# Econ 675 Assignment 3

Nathan Mather*

November 4, 2019

## Contents



## 1 Question 1: Non-linear Least Squares

### 1.1 Q1 Part 1

The general non-linear least squares estimator is

$$\hat{\boldsymbol{\beta}}_n = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu(\boldsymbol{x_i'}\boldsymbol{\beta}))^2$$

*Shouts out to Ani, Paul, Tyler, Erin, Caitlin and others for all the help with this

Now for $\boldsymbol{\beta}_0 = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathrm{E}[(y_i - \mu(\boldsymbol{x_i'}\boldsymbol{\beta}))^2]$ to be identifiable we need:

$$\boldsymbol{\beta}_0 = \boldsymbol{\beta}_0^*$$

$$\iff \boldsymbol{\beta}_0^* = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathrm{E}[(y_i - \mu(\boldsymbol{x_i'}\boldsymbol{\beta}))^2]$$

To find this start by noting that

$$\mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2] = \mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) + \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2]$$

$$= \mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2] + \mathrm{E}[(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2] + 2\mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))]$$

$$= \mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2] + \mathrm{E}[(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2]$$

The last equality comes from the last term being zero by iterated expectations. I show this below.

$$\mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))] = \mathrm{E}[\mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))]|\boldsymbol{x}_i]$$

$$= \mathrm{E}[(\mathrm{E}[y_i|\boldsymbol{x}_i] - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))] = 0$$

Using this fact we have that

$$\mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2] = \mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2] + \mathrm{E}[(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2] \geq \mathrm{E}[(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0))^2]$$

$$\forall \boldsymbol{\beta} \neq \boldsymbol{\beta}_0$$

This is strictly greater than unless $\exists \boldsymbol{\beta} \neq \boldsymbol{\beta}_0$ such that $\mathrm{E}[(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2] = 0$ Thus this give us an identification condition that $\mathrm{E}[(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2] \neq 0 \ \forall \boldsymbol{\beta} \neq \boldsymbol{\beta}_0$. This means that $\boldsymbol{\beta}_0$ is the unique minimizer of $\mathrm{E}[(y_i - \mu(\boldsymbol{x_i'}\boldsymbol{\beta}))^2]$
Next note that if $\mu(\cdot)$ is a linear function, $\boldsymbol{\beta}_0$ is the coefficient of the best linear predictor and has the usual closed form $\boldsymbol{\beta}_0 = \mathrm{E}[\boldsymbol{x}_i\boldsymbol{x}_i']^{-1}\mathrm{E}[\boldsymbol{x}_i y_i]$

### 1.2 Q1 Part 2

In order to set this up as a Z estimator lets take a first order condition. This gives use the following condition.

$$\mathrm{E}[(\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta})\boldsymbol{x}_i] = 0$$

Now take the sample analog and let $m(\boldsymbol{z}_i, \boldsymbol{\beta}) = (y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta})\boldsymbol{x}_i$ where $\boldsymbol{z}_i = (y_i, \boldsymbol{x}_i')'$. We can write $\hat{\boldsymbol{\beta}}_n$ as the Z-estimator that solves:

$$0 = \frac{1}{n}\sum_{i=1}^{n} m(\boldsymbol{z}_i, \hat{\boldsymbol{\beta}}_n)$$

Now assuming $\hat{\boldsymbol{\beta}}_n \to \boldsymbol{\beta}_0$ and regularity conditions we get the standard M estimation result.

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

Were

$$\boldsymbol{H}_0 = \mathrm{E}\left[\frac{\partial}{\partial \boldsymbol{\beta}} m(\boldsymbol{z}_i, \boldsymbol{\beta}_0)\right] = \mathrm{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0)^2 \boldsymbol{x_i}\boldsymbol{x_i'}]$$

and

$$\boldsymbol{\Sigma_0} = \mathrm{V}[m(\boldsymbol{z}_i, \boldsymbol{\beta}_0)] = \mathrm{E}[\sigma^2(\boldsymbol{x_i})\dot{\mu}(\boldsymbol{x_i'}\boldsymbol{\beta_o})^2 \boldsymbol{x_i}\boldsymbol{x_i'}]$$

## 1.3 Q1 Part 3

$$\hat{\mathrm{V}}_n^{HC} = \left(\frac{1}{n}\sum_{i=1}^{n} \hat{\boldsymbol{m}}\hat{\boldsymbol{m}}'\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \hat{\boldsymbol{m}}\hat{\boldsymbol{m}}'\hat{e}_i^2\right) \left(\frac{1}{n}\sum_{i=1}^{n} \hat{\boldsymbol{m}}\hat{\boldsymbol{m}}'\right)^{-1}$$

where $\hat{\boldsymbol{m}} = \boldsymbol{m_\beta(z_i, \hat{\beta})}$ and $hate = y_i - \boldsymbol{m(z_i, \hat{\beta})}$
Now by the delta method and letting $g(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\| = \sum_{k=1}^{d} \boldsymbol{\beta}_k^2$ we get

$$\sqrt{n}(g(\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}) - g(\boldsymbol{\beta}_0)) \rightarrow_d \mathcal{N}(0, \dot{g}(\boldsymbol{\beta}_0)\boldsymbol{V}_0\dot{g}(\boldsymbol{\beta}_0)')$$

where $\dot{g}(\boldsymbol{\beta}_0) = \frac{d}{d\boldsymbol{\beta}'}g(\boldsymbol{\beta}) = 2\boldsymbol{\beta}'$ Hence the confidence interval is given by

$$CI_{0.95} = \left[\|\hat{\boldsymbol{\beta}}_n\|^2 - 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}{}'\hat{\boldsymbol{V}}\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}}{n}}, \|\hat{\boldsymbol{\beta}}_n\|^2 + 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}{}'\hat{\boldsymbol{V}}\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}}{n}}\right]$$

## 1.4 Q1 Part 4

In this case we get $\boldsymbol{\Sigma_0} = \sigma^2 \boldsymbol{H}_0$ and the asymptotic variance reduces to

$$\boldsymbol{V}_0 = \sigma^2 \boldsymbol{H}_0^{-1} = \sigma^2 \mathrm{E}[\dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0)^2 \boldsymbol{x_i}\boldsymbol{x_i'}]^{-1}$$

We can estimate variane using $\hat{\boldsymbol{V}} = \hat{\sigma}^2 \hat{\boldsymbol{H}}^{-1}$ where

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

and

$$\hat{\boldsymbol{H}} = \frac{a}{n}\sum_{i=1}^{n} \dot{\mu}(\boldsymbol{x}_i'\boldsymbol{\beta}_0)^2 \boldsymbol{x_i}\boldsymbol{x_i'}$$

Which is consistent by the continuous mapping theorem. Now by the delta method and letting $g(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\| = \sum_{k=1}^{d} \boldsymbol{\beta}_k^2$ we get

$$\sqrt{n}(g(\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}) - g(\boldsymbol{\beta}_0)) \rightarrow_d \mathcal{N}(0, \dot{g}(\boldsymbol{\beta}_0)\boldsymbol{V}_0\dot{g}(\boldsymbol{\beta}_0)')$$

where $\dot{g}(\boldsymbol{\beta}_0) = \frac{d}{d\boldsymbol{\beta}'}g(\boldsymbol{\beta}) = 2\boldsymbol{\beta}'$ Hence the confidence interval is given by

$$CI_{0.95} = \left[\|\hat{\boldsymbol{\beta}}_n\|^2 - 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}{}'\hat{\boldsymbol{V}}\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}}{n}}, \|\hat{\boldsymbol{\beta}}_n\|^2 + 1.96\sqrt{\frac{4\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}{}'\hat{\boldsymbol{V}}\hat{\boldsymbol{\beta}}_{\boldsymbol{n}}}{n}}\right]$$

## 1.5 Q1 Part 5

The conditional likelihod funciton is

$$f_{y|x}(y_i|\boldsymbol{x}_i) = \frac{1}{(2\pi)^{n/2}\sigma^2} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}))^2\right)$$

with log likelihood

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

This gives us the following first order conditions

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

$$\frac{\partial}{\partial \sigma^2}\ell_n(\boldsymbol{\beta}, \sigma^2) = \frac{1}{2\hat{\sigma}^4_{ML}}\sum_{i=1}^{n}(y_i - \mu(\boldsymbol{x}_i'\boldsymbol{\beta}_{ML}))^2 - \frac{n}{2\hat{\sigma}^2_{ML}} = 0$$

These conditions are equivalent to those found above.

## 1.6 Q1 Part 6

If the link funciton is unknown, $\boldsymbol{\beta}_0$ is not identified. To see this, consider two pairs of parameters $(\mu(\cdot), \boldsymbol{\beta}_0)$ and $(\tilde{\mu}(\cdot), \tilde{\boldsymbol{\beta}}_0)$ where $\tilde{\mu}(z) = \mu(z/k)$ and $\tilde{\boldsymbol{\beta}}_0 = k\boldsymbol{\beta}_0$ for some $k \neq 0$. Then the parameters are clearly different, but $(\mu(\cdot), \boldsymbol{\beta}_0) = (\tilde{\mu}(\cdot), \tilde{\boldsymbol{\beta}}_0)$. A common normalization is $\|\boldsymbol{\beta}_0\| = 1$, but more conditions are needed to regain identification.

## 1.7 Q1 Part 7

The link function is

$$\mu(\boldsymbol{x}_i'\boldsymbol{\beta}_0) = \mathrm{E}[y_i|\boldsymbol{x}_i] = \mathrm{E}[\mathbb{1}(\boldsymbol{x}_i'\boldsymbol{\beta}_0 \geq \epsilon_i)|\boldsymbol{x}_i] = Pr[\boldsymbol{x}_i'\boldsymbol{\beta}_0 \geq \epsilon_i|\boldsymbol{x}_i] = F(\boldsymbol{x}_i'\boldsymbol{\beta}_0) = \frac{1}{1 + exp(-\boldsymbol{x}_i'\boldsymbol{\beta}_0)}$$

The conditional variance of $y_i$ is

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

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 this implies

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

Now to derie the asymptotic variance we note that for the logistic CDF: $\dot{\mu}(u) = (1 - \mu(u))\mu(u)$.THis gives us and asymptotic variance of

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

Where

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

and

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

## 1.8 Q1 Part 8

MLE gives the same point estimator as NLS but MLE is more efficient and so it has lower variance.

## 1.9 Q1 Part 9

(a)

R table

| term | estimate | std.error | statistic | p.value | CI_L | CI_H |
|---|---|---|---|---|---|---|
| (Intercept) | 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_inc | -0.119 | 0.044 | -2.707 | 0.007 | -0.205 | -0.033 |

stata table

| | (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|---|
| | s | | | | |
| VARIABLES | coef | se | tstat | pval | ci |
| s | . | . | . | . | . - . |
| S_age | 1.333 | 0.123 | 10.832 | 0.000 | 1.092 - 1.575 |
| S_HHpeople | -0.067 | 0.023 | -2.872 | 0.004 | -0.112 - -0.021 |
| log_inc | -0.119 | 0.044 | -2.708 | 0.007 | -0.205 - -0.033 |
| Constant | 1.755 | 0.334 | 5.247 | 0.000 | 1.099 - 2.411 |

(b)

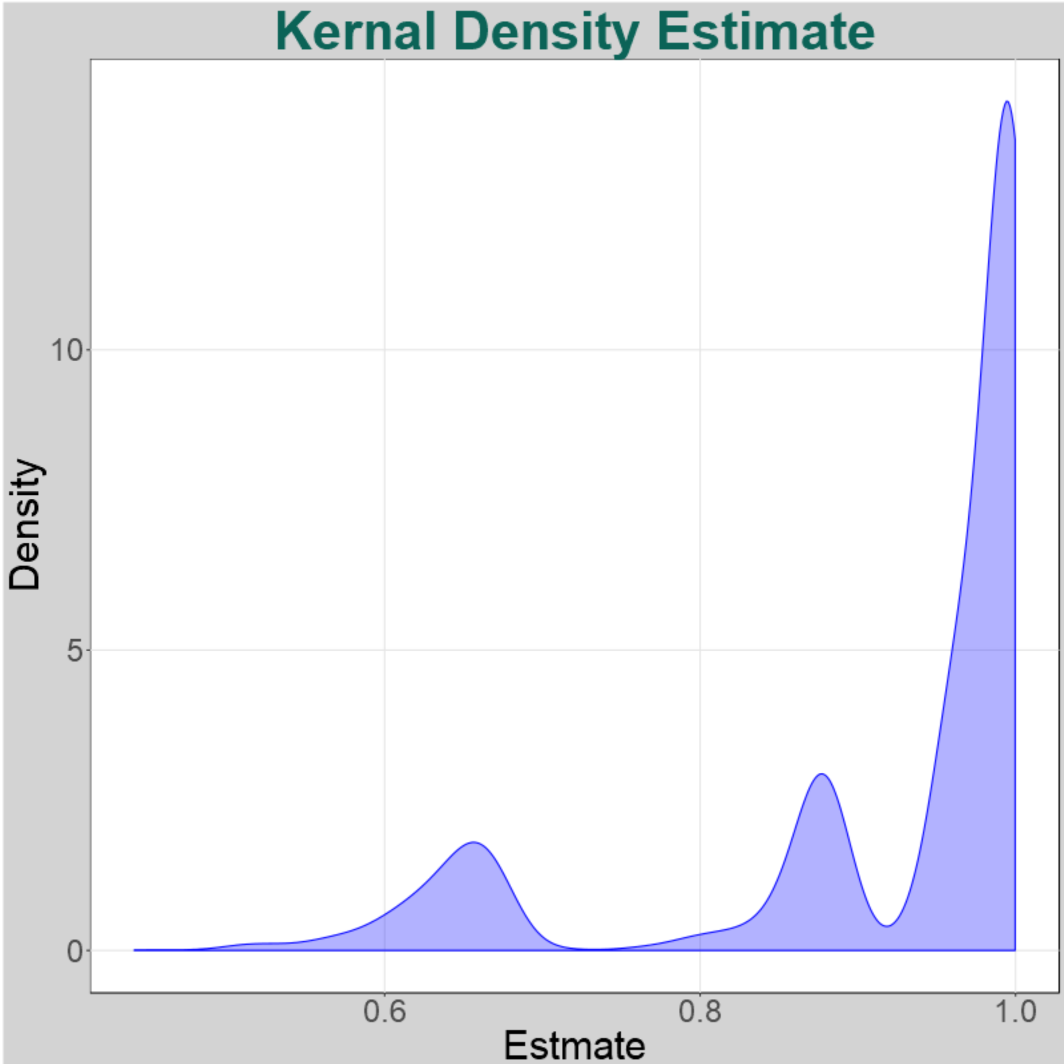
R table

| term | estimate | std.error | t_q.975 | t_q.025 | CI_L | CI_H | pvalue |
|---|---|---|---|---|---|---|---|
| (Intercept) | 1.755 | 0.335 | 2.167 | -1.774 | 1.161 | 2.480 | 0.000 |
| S_HHpeople | -0.067 | 0.023 | 2.040 | -1.958 | -0.112 | -0.019 | 0.000 |
| S_age | 1.333 | 0.123 | 2.210 | -1.558 | 1.142 | 1.606 | 0.000 |
| log_inc | -0.119 | 0.044 | 1.817 | -2.155 | -0.213 | -0.039 | 0.000 |

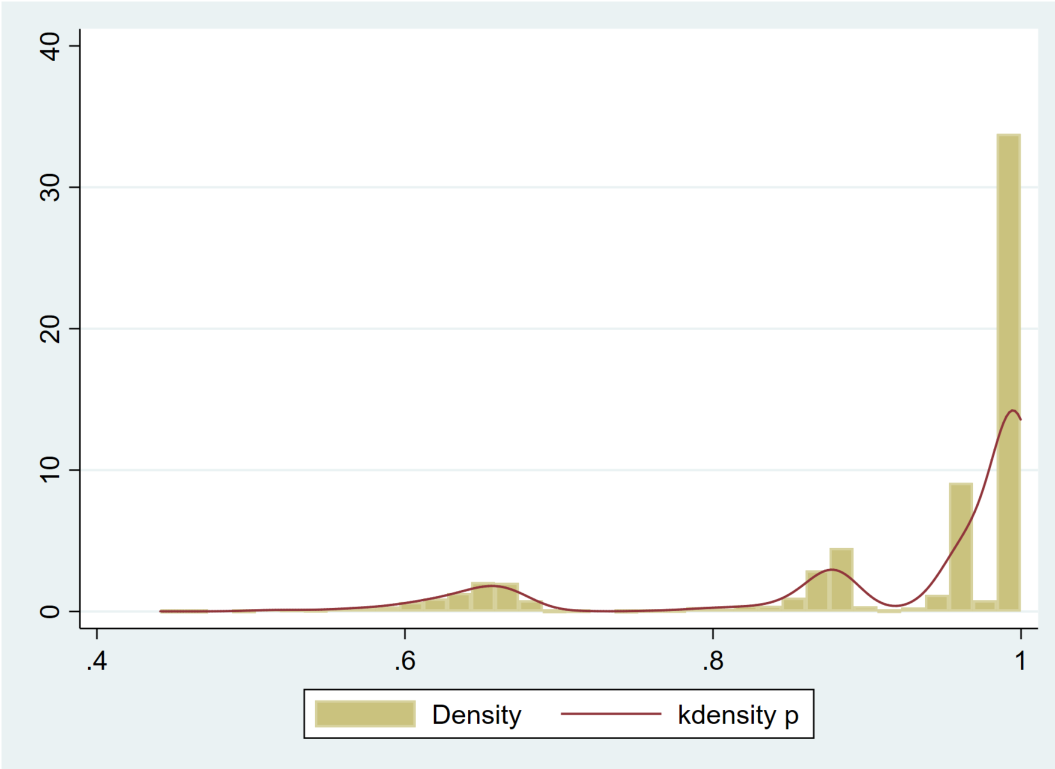
Stata table

| | (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|---|
| | s | | | | |
| VARIABLES | coef | se | tstat | pval | ci |
| s | . | . | . | . | . - . |
| S_age | 1.333 | 0.124 | 10.740 | 0.000 | 1.090 - 1.577 |
| S_HHpeople | -0.067 | 0.023 | -2.870 | 0.004 | -0.112 - -0.021 |
| log_inc | -0.119 | 0.046 | -2.580 | 0.010 | -0.209 - -0.029 |
| Constant | 1.755 | 0.350 | 5.014 | 0.000 | 1.069 - 2.441 |

(c) R Graph




Stata Graph




## 2 Question 2: Semiparametric GMM with Missing Data

### 2.1 Q2 part 1

Start with the moment condition we are given

$$0 = \mathbb{E}[m(y_i^*, t_i, x_i; \beta_0)|t_i, x_i]$$

Now by law of iterated expectations we can multiply by a function g and still get zero

$$0 = \mathbb{E}[g(t_i, x_i)\mathbb{E}[m(y_i^*, t_i, x_i; \beta_0)|t_i, x_i]] \text{ for any } g(t_i, x_i)$$

Next we can put g inside the first expectation

$$0 = \mathbb{E}[\mathbb{E}[g(t_i, x_i)m(y_i^*, t_i, x_i; \beta_0)|t_i, x_i]]$$

now removing the inside expectation

$$0 = \mathbb{E}[g(t_i, x_i)m(y_i^*, t_i, x_i; \beta_0)]$$

We want to find $g_0$, the optimal $g$ that minimizes AsyVar$(\hat{\beta})$. Let $z_i = (t_i, x_i)$, $w_i = (y_i^*, t_i, x_i)$, and $\theta = \beta$.

The first thing we need to do is determine the asymptotic variance V associated with

$$\hat{\theta} = \arg\min(\frac{1}{n}\sum_i g(z_i)m(w_i, \theta))'W(\frac{1}{n}\sum_i g(z_i)m(w_i, \theta))$$

Taking first order conditions and setting equal to zero we get

$$\text{FOC: } 0 = [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\theta)]' W [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\theta)]$$

or

$$0 = [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\hat{\theta})]' W [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\hat{\theta})]$$

Since it is euqual to zero we can add another of the same term and multiply by $(\hat{\theta} - \theta_0)$ giving

$$0 = [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\theta_0)]' W [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\theta_0)] + [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\hat{\theta})]' W [\frac{1}{n}\sum_i \frac{\partial}{\partial\theta} g(z_i)m(w_i,\hat{\theta})](\hat{\theta} - \theta_0)$$

Which can be rearanged to give

$$\sqrt{n}(\hat{\theta} - \theta_0) = (\Omega_0' W_0 \Omega_0)^{-1} \Omega_0 W_0 \frac{1}{\sqrt{n}} \sum_i g(z_i)m(w_i,\theta) + o_p(1)$$

And then By the CLT, $\sqrt{n}(\hat{\theta} - \theta_0) \to_d N(0, V_0)$

where $V_0 = (\Omega_0' W_0 \Omega_0)^{-1} \Omega_0' W_0 \Sigma_0 W_0 \Omega_0 (\Omega_0' W_0 \Omega_0)^{-1}$

and where $\Sigma_0 = \mathbb{V}[g(z_i)m(w_i,\theta)]$
setting values Optimally to minimize V gives us the following conditions.

$$W_0^* = \Sigma_0^{-1} \text{ and } V_0^* = \Omega_0' \Sigma_0 \Omega_0$$

$$g^*(z_i) = \frac{\partial m_i}{\partial\theta} \mathbb{V}[m(w_i,\theta_0)|z_i]^{-1}$$

Now applying this specifically to a probit model gives

$$\mathbb{V}[m(y_i^*, t_i, x_i, \beta_0)|t_i, x_i] = F(t_i \cdot \theta_0 + x_i\gamma_0)(1 - F(t_i \cdot \theta_0 + x_i'\gamma_0))$$

$$\mathbb{E}[\frac{\partial}{\partial\beta} m(y_i^*, t_i, x_i, \beta_0)|t_i, x_i] = \mathbb{E}[f(t_i \cdot \theta_0 + x_i\gamma_0)(t_i, x_i)|t_i, x_i] = f(t_i \cdot \theta_0 + x_i\gamma_0)[t_i, x_i']'$$

$$\text{Therefore, } g_0(t_i, x_i) = \frac{f(t_i \cdot \theta_0 + x_i\gamma_0)}{F(t_i \cdot \theta_0 + x_i\gamma_0)(1 - F(t_i \cdot \theta_0 + x_i'\gamma_0))}[t_i, x_i']'$$

If $F$ is the logistic cdf we instead get

$$F(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \frac{\partial}{\partial x} F(x) = \frac{-e^{-x}}{(1 + e^{-x})^2} = -e^{-x} F(x)^2$$

$$\frac{f(x)}{F(x)(1 - F(x))} = \frac{-e^{-x} F(x)^2}{F(x)(1 - F(x))} = \frac{-e^{-x} F(x)}{1 - F(x)} = 1$$

$$g_0(t_i, x_i) = [t_i, x_i']'$$

## 2.2 Q2 part 2

(a) The optimal unconditional moment condition is:

$$0 = \mathbb{E}[g(t_i, x_i)m(y_i^*, t_i, x_i; \beta_0)]$$

In part 2.1 we showed that, setting $g = g_0$ this is equivalent to:

$$0 = \mathbb{E}[m(y_i^*, t_i, x_i; \beta_0)|t_i, x_i]$$

Since $s_i \perp (y_i^*, t_i, x_i)$:

$$\begin{aligned} 0 &= \mathbb{E}[m(y_i^*, t_i, x_i; \beta_0)|t_i, x_i] \\ 0 &= \mathbb{E}[g_0(t_i, x_i)m(y_i^*, t_i, x_i; \beta_0)] \\ 0 &= \mathbb{E}[g_0(t_i, x_i)m(y_i^*, t_i, x_i; \beta_0)|s_i = 1] \end{aligned}$$

Thus, $\hat{\beta}_{MCAR}$ solving $0 \approx \hat{\mathbb{E}}[g_0(t_i, x_i)m(y_i, t_i, x_i; \hat{\beta}_{MCAR})|s_i = 1]$ is consistent for $\beta_0$
$\hat{\beta}_{MCAR,feasible}$ solves $0 \approx \hat{\mathbb{E}}[\hat{g}(t_i, x_i)m(y_i, t_i, x_i; \hat{\beta}_{MCAR})|s_i = 1]$

(b) The tables for this section are below, they are different because the programs estimate logits differently

R table

| term | Estimate | sd | p_value | t | CI_L | CI_H |
|---|---|---|---|---|---|---|
| dpisofirme | -0.317 | 0.073 | 0.008 | -4.363 | -0.453 | -0.187 |
| S_age | -0.244 | 0.020 | 0.000 | -11.975 | -0.284 | -0.205 |
| S_HHpeople | 0.024 | 0.013 | 0.104 | 1.775 | -0.002 | 0.049 |
| log_inc | 0.033 | 0.014 | 0.020 | 2.397 | 0.006 | 0.058 |

stata table

| | coef | se | tstat | CI low | CI high |
|---|---|---|---|---|---|
| dpisofirme | -.316 | .067 | -4.721 | -.447 | -.185 |
| S age | -.246 | .02 | -12.367 | -.285 | -.207 |
| S HHpeople | .021 | .013 | 1.685 | -.003 | .046 |
| log inc | .035 | .012 | 2.835 | .011 | .059 |

## 2.3 Q2 part 3

(a) With the MAR assumption we get the same results. First by MAR we can split out the conditional moment restriction and then calculate the conditional variance restriction.

$$\mathrm{E}[s_i \cdot m(y_i^*, t_i, x_i)|t_i, x_i] = \mathrm{E}[s_i|t_i, x_i] \cdot \mathrm{E}[m(y_i^*, t_i, x_i)|t_i, x_i] = 0$$

and the conditional variance follows

$$\mathrm{V}[s_i \cdot m(y_i^*, t_i, x_i)|t_i.x_i] = \mathrm{E}[s_i^2 \cdot m(y_i^*, t_i, x_i)^2|t_i, x_i] = \mathrm{E}[s_i|t_i, x_i] \cdot F(t_i \cdot \theta_0 + x_i'\gamma_0)F(t_i \cdot \theta_0 + x_i'\gamma_0)$$

This along with the FOC gives use the same result in part 1
(b) We can estimate $\tilde{\beta}_{MAR}$ by estimating $P_0$ and plugging it in. This will give us a consistent estimator.
(c)

R table

| Term | Estimate | Std.Error | t | p-value | CI_L | CI_U |
|---|---|---|---|---|---|---|
| dpisofirme | -0.317 | 0.073 | -4.363 | 0.008 | -0.453 | -0.187 |
| S_age | -0.244 | 0.020 | -11.975 | 0.000 | -0.284 | -0.205 |
| S_HHpeople | 0.024 | 0.013 | 1.775 | 0.104 | -0.002 | 0.049 |
| log_inc | 0.033 | 0.014 | 2.397 | 0.020 | 0.006 | 0.058 |

Stata table

| | coef | se | tstat | CI low | CI high |
|---|---|---|---|---|---|
| dpisofirme | -.308 | .068 | -4.52 | -.442 | -.175 |
| S age | -.245 | .019 | -12.762 | -.283 | -.208 |
| S HHpeople | .022 | .012 | 1.8 | -.002 | .045 |
| log inc | .034 | .012 | 2.759 | .01 | .058 |

(d)

R table

| Term | Estimate | Std.Error | t | p-value | CI_L | CI_U |
|---|---|---|---|---|---|---|
| dpisofirme | -0.317 | 0.073 | -4.363 | 0.008 | -0.453 | -0.187 |
| S_age | -0.244 | 0.020 | -11.975 | 0.000 | -0.284 | -0.205 |
| S_HHpeople | 0.024 | 0.013 | 1.775 | 0.104 | -0.002 | 0.049 |
| log_inc | 0.033 | 0.014 | 2.397 | 0.020 | 0.006 | 0.058 |

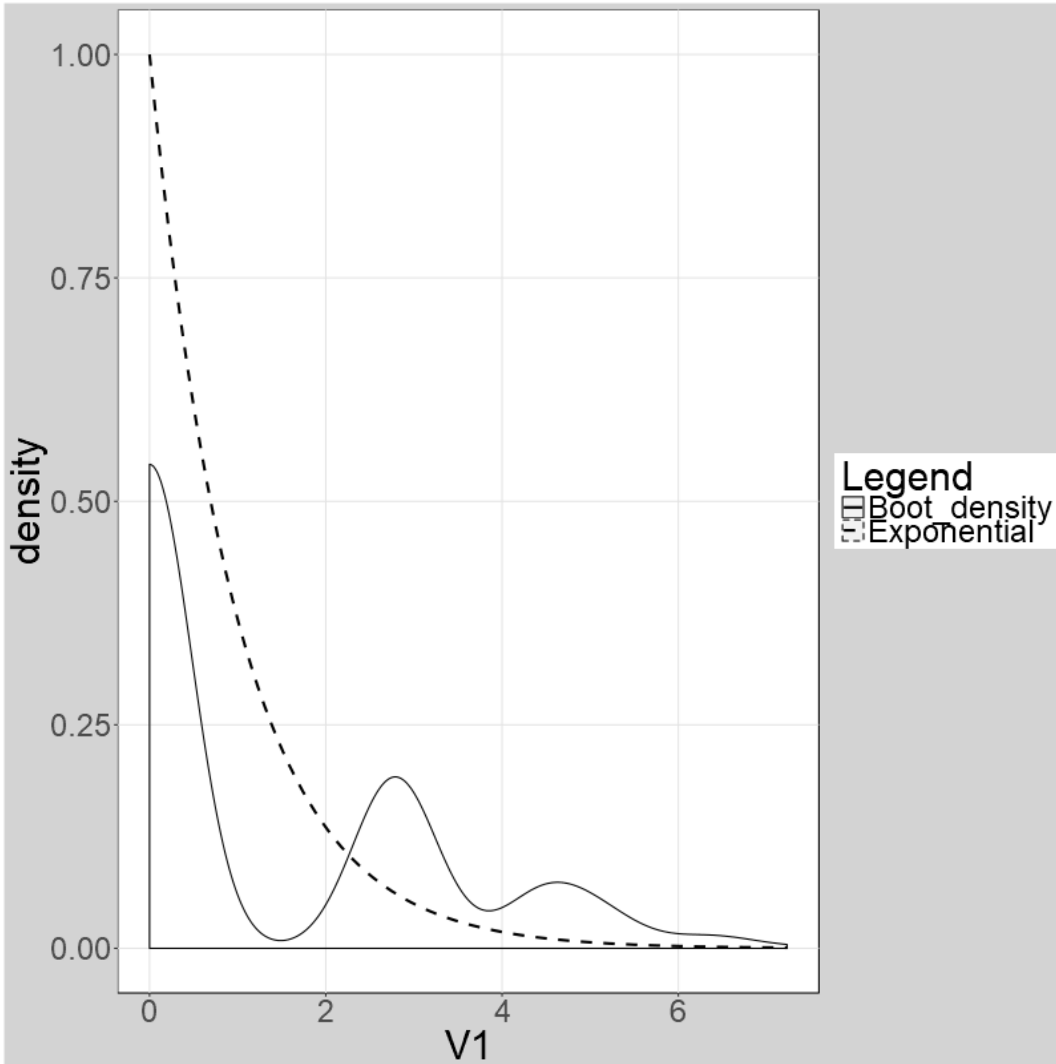
Stata table

| | coef | se | tstat | CI low | CI high |
|---|---|---|---|---|---|
| dpisofirme | -.308 | .066 | -4.681 | -.437 | -.179 |
| S age | -.245 | .02 | -12.37 | -.284 | -.207 |
| S HHpeople | .022 | .015 | 1.487 | -.007 | .05 |
| log inc | .034 | .017 | 2.018 | .001 | .067 |

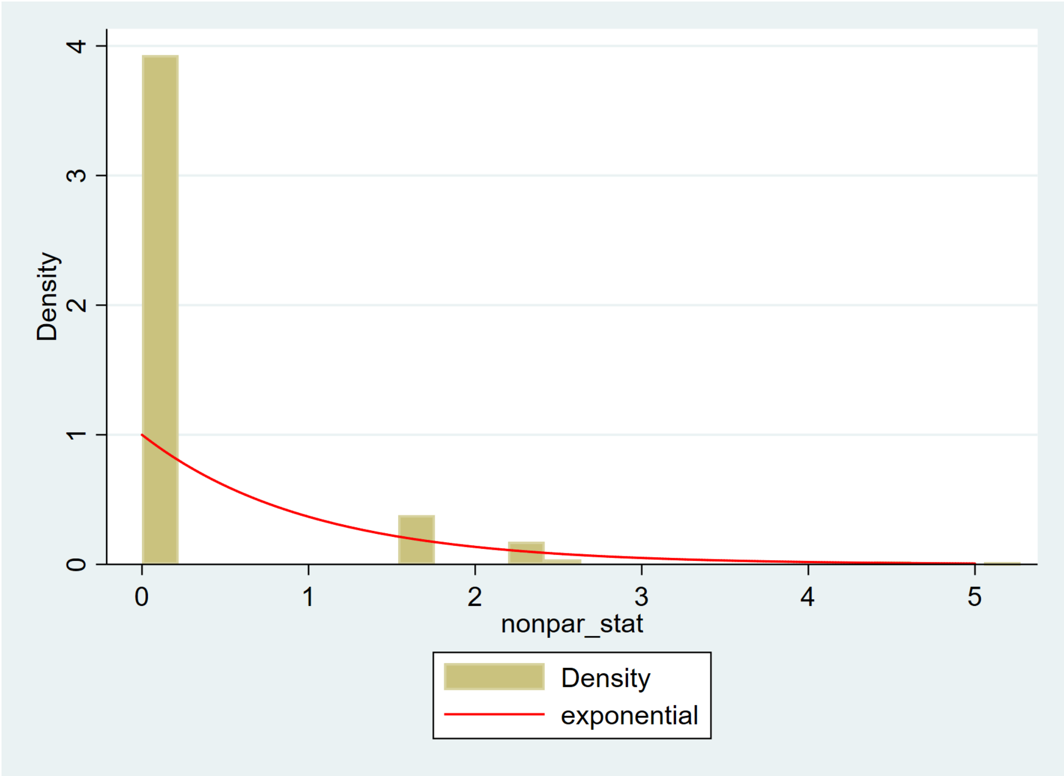
# 3 Question 3: When Bootstrap Fails

## 3.1 Q3 part 1

R Graph

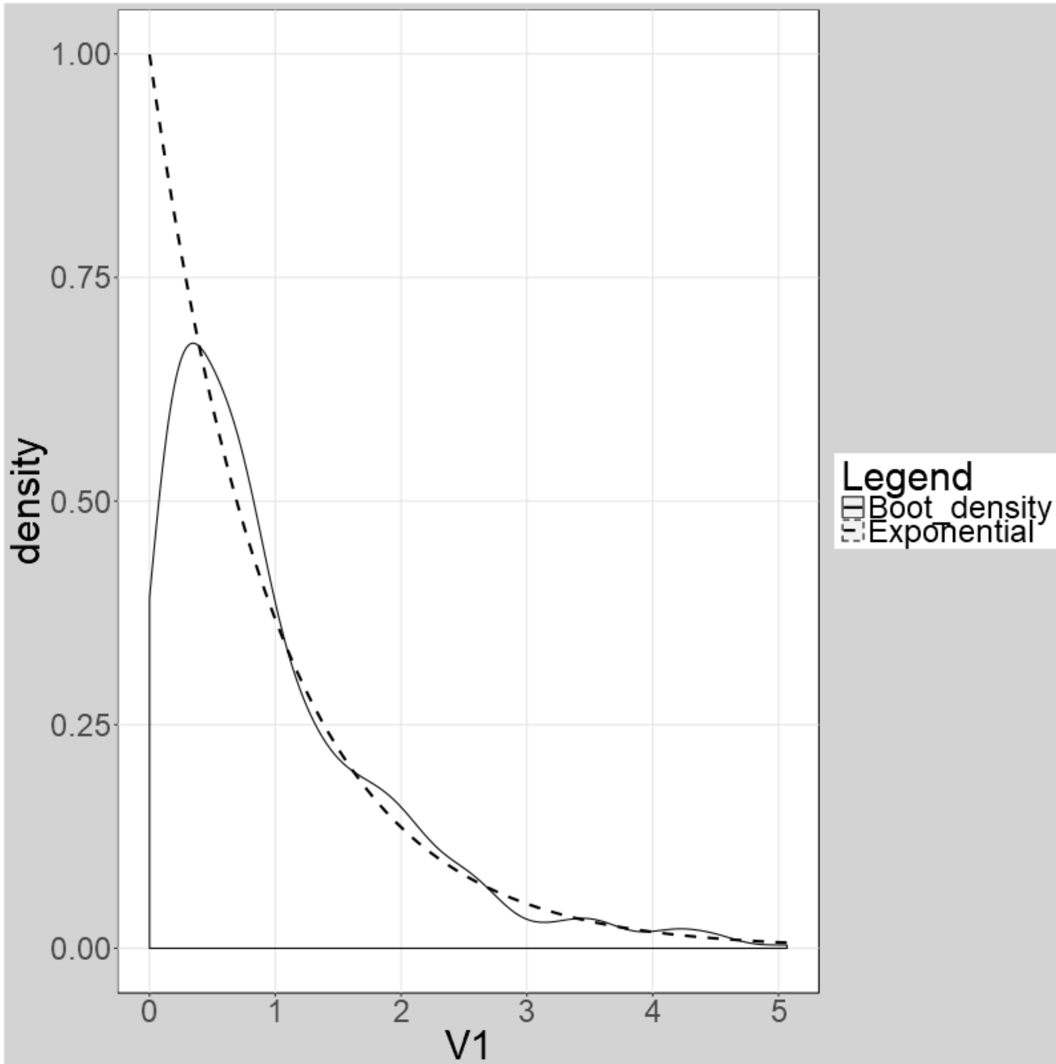

density
1.00
0.75
0.50
0.25
0.00
0
2
4
6
V1
Legend
Boot_density
Exponential

Stata Graph

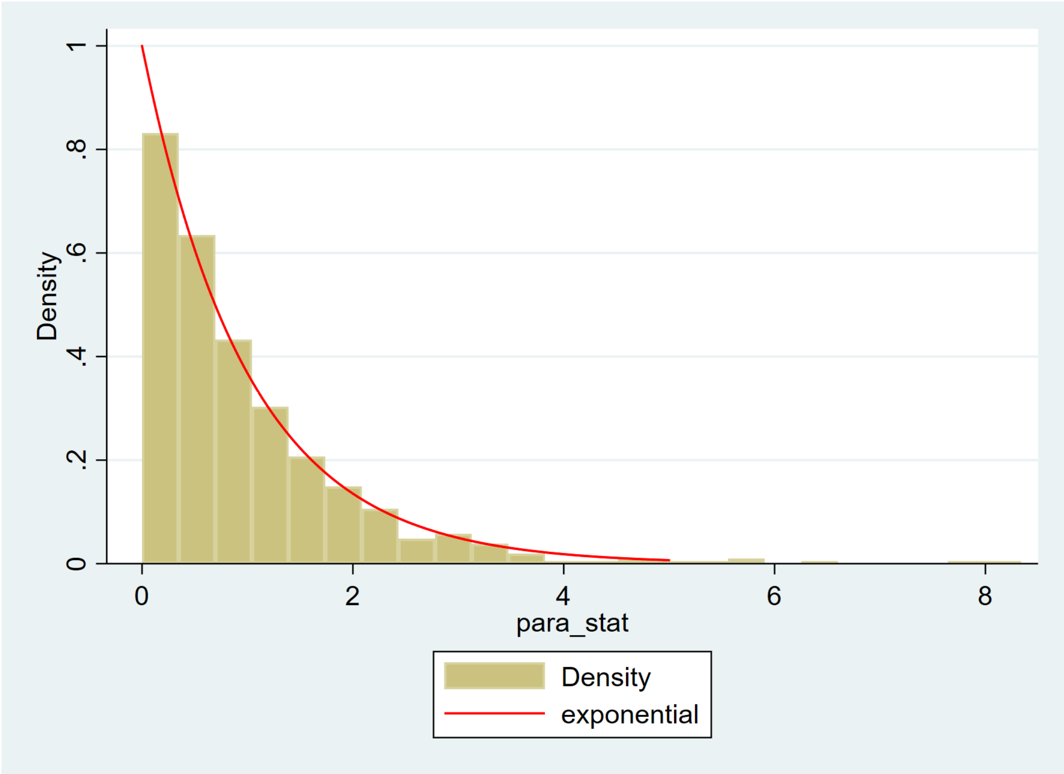



### 3.2 Q3 part 2

R Graph


density
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5
V1
Legend
Boot_density
Exponential

Stata Graph




### 3.3 Q3 part 3

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

# 4 Appendix

## 4.1 R Code

# pset 2 Labor

```r
#==========================#
# ==== Metrics 675 ps 3 ====
#==========================#



#========================================#
# ==== load packages and clear data ====
#========================================#

library(data.table)
library(doParallel)
library(foreach)
library(ggplot2)
library(Matrix)
library(sandwich)
library(lmtest)
library(xtable)
library(boot)
library(gmm)
library(broom)

# clear data and consol
rm(list = ls(pos = ".GlobalEnv"), pos = ".GlobalEnv")
options(scipen = 999)
cat("\f")

# set options
opt_test_run <- FALSE

# set attributes for plot to default ea theme
plot_attributes <- theme( plot.background = element_rect(fill = "lightgrey"),
                          panel.grid.major.x = element_line(color = "gray90"),
                          panel.grid.minor  = element_blank(),
                          panel.background = element_rect(fill = "white", colour = "black") ,
                          panel.grid.major.y = element_line(color = "gray90"),
                          text = element_text(size= 30),
                          plot.title = element_text(vjust=0,
                                                    hjust = 0.5,
                                                    colour = "#0B6357",
                                                    face = "bold",
                                                    size = 40))


#=====================#
# ==== Question 1 ====
#=====================#

#=================#
# ==== part 9 a ====
```

```
#==================#


# load in data
p_dt <- fread("c:/Users/Nmath_000/Documents/MI_school/Second Year/675 Applied Econometrics/hw/hw3/pisof

# create si variable
p_dt[,s := 1-dmissing]


# create log variable
p_dt[, log_inc := log(S_incomepc + 1)]

# estimate the logic model
reg <- glm(s ~ S_age + S_HHpeople + log_inc,
           data = p_dt,
           family = binomial(link = 'logit'))


# get robust standard errors
reg_c_r <- coeftest(reg, vcov = vcovHC(reg, type="HC1"))

# tidy that up
reg_c_r <- data.table(tidy(reg_c_r))


# make a copy for tex results
table_q1_9_a <- copy(reg_c_r)

# make confidence interval for part a
table_q1_9_a[,CI_L := estimate - qnorm(.975)*std.error]
table_q1_9_a[,CI_H := estimate + qnorm(.975)*std.error]


#======================#
# ==== Q1 part 9 b ====
#======================#


# right a funciton to bootstrap
b_st1 <- function(in_data, sample, in_reg_c_r){

  # run regression
  b_reg <- glm(s ~ S_age + S_HHpeople + log_inc,
             data = in_data[sample],
             family = binomial(link = 'logit'))


  # get robust standard errors
  b_reg_r <- coeftest(b_reg, vcov = vcovHC(b_reg, type="HC1"))

  # tidy that up
  b_reg_r <- data.table(tidy(b_reg_r))
```

```r
  # calculate t stat
  t_stat <- (b_reg_r[, estimate] - in_reg_c_r[, estimate])/in_reg_c_r[, std.error]

  # get the stuff I want
  return(t_stat)
}

# now run the bootstrap
boot_results <- boot(data = p_dt, R = 999, statistic = b_st1, in_reg_c_r = reg_c_r)

#Now get the stats from every sample
boot_samp <- data.table(boot_results$t)

# create balnk data.table for quantiles
quant_dt <- reg_c_r[, "term"]

quant_dt[, t_q.975 := unlist(lapply(boot_samp,  quantile, .975))]
quant_dt[, t_q.025 := unlist(lapply(boot_samp,  quantile, .025))]


# merge on quantiles to original estimates ans standard errors
table_q1_9_b <- merge(reg_c_r[, c("term", "estimate", "std.error")], quant_dt,  by = "term")

# now get bootstrap confidence intervals
table_q1_9_b[,CI_L := estimate + t_q.025*std.error]
table_q1_9_b[,CI_H := estimate + t_q.975*std.error]

# calculate p_values
p_value_fun <- function(vector, theta){

  mean(abs(vector) > abs(theta))
}

# write p value function
P_value_fun <- function(vector, theta){

  2 * max( mean(vector-theta>=abs(theta)), mean(vector-theta<=-1*abs(theta)))
}


table_q1_9_b[, pvalue := mapply(FUN = P_value_fun,
                                vector = as.list(data.table(boot_results$t)),
                                as.list(boot_results$t0))]

#======================#
# ==== Q1 Part 9 c ====
#======================#

# grab the data from the original regression
ests <- data.table(Estmate = reg$fitted.values)


plot_q1_9_c <-  ggplot(ests, aes(x=Estmate)) +
```

```r
                geom_density(color = "blue", fill = "blue", kernel = 'gaussian',alpha = .3) +
                plot_attributes + ylab("Density") +
                ggtitle("Kernal Density Estimate")


#======================#
# ==== Question 2 ====
#======================#

#==============================#
# ==== question 2 part 2 B ====
#==============================#

# subset data to complete cases
p_dt <- na.omit(p_dt, c("danemia", "dpisofirme", "S_age", "S_HHpeople", "log_inc"))

# to test function
data = p_dt
theta = c(1,2,3,4)


# logistic bootstrap function, need to put it all in one functon for parallel to function.
boot.T_logistic <- function(boot.data, ind) {
  require(data.table)
  g_logistic <- function(theta, data) {
    a <- data[, (danemia -plogis(theta[1]*dpisofirme +
                                   theta[2]*S_age +
                                   theta[3]*S_HHpeople +
                                   theta[4]*log_inc))*dpisofirme]

    b <- data[, (danemia -plogis(theta[1]*dpisofirme +
                                   theta[2]*S_age +
                                   theta[3]*S_HHpeople +
                                   theta[4]*log_inc))*S_age]

    c <- data[, (danemia -plogis(theta[1]*dpisofirme +
                                   theta[2]*S_age +
                                   theta[3]*S_HHpeople +
                                   theta[4]*log_inc))*S_HHpeople]

    d <- data[, (danemia -plogis(theta[1]*dpisofirme +
                                   theta[2]*S_age +
                                   theta[3]*S_HHpeople +
                                   theta[4]*log_inc))*log_inc]

    cbind(a, b, c, d)
  }

  gmm::gmm(g_logistic, boot.data[ind], t0=c(0,0,0,0), wmatrix="ident", vcov="iid")$coef
}

# run bootsrtap in parallel
ptm <- proc.time()
```

```r
set.seed(123)
temp <- boot(data=p_dt[s==1, ],
             R=499,
             statistic = boot.T_logistic,
             stype = "i",
             parallel = "snow",
             ncpus=4 )
proc.time() - ptm

# create data.table of results
table_q2_2_b <- data.table(term =  c("dpisofirme", "S_age", "S_HHpeople", "log_inc"),
                           Estimate = temp$t0,
                           sd = apply(temp$t, 2, sd))

# write p value function
P_value_fun <- function(vector, theta){

  2 * max( mean(vector-theta>=abs(theta)), mean(vector-theta<=-1*abs(theta)))
}

# apply p value function to data
table_q2_2_b[, p_value := mapply(FUN = P_value_fun,
                                 vector = as.list(data.table(temp$t)),
                                 as.list(temp$t0))]

# create t stat
table_q2_2_b[, t := Estimate/sd]

#  CI function
CI_fun <- function(vector, theta, quant){

  2 * theta - quantile(vector, quant)

}

# get quantiles
table_q2_2_b[, CI_L := mapply(FUN = CI_fun,
                              vector = as.list(data.table(temp$t)),
                              theta = as.list(temp$t0), quant = .975)]
table_q2_2_b[, CI_H := mapply(FUN = CI_fun,
                              vector = as.list(data.table(temp$t)),
                              theta = as.list(temp$t0), quant = 0.025)]


#================================#
# ==== question 2 part 3 C ====
#================================#

# GMM moment condition
g_MAR <- function(theta, data) {
  data <- data[data$s==1, ]
  a <- (data$danemia - plogis(theta[1]*data$dpisofirme +
                                theta[2]*data$S_age +
```

```
                                theta[3]*data$S_HHpeople +
                                theta[4]*log(1+data$S_incomepc)))*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$S_incomepc)))*data$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$S_incomepc)))*data$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$S_incomepc)))*log(1+data$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(s ~ dpisofirme + S_age + S_HHpeople +I(log_inc) - 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=p_dt, 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$t0[i]<=
  table5[i, 5] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.975)
  table5[i, 6] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.025)
}

table_q2_3_c <- data.table(cbind(table_q2_2_b$term, table5))
setnames(table_q2_3_c, colnames(table_q2_3_c), c("Term","Estimate", "Std.Error", "t", "p-value", "CI_L"

#=================#
# ==== part d ====
#=================#

# set this up to use data.table and run in parallel so it doesn't take half my life
# logistic bootstrap function, need to put it all in one functon for parallel to work
```

```r
boot.trim <- function(boot.data, ind) {

  require(data.table)

  data.temp <- boot.data[ind, ]

  fitted <- glm(s ~ dpisofirme + S_age + S_HHpeople +I(log_inc) - 1,
                data = data.temp,
                family = binomial(link = "logit"))$fitted
  data.temp[,weights := 1 / fitted]

  # GMM moment condition with trimming
  g_logistic <- function(theta, data) {
    data.temp <- data.temp[s==1 & weights<=1/0.1, ]
    a <- data[, (danemia -plogis(theta[1]*dpisofirme +
                               theta[2]*S_age +
                               theta[3]*S_HHpeople +
                               theta[4]*log_inc))*dpisofirme *weights]

    b <- data[, (danemia -plogis(theta[1]*dpisofirme +
                               theta[2]*S_age +
                               theta[3]*S_HHpeople +
                               theta[4]*log_inc))*S_age*weights]

    c <- data[, (danemia -plogis(theta[1]*dpisofirme +
                               theta[2]*S_age +
                               theta[3]*S_HHpeople +
                               theta[4]*log_inc))*S_HHpeople*weights]

    d <- data[, (danemia -plogis(theta[1]*dpisofirme +
                               theta[2]*S_age +
                               theta[3]*S_HHpeople +
                               theta[4]*log_inc))*log_inc*weights]

    cbind(a, b, c, d)
  }

  gmm::gmm(g_logistic, data.temp, t0=c(0,0,0,0), wmatrix="ident", vcov="iid")$coef
}




ptm <- proc.time()
set.seed(123)
temp <- boot(data=p_dt, R=499, statistic = boot.trim, stype = "i",parallel = "snow", ncpus=4 )
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$t0[i]<=-
  table6[i, 5] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.975)
  table6[i, 6] <- 2 * temp$t0[i] - quantile(temp$t[, i], 0.025)
}


table_q2_3_d <- data.table(cbind(table_q2_2_b$term, table6))
setnames(table_q2_3_d,
         colnames(table_q2_3_d),
         c("Term","Estimate", "Std.Error", "t", "p-value", "CI_L", "CI_U"))

# convert to numeric
to_change <- c("Estimate", "Std.Error", "t", "p-value", "CI_L", "CI_U")
table_q2_3_c[, to_change] <- lapply(table_q2_3_c[,to_change, with = FALSE],  as.numeric)
table_q2_3_d[, to_change] <- lapply(table_q2_3_d[,to_change, with = FALSE],  as.numeric)


#=====================#
# ==== Question 3 ====
#=====================#


#=================#
# ==== Part 1 ====
#=================#

n <- 1000
set.seed(123)

# generate random data
r_dt <- runif(n, 0,1)

# get max of data
r_max <- max(r_dt)

# write functin for bootstrap
b_fun <- function(in_data, sample){

  n*(r_max-max(in_data[sample]))

}

# run bootsrtap
u_b <- boot(data=r_dt, R=499, statistic = b_fun, stype = "i" )

# get distribution of statistic
u_b_d <- data.table(u_b$t)


# plot data
plot_q3_1 <- ggplot(u_b_d) +
  geom_density(aes(x = V1, linetype = "Boot_density") ) +
  stat_function(fun = function(x) dexp(x), size = 1, aes(linetype = 'Exponential')) +
```

```r
  plot_attributes +
  scale_linetype_manual(name = 'Legend',
                        values = c(Boot_density=1,
                                   Exponential=2))


plot_q3_1

#=================#
# ==== part 2 ====
#=================#

# take random data and do parametric bootstrap
# just write loop for bootstrap
stat_list <- vector("list", length = 599)
for(iter in 1:599){

  # generate random uniform data
  r_data_i <- runif(n, 0,r_max)
  stat_list[[iter]] <- n*(r_max-max(r_data_i))
}
param_b <- data.table(unlist(stat_list))

# plot data
plot_q3_2 <- ggplot(param_b) +
  geom_density(aes(x = V1, linetype = "Boot_density") ) +
  stat_function(fun = function(x) dexp(x), size = 1, aes(linetype = 'Exponential')) +
  plot_attributes +
  scale_linetype_manual(name = 'Legend',
                           values = c(Boot_density=1,
                                      Exponential=2))


plot_q3_2


#======================#
# ==== save output ====
#======================#

# grab all tables to save
tab_list <- grep("table_q", ls(), value=TRUE)

# save all the tables by name

for(tab_i in tab_list){

  print(xtable(get(tab_i), type = "latex",
               digits = 3),
        file = paste0("C:/Users/Nmath_000/Documents/Code/courses/econ 675/PS_3_tex/", tab_i, ".tex"),
        include.rownames = FALSE,
        floating = FALSE)
}
```

```r
# save plot

# save plots
plot_list <- grep("plot_q", ls(), value = TRUE)
for(plot_i in plot_list){
png(paste0("c:/Users/Nmath_000/Documents/Code/courses/econ 675/PS_3_tex/",
           plot_i, ".png"), height = 800, width = 800, type = "cairo")
print(get(plot_i))
dev.off()
}
```

### 4.2 STATA Code

```
******************************************
* PS 3 675 metrics
*Nate Mather
*Stata section
******************************************

clear
**************
* Question 1 *
**************
* load data
use "C:\Users\Nmath_000\Documents\MI_school\Second Year\675 Applied Econometrics\hw\hw3\pisofirme.dta",clear

* set wd
cd "C:\Users\Nmath_000\Documents\Code\courses\econ 675\PS_3_tex"

gen s = 1-dmissing
gen log_inc = ln(S_incomepc + 1)

*******
*part1*
*******

logit s S_age S_HHpeople log_inc, vce(robust)

* output for LaTeX
outreg2 using stata_tab_q1_9_a.tex, side stats(coef se tstat pval ci) ///
 noaster noparen nor2 noobs dec(3) replace tex(frag)

 *******
*part 2*
********

* nonparametric bootstrap
logit s S_age S_HHpeople log_inc, vce(bootstrap, reps(999))

* output for LaTeX
outreg2 using stata_table_q1_9_b.tex, side stats(coef se tstat pval ci) ///
 noaster noparen nor2 noobs dec(3) replace  tex(frag)

* Q1.9c - propensity scores
* logit regression, robust standard errors
logit s S_age S_HHpeople log_inc, vce(robust)

* predict propensity score
predict p

* plot histogram, overlay kernel density
twoway histogram p || kdensity p, k(gaussian)


* save
graph export stata_plot_q1_9_c.png, replace



**************************************************
**** Question 2

**************************************************


* gmm, four moment conditions
local vars = "dpisofirme S_age S_HHpeople log_inc"
gmm ((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3}))))*dpisofirme) ///
((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3}))))*S_age) ///
((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3
```

```
}))))*S_HHpeople) ///
((danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3}))))*log_inc), ///
instruments(`vars') winitial(identity) vce(boot)

* output for LaTeX
 mata:
    coef = st_matrix("e(b)")'
    se = st_matrix("e(se)")'

    tstat = coef:/se

    CI_low = coef - 1.96:*se
    CI_high = coef + 1.96:*se

    stats = round((coef,se,tstat,CI_low,CI_high),.001)

    st_matrix("stats",stats)
end
mat rownames stats = `vars'
mat colnames stats = coef se tstat CI_low CI_high
outtable using stata_table_q2_2_b, mat(stats) replace nobox

* Q2.3c (MAR)- feasible estimator
* we predicted p before, but did not use t, so do that now:
* logit regression, robust standard errors
logit s dpisofirme S_age S_HHpeople log_inc, vce(robust)

* predict propensity score
predict p_witht

* now run gmm adding in new term s/p
local vars = "dpisofirme S_age S_HHpeople log_inc"
gmm ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*dpisofirme) ///
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*S_age) ///
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*S_HHpeople) ///
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*log_inc), ///
instruments(`vars') winitial(identity) vce(boot)

* output for LaTeX
 mata:
    coef = st_matrix("e(b)")'
    se = st_matrix("e(se)")'

    tstat = coef:/se

    CI_low = coef - 1.96:*se
    CI_high = coef + 1.96:*se

    stats = round((coef,se,tstat,CI_low,CI_high),.001)

    st_matrix("stats",stats)
end
mat rownames stats = `vars'
mat colnames stats = coef se tstat CI_low CI_high
outtable using stata_table_q2_3_c, mat(stats) replace nobox

* Q2.3d (MAR)- feasible estimator, trimmed
* we predicted p before, and have s, so add that before the moment conditions
local vars = "dpisofirme S_age S_HHpeople log_inc"
gmm ((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*dpisofirme) ///
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*S_age) ///
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+log_inc*{gamma3})))*S_HHpeople) ///
```

```
((s/p_witht)*(danemia - invlogit((dpisofirme*{theta}+S_age*{gamma1}+S_HHpeople*{gamma2}+
log_inc*{gamma3})))*log_inc) ///
if p_witht >= 0.1, instruments(`vars') winitial(identity) vce(boot)

* output for LaTeX
 mata:
    coef = st_matrix("e(b)")'
    se = st_matrix("e(se)")'

    tstat = coef:/se

    CI_low = coef - 1.96:*se
    CI_high = coef + 1.96:*se

    stats = round((coef,se,tstat,CI_low,CI_high),.001)

    st_matrix("stats",stats)
end
mat rownames stats = `vars'
mat colnames stats = coef se tstat CI_low CI_high
outtable using stata_table_q2_3_d, mat(stats) replace nobox






********************************
*** 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 stata_plot_q3_1.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 stata_plot_q3_2.png, replace
```