<=-1*abs(temp$t0[i])) )
  table5[i, 5] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.975)
  table5[i, 6] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.025)
}

######################################################################
# Q3(d): Trimming
######################################################################
g_MAR2 <- function(theta, data) {
  data <- data[data$nmissing==1 & data$weights<=1/0.1, ]
  a <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$dpisofirme * data$weights
  b <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$S_age * data$weights
  c <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    data$S_HHpeople * data$weights
  d <- (data$danemia - plogis(theta[1]*data$dpisofirme + theta[2]*data$S_age + theta[3]*data$S_HHpeople + theta[4]*log(1+data$S_inc
    log(1+data$S_incomepc) * data$weights
  cbind(a, b, c, d)
}

# logistic bootstrap
boot.T_MAR2 <- function(boot.data, ind) {
  data.temp <- boot.data[ind, ]
  fitted <- glm(nmissing ~ dpisofirme + S_age + S_HHpeople +I(log(S_incomepc+1)) - 1,
                data = data.temp,
                family = binomial(link = "logit"))$fitted
  data.temp$weights <- 1 / fitted
  gmm(g_MAR2, data.temp, t0=c(0,0,0,0), wmatrix="ident", vcov="iid")$coef
}

ptm <- proc.time()
set.seed(123)
temp <- boot(data=pisofirme, R=499, statistic = boot.T_MAR2, stype = "i")
proc.time() - ptm
table6 <- matrix(NA, ncol=6, nrow=4)
for (i in 1:4) {
  table6[i, 1] <- temp$t0[i]
  table6[i, 2] <- sd(temp$t[, i])
  table6[i, 3] <- table6[i, 1] / table6[i, 2]
  table6[i, 4] <- 2 * max( mean(temp$t[, i]-temp$t0[i]>=abs(temp$t0[i])), mean(temp$t[, i]-temp$t0[i]<=-1*abs(temp$t0[i])) )
  table6[i, 5] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.975)
  table6[i, 6] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.025)
}
```

4.1.3 Question 3

```
## ECON675: ASSIGNMENT 3
## Q3: WHEN BOOTSTRAP FAILS
## Anirudh Yadav
## 10/26/2018

######################################################################
# Load packages, clear workspace
######################################################################
rm(list = ls())             #clear workspace
```

```
library(foreach)           #for looping
library(data.table)        #for data manipulation
library(dplyr)             #melting for ggplot
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
options(scipen = 999)      #forces R to use normal numbers instead of scientific notation


######################################################################
# Q1: Nonparametric bootstrap fail
######################################################################
set.seed(123)

N = 1000

# Simulate runif data
X = runif(N,0,1)

# Get max
x.max.obs = max(X)

# Write function for bootrap statistic
boot.stat = function(data, i){
  N*(x.max.obs-max(data[i]))
}

# Run bootsrap with 599 replications
boot.results = boot(data = X, R = 599, statistic = boot.stat)

# Make frequency plot
h          = hist(boot.results$t,plot=FALSE)
h$density = h$counts/sum(h$counts)
plot(h,freq=FALSE,main="Distribution of Bootstrap Statistic",xlab="Bootstrap statistic")


######################################################################
# Q2: Parametric bootstrap fail
######################################################################

# Generate parametric bootstrap samples
X.boot = replicate(599,runif(N,0,x.max.obs))

# Compute maximums for each replications
x.max.boot = sapply(1:599,function(i) max(X.boot[,i]))

# Compute bootstrap statistic
t.boot     = N*(x.max.obs-x.max.boot)

# Make frequency plot
h2         = hist(t.boot,plot=FALSE)
h2$density = h2$counts/sum(h2$counts)
plot(h2,freq=FALSE,main="Distribution of Parametric Bootstrap Statistic",xlab="Parametric bootstrap statistic",ylim=c(0,0.4),xlim=c
```

4.2 STATA code

4.2.1 Question 1

```
********************************************************************************
* ECON675: ASSIGNMENT 3
* Q1: NLS
* Anirudh Yadav
* 10/24/2018
********************************************************************************

********************************************************************************
* Preliminaries
********************************************************************************
clear all
set more off

* Set working directory
global dir "/Users/Anirudh/Desktop/GitHub"


********************************************************************************
* Import data, create additional covariates
********************************************************************************

* Import LaLonde data
import delimited using "$dir/PhD_Coursework/ECON675/HW3/pisofirme.csv"

* Generate additional variables
gen      s = 1-dmissing
gen log_inc = log(s_incomepc+1)


********************************************************************************
* Q9(a): Estimate logistic regression
********************************************************************************
logit s s_age s_hhpeople log_inc, robust


********************************************************************************
* Q9(b): Bootstrap statistics
********************************************************************************
logit s s_age s_hhpeople log_inc, vce(bootstrap, reps(999))


********************************************************************************
* Q9(b): Predicted probabilities
********************************************************************************
logit s s_age s_hhpeople log_inc, vce(robust)

* predict propensity score
predict p

* plot kernel density estimates
twoway histogram p || kdensity p, k(gaussian) || ///
 kdensity p, k(epanechnikov) || kdensity p, k(triangle) ///
 leg(lab(1 "Propensity Score") lab(2 "Gaussian") ///
 lab(3 "Epanechnikov") lab(4 "Triangle"))
```

4.2.2 Question 2

```
********************************************************************************
* ECON675: ASSIGNMENT 3
* Q1: SEMIPARAMETRIC GMM
* Anirudh Yadav
* 10/24/2018
********************************************************************************
```

```
********************************************************************************
* Preliminaries
********************************************************************************
clear all
set more off
set matsize 10000

* Set working directory
global dir "/Users/Anirudh/Desktop/GitHub"

import delimited using "$dir/PhD_Coursework/ECON675/HW3/pisofirme.csv"

* Part 2b
gen x3=log(S_incomepc+1)
gmm (dpisofirme*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age +{b2}*S_HHpeople+{b3}*x3))) (S_age*(danemia-invlogit({t1}*dpisofirme+{

* Part 3c
gen s = 1-dmissing

*Propensity Score Estimation
logit s dpisofirme S_age S_HHpeople x3, nocons
predict phat

*Generating the Instrumental Varibles
gen inst1=dpisofirme/(1-phat)
gen inst2=S_age/(1-phat)
gen inst3=S_HHpeople/(1-phat)
gen inst4=x3/(1-phat)

*GMM Estimation
gmm (s*inst1*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age +{b2}*S_HHpeople+{b3}*x3))) (s*inst2*(danemia-invlogit({t1} *dpisofirme+{
+{b2}*S_HHpeople+{b3}*x3))) (s*inst4*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age ///
+{b2}*S_HHpeople+{b3}*x3))), instruments(inst1 inst2 inst3 inst4)
vce(boot, reps(1000)) winit(i)

*Part 3d
*Trimming the data
drop if (1-phat)<0.1
gmm (s*inst1*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age ///
+{b2}*S_HHpeople+{b3}*x3))) (s*inst2*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age ///
+{b2}*S_HHpeople+{b3}*x3))) (s*inst3*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age ///
+{b2}*S_HHpeople+{b3}*x3))) (s*inst4*(danemia-invlogit({t1}*dpisofirme+{b1}*S_age ///
+{b2}*S_HHpeople+{b3}*x3))), instruments(inst1 inst2 inst3 inst4) vce(boot, reps(1000)) winit(i)
```

4.2.3 Question 3

```
* Q3.1 - nonparametric bootstrap
clear all

* generate sample
set seed 123
set obs 1000
gen X = runiform()

* save actual max
sum X
local maxX=r(max)

* run nonparametric bootstrap of max
bootstrap stat=r(max), reps(599) saving(nonpar_results, replace): summarize X

* load results
use nonpar_results, clear
```

```
* generate statistic
gen nonpar_stat = 1000*(`maxX'-stat)

* plot
hist nonpar_stat, ///
 plot(function exponential = 1-exponential(1,x), range(0 5) color(red))
graph export q3_1_S.png, replace


* Q3.2 - parametric bootstrap
clear all

tempname memhold
tempfile para_results

* generate sample
set seed 123
set obs 1000
gen X = runiform()

* save actual max
sum X
local maxX=r(max)

* parametric bootstrap
postfile `memhold' max using `para_results'
forvalues i = 1/599{
capture drop sample
gen sample = runiform(0,`maxX')
sum sample
post `memhold' (r(max))
}
postclose `memhold'

* load results
use `para_results', clear

* generate statistic
gen para_stat = 1000*(`maxX'-max)

* plot
hist para_stat, ///
 plot(function exponential = 1-exponential(1,x), range(0 5) color(red))
graph export q3_2_S.png, replace
```