Economics 675: Applied Microeconometrics
Fall 2018 - Assignment 1

Paul R. Organ

September 24, 2018

Contents

1 Question 1: Simple Linear Regression with Measurement Error

1.1 OLS Estimator

True model: $y_i = x_i\beta + \varepsilon_i$
Observed variables: y_i and $\tilde{x}_i = x_i + u_i$
Plug observed into true: $y_i = (\tilde{x}_i - u_i)\beta + \varepsilon_i = \tilde{x}_i\beta + v_i$, where $v_i = -u_i\beta + \varepsilon_i$.
Then using the Law of Large Numbers:

$$\hat{\beta}_{LS} = (\tilde{x}'\tilde{x})^{-1}\tilde{x}'y = (\tilde{x}'\tilde{x})^{-1}\tilde{x}'(\tilde{x}\beta + v_i) = \beta + (\tilde{x}'\tilde{x})^{-1}\tilde{x}'v_i$$

$$\beta + (\tilde{x}'\tilde{x})^{-1}\tilde{x}'v_i \to_p \beta + \frac{\mathbb{E}[\tilde{x}_i v_i]}{\mathbb{E}[\tilde{x}_i^2]} = \beta + \frac{\mathbb{E}[(x_i + u_i)(-u_i\beta + \varepsilon_i)]}{\mathbb{E}[(x_i + u_i)^2]}$$

$$= \beta + \frac{\mathbb{E}[-x_i u_i\beta + x_i\varepsilon_i - u_i^2\beta + u_i\varepsilon_i]}{\mathbb{E}[x_i^2 + 2x_i u_i + u_i^2]} = \beta(1 - \frac{\mathbb{E}[u_i^2]}{\mathbb{E}[x_i^2] + \mathbb{E}[u_i^2]}) = \beta(\frac{\mathbb{E}[x_i^2]}{\mathbb{E}[x_i^2] + \mathbb{E}[u_i^2]})$$

Using the given characteristics of $\mathbb{E}[x_i\varepsilon_i]$, $\mathbb{E}[\varepsilon_i u_i]$, etc.
The attenuation factor $\lambda = \frac{\mathbb{E}[x_i^2]}{\mathbb{E}[x_i^2]+\mathbb{E}[u_i^2]} = \frac{\sigma_x^2}{\sigma_x^2+\sigma_u^2} < 1$; $\hat{\beta}_{LS}$ is biased downwards for β.

1.2 Standard Errors

Observed error: $\hat{\varepsilon} = y - \tilde{x}\hat{\beta}_{LS} = y - (x+u)\hat{\beta}_{LS}$
True error: $\varepsilon = y - x\beta$
Add and subtract true error:
$\hat{\varepsilon} = (y - x\beta) + y - (x+u)\hat{\beta}_{LS} - (y - x\beta) = \varepsilon + x(\beta - \hat{\beta}_{LS}) - u\hat{\beta}_{LS}$
Then:

$$\begin{aligned}\frac{\hat{\varepsilon}'\hat{\varepsilon}}{n} &\to_p \mathbb{E}[(\varepsilon + x(\beta - \hat{\beta}_{LS}) - u\hat{\beta}_{LS})'(\varepsilon + x(\beta - \hat{\beta}_{LS}) - u\hat{\beta}_{LS})] \\ &= \mathbb{E}[\varepsilon'\varepsilon + \varepsilon' x(\beta - \hat{\beta}_{LS}) - \varepsilon' u\hat{\beta}_{LS} \\ &\quad + x(\beta - \hat{\beta}_{LS})'\varepsilon + (x(\beta - \hat{\beta}_{LS}))'(x(\beta - \hat{\beta}_{LS})) - (x(\beta - \hat{\beta}_{LS}))' u\hat{\beta}_{LS} \\ &\quad - (u\hat{\beta}_{LS})'\varepsilon - (u\hat{\beta}_{LS})'(x(\beta - \hat{\beta}_{LS})) + (u\hat{\beta}_{LS})' u\hat{\beta}_{LS}]\end{aligned}$$

All but the first, fifth, and ninth terms drop out. Then:

$$\frac{\hat{\varepsilon}'\hat{\varepsilon}}{n} \to_p \sigma_\varepsilon^2 + \sigma_x^2(\beta - \lambda\beta)'(\beta - \lambda\beta) + \sigma_u^2(\lambda\beta)'(\lambda\beta) = \sigma_\varepsilon^2 + (1-\lambda)^2\beta^2\sigma_x^2 + \lambda^2\beta^2\sigma_u^2$$

Since all three terms are positive, we have that $\hat{\sigma}_\varepsilon^2$ is biased upwards for σ_ε^2.

$$\begin{aligned}\text{plim}\frac{\hat{\sigma}_\varepsilon^2}{\sigma_{\tilde{x}}^2} &= \frac{\sigma_\varepsilon^2 + (1-\lambda)^2\beta^2\sigma_x^2 + \lambda^2\beta^2\sigma_u^2}{\sigma_x^2 + \sigma_u^2} \\ &= \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}(\frac{\sigma_\varepsilon^2}{\sigma_x^2}) + \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}(1-\lambda)^2\beta^2 + \frac{\sigma_u^2}{\sigma_x^2 + \sigma_u^2}\lambda^2\beta^2 \\ &= \lambda\frac{\sigma_\varepsilon^2}{\sigma_x^2} + \lambda(1-\lambda)^2\beta^2 + \lambda^2(1-\lambda)\beta^2 = \lambda\frac{\sigma_\varepsilon^2}{\sigma_x^2} + \lambda(1-\lambda)\beta^2\end{aligned}$$

In the final expression, the first term biases downwards, but the second term is positive and of unknown magnitude; we cannot say for certain if the total is biased upward or downward.

1.3 t-test

This is a straightforward application of Slutsky's Theorem, using what we showed in parts (1.1) and (1.2): $\hat{\beta}_{LS} \to_p \lambda\beta$ and $\hat{\sigma}_\varepsilon^2/\sigma_x^2 \to_p \lambda(\sigma_\varepsilon^2/\sigma_x^2) + \lambda(1-\lambda)\beta^2$
Thus applying Slutsky's Theorem we have:

$$\frac{\hat{\beta}_{LS}}{\sqrt{\hat{\sigma}_\varepsilon^2/\sigma_x^2}} \to_p \frac{\lambda\beta}{\sqrt{\lambda(\sigma_\varepsilon^2/\sigma_x^2) + \lambda(1-\lambda)\beta^2}} = \frac{\sqrt{\lambda}\beta}{\sqrt{(\sigma_\varepsilon^2/\sigma_x^2) + (1-\lambda)\beta^2}}$$

This is biased downwards, since the usual t-ratio is $\beta/\sqrt{(\sigma_\varepsilon^2/\sigma_x^2)}$

1.4 Second Measurement, Consistency

Using the usual definition of the IV estimator, we have:

$$\hat{\beta}_{IV} = (\breve{x}'\tilde{x})^{-1}\breve{x}'y$$

Then, using the Law of Large Numbers, we have:

$$\hat{\beta}_{IV} = \frac{\breve{x}'y}{\breve{x}'\tilde{x}} = \frac{\breve{x}'x/n}{\breve{x}'\tilde{x}/n}\beta + \frac{\breve{x}'\varepsilon/n}{\breve{x}'\tilde{x}/n} \to_p \frac{\mathbb{E}[\breve{x}_i x_i]}{\mathbb{E}[\breve{x}_i(x_i+u_i)]}\beta + \frac{\mathbb{E}[\breve{x}_i\varepsilon_i]}{\mathbb{E}[\breve{x}_i(x_i+u_i)]} = \beta$$

using $\mathbb{E}[\breve{x}_i u_i] = 0$ (so the first term simplifies to β) and $\mathbb{E}[\breve{x}_i\varepsilon_i] = 0$ (so the second term drops out).

1.5 Second Measurement, Distribution

As noted above in (1.1), we have $y_i = \tilde{x}_i\beta + (\varepsilon_i - u_i\beta)$, or $\mathbf{y} = \tilde{\mathbf{x}}\beta + (\varepsilon - \mathbf{u}\beta)$.
Then we have:

$$\sqrt{n}(\hat{\beta}_{IV} - \beta) = \frac{\breve{\mathbf{x}}'(\varepsilon - \mathbf{u}\beta)/\sqrt{n}}{\breve{\mathbf{x}}'\tilde{\mathbf{x}}/n} = (\mathbb{E}[\breve{x}_i x_i])^{-1}\frac{\breve{\mathbf{x}}'(\varepsilon - \mathbf{u}\beta)}{\sqrt{n}} + o_p(1) \to_d \mathcal{N}(0, V_{IV})$$

by the Central Limit Theorem and Law of Large Numbers, with $V_{IV} = \frac{\mathbb{E}[\breve{x}_i^2(\varepsilon_i - u_i\beta)^2]}{(\mathbb{E}[\breve{x}_i x_i])^2}$

1.6 Second Measurement, Inference

First we construct an estimate for V_{IV}, then build a confidence interval for β using $\hat{\beta}_{IV}$.
To estimate V_{IV}, we use the traditional heteroskedasticity-consistent standard error estimator:

$$\hat{V}_{IV} = (\frac{\breve{\mathbf{x}}'\tilde{\mathbf{x}}}{n})^{-1}(\frac{1}{n}\sum_{i=1}^{n}\breve{x}_i^2\hat{v}_i^2)(\frac{\breve{\mathbf{x}}'\tilde{\mathbf{x}}}{n})^{-1}$$

with $\hat{v}_i$ the predicted residuals, i.e. $\hat{v}_i = y_i - \tilde{x}_i\hat{\beta}_{IV}$.
Given an estimate for V_{IV} of this form, we will have that $\hat{V}_{IV} \to_p V_{IV}$, and the 95% confidence interval for β will be:

$$CI_{.95} = \left[\hat{\beta}_{IV} \pm 1.96 \cdot \sqrt{\hat{V}_{IV/n}}\right]$$

1.7 Validation Sample, Consistency

We showed in part (1.1) that $\hat{\beta}_{LS} \to_p \lambda\beta$, with $\lambda = \frac{\sigma_x^2}{\sigma_x^2+\sigma_u^2}$.
Here, we can use this estimator $\breve{\sigma}_x^2$ to construct a consistent estimator of λ, and then use that to back out a consistent estimator of β:

$$\hat{\lambda} = \frac{\breve{\sigma}_x^2}{\tilde{\mathbf{x}}'\tilde{\mathbf{x}}} \to_p \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2} = \lambda$$

Then using Slutsky's Theorem, we have $\hat{\beta}_{VS} = \hat{\beta}_{LS}/\hat{\lambda} \to_p \beta$.

1.8 Validation Sample, Distribution

Here we can apply the Delta Method, with three-dimensional $\mathbf{w}$, and $g(\mathbf{w}) = w_1 w_2 / w_3$, and $\dot{g}(w) = dg(\mathbf{w})/d\mathbf{w}$. Then we have:

$$\sqrt{n}\left(g\left(\left[\begin{array}{c}\hat{\beta}_{LS} \\ \tilde{\mathbf{x}}'\tilde{\mathbf{x}}/n \\ \breve{\sigma}_x^2\end{array}\right]\right) - g\left(\left[\begin{array}{c}\lambda\beta \\ \sigma_x^2+\sigma_u^2 \\ \sigma_x^2\end{array}\right]\right)\right) = \sqrt{n}\left(\frac{\hat{\beta}_{LS}}{\hat{\lambda}} - \beta\right) \rightarrow_d \mathcal{N}(\mathbf{0}, V_{VS})$$

And we have the usual form for V_{VS}, i.e., $V_{VS} = \dot{g}(\mathbf{w}_0)\mathbf{\Sigma}\dot{g}(\mathbf{w}_0)$, and $\mathbf{w}_0 = (\lambda\beta, \sigma_x^2+\sigma_u^2, \sigma_x^2)'$.

1.9 Validation Sample, Inference

Here we can use $\hat{\mathbf{w}} = (\hat{\beta}_{LS}, \tilde{\mathbf{x}}'\tilde{\mathbf{x}}/n, \breve{\sigma}_x^2)'$ as an estimator for $\mathbf{w}_0$ above in (1.8)
Plug that and the estimator $\hat{\mathbf{\Sigma}}$ into our equation for V_{VS} in (1.8) to produce the estimator $\hat{V}_{VS}$.
And take $g(\hat{\mathbf{w}})$ to produce an estimator for β, which is the same as $\hat{\beta}_{VS}$ defined in (1.7).
Then the 95% confidence interval for β will take the form:

$$CI_{.95} = \left[\hat{\beta}_{VS} \pm 1.96 \cdot \sqrt{\hat{V}_{VS}/n}\right]$$

1.10 FE Estimator, Consistency

Since we have $T = 2$, the fixed-effects estimator is really a first-difference estimator. That is, we have

$$\hat{\beta}_{FE} = \hat{\beta}_{FD} = (\triangle\tilde{\mathbf{x}}' \triangle \tilde{\mathbf{x}})^{-1} \triangle \tilde{\mathbf{x}}' \triangle \mathbf{y}$$

Then the consistency will follow as in (1.1), using the differences instead of the observed values themselves:

$$\hat{\beta}_{FE} \rightarrow_p \frac{\mathbb{E}[(\triangle x_i + \triangle u_i) \triangle x_i]}{\mathbb{E}[(\triangle x_i + \triangle u_i)^2]}\beta = \frac{\mathbb{E}[(\triangle x_i)^2]}{\mathbb{E}[(\triangle x_i)^2] + \mathbb{E}[(\triangle u_i)^2]}\beta = \gamma\beta$$

with $\gamma = \frac{\sigma_{\triangle x}^2}{\sigma_{\triangle x}^2 + \sigma_{\triangle u}^2} < 1$, so that we see $\hat{\beta}_{FE}$ is biased downwards for β, as was $\hat{\beta}_{LS}$.

1.11 FE Estimator, Time Dependence

First, from the definition of variance we have

$$\sigma_{\triangle x}^2 = \mathbb{V}[x_{i2} - x_{i1}] = \mathbb{V}[x_{i2}] + \mathbb{V}[x_{i1}] - 2\mathbb{C}[x_{i2}, x_{i1}] = 2\sigma_x^2 - 2\mathbb{C}[x_{i2}, x_{i1}]$$

Then using the definition of γ from (1.10) above, we have:

$$\gamma = \frac{\sigma_{\triangle x}^2}{\sigma_{\triangle x}^2 + \sigma_{\triangle u}^2} = \frac{\sigma_x^2 - \mathbb{C}[x_{i2}, x_{i1}]}{\sigma_x^2 - \mathbb{C}[x_{i2}, x_{i1}] + \sigma_u^2 - \mathbb{C}[u_{i2}, u_{i1}]} = \frac{\sigma_x^2(1-\rho_x)}{\sigma_x^2(1-\rho_x) + \sigma_u^2(1-\rho_u)} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2\frac{1-\rho_u}{1-\rho_x}}$$

1.12 FE Estimator, Implications

Note that the terms ρ_x and ρ_u are the only thing that makes λ different from γ:

$$\lambda = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}; \quad \gamma = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2\frac{1-\rho_u}{1-\rho_x}}$$

If ρ_x is close to 1, and ρ_u is close to 0, the denominator of γ will get very large, and thus γ will tend towards zero, with $\gamma < \lambda$. So the fixed-effects estimator will suffer from even worse attenuation bias than the plan OLS estimator when there is measurement error in x.

2 Question 2: Implementing Least-Squares Estimators

2.1 Derivation of $\hat{\boldsymbol{\beta}}(\mathbf{W})$

I will use the "add and subtract" method. Take the arg min over $\boldsymbol{\beta} \in \mathbb{R}^d$:

$$\begin{aligned}
||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2_{\mathbf{W}} &= ||\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{X}\tilde{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}||^2_{\mathbf{W}} \\
&= (\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{X}\tilde{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta})'\mathbf{W}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{X}\tilde{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}) \\
&= (\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})'\mathbf{W}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}) + 2(\mathbf{X}\tilde{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta})'\mathbf{W}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}) + (\mathbf{X}\tilde{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta})'\mathbf{W}(\mathbf{X}\tilde{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}) \\
&= ||\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}||^2_{\mathbf{W}} + 2(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})'\mathbf{X}'\mathbf{W}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}) + ||\mathbf{X}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}||^2_{\mathbf{W}}
\end{aligned}$$

To set the middle term equal to zero, we must have $\mathbf{X}'\mathbf{W}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}) = 0$, which implies $\tilde{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$. Then, if the middle term is zero, we can minimize the remaining terms by taking $\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}}$, which yields $\hat{\boldsymbol{\beta}}(\mathbf{W}) = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$ as desired.

2.2 Distribution of $\hat{\boldsymbol{\beta}}(\mathbf{W})$

First, plug in our expression for $\hat{\boldsymbol{\beta}}(\mathbf{W})$ and $\mathbf{y}$:

$$\sqrt{n}((\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}(\mathbf{X}\boldsymbol{\beta} + \varepsilon) - \boldsymbol{\beta}) = \sqrt{n}((\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\varepsilon)$$

Then, we can rewrite this and apply the LLN, CLT, and Slutsky Theorem:

$$\sqrt{n}((\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\varepsilon = (\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\mathbf{x}_i)^{-1}(\sqrt{n}\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\varepsilon_i) \to_d \mathcal{N}(\mathbf{0}, \mathbf{V}(\mathbf{W}))$$

with $\mathbf{V}(\mathbf{W})$ the normal sandwich form: $\mathbb{E}[\mathbf{x}_i'\mathbf{W}\mathbf{x}_i]^{-1}\mathbb{V}[\mathbf{x}_i\mathbf{W}\varepsilon_i]\mathbb{E}[\mathbf{x}_i'\mathbf{W}\mathbf{x}_i]^{-1}$,
using that $\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\mathbf{x}_i \to_p \mathbb{E}[\mathbf{x}_i'\mathbf{W}\mathbf{x}_i]$, and $\sqrt{n}\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\varepsilon_i \to_d \mathcal{N}(\mathbf{0}, \mathbb{V}[\mathbf{x}_i\mathbf{W}\varepsilon_i])$, since $\mathbb{E}[\mathbf{x}_i'\mathbf{W}\varepsilon_i] = 0$.
If $\mathbb{V}[\mathbf{y}|\mathbf{X}, \mathbf{W}] = \sigma^2\mathbf{I}_n$, then some of the terms in $\mathbf{V}(\mathbf{W})$ will cancel out, leaving $\mathbf{V}(\mathbf{W}) = \sigma^2\mathbb{E}[\mathbf{x}_i'\mathbf{W}\mathbf{x}_i]^{-1}$.

2.3 Estimating $\mathbf{V}(\mathbf{W})$

I propose the following VCE of $\mathbf{V}(\mathbf{W})$:

$$\hat{\mathbf{V}}(\mathbf{W}) = (\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\mathbf{x}_i)^{-1}(\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\hat{\varepsilon}_i\hat{\varepsilon}_i'\mathbf{W}\mathbf{x}_i)(\frac{1}{n}\sum_i \mathbf{x}_i'\mathbf{W}\mathbf{x}_i)^{-1}$$

with $\hat{\varepsilon}_i = y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}$ (the estimated residuals).
Or in matrix form:

$$\hat{\mathbf{V}}(\mathbf{W}) = (\frac{1}{n}\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}(\frac{1}{n}\mathbf{X}'\mathbf{W}\mathrm{diag}(\hat{\varepsilon}_i^2)\mathbf{W}\mathbf{X})(\frac{1}{n}\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$$

2.4 Matrix Implementation in Matlab

See Appendix 1 for the MATLAB code.

2.5 Implementation in R and Stata

The following tables show the results from MATLAB, R, and STATA. See Appendix 2 for the R code and Appendix 3 for the STATA code.

Table 1: Matrix outputs from Matlab using symmetric inverse

Variable	Coef	se	t_stat	p_val	CI_low	CI_high
Constant	6,485.55	5,701.83	1.14	0.26	-4,690.04	17,661.15
treat	1,535.48	637.09	2.41	0.02	286.78	2,784.18
black	-2,592.38	897.20	-2.89	0.00	-4,350.90	-833.86
age	39.34	45.46	0.87	0.39	-49.76	128.44
educ	-740.54	1,083.74	-0.68	0.49	-2,864.67	1,383.59
educ2	60.08	56.32	1.07	0.29	-50.31	170.47
earn74	-0.03	0.16	-0.18	0.85	-0.35	0.29
blackXearn74	0.18	0.17	1.06	0.29	-0.15	0.50
u74	1,316.03	1,170.47	1.12	0.26	-978.09	3,610.16
u75	-1,167.69	946.95	-1.23	0.22	-3,023.71	688.33

Table 2: Matrix outputs from Matlab using Cholseky inverse

Variable	Coef	se	t_stat	p_val	CI_low	CI_high
Constant	6,485.55	5,701.83	1.14	0.26	-4,690.04	17,661.15
treat	1,535.48	637.09	2.41	0.02	286.78	2,784.18
black	-2,592.38	897.20	-2.89	0.00	-4,350.90	-833.86
age	39.34	45.46	0.87	0.39	-49.76	128.44
educ	-740.54	1,083.74	-0.68	0.49	-2,864.67	1,383.59
educ2	60.08	56.32	1.07	0.29	-50.31	170.47
earn74	-0.03	0.16	-0.18	0.85	-0.35	0.29
blackXearn74	0.18	0.17	1.06	0.29	-0.15	0.50
u74	1,316.03	1,170.47	1.12	0.26	-978.09	3,610.16
u75	-1,167.69	946.95	-1.23	0.22	-3,023.71	688.33

Table 3: Regression outputs using R

	coefficient	std_error	t_stat	p_val	conf_int
(Intercept)	6485.55	5701.83	1.14	0.26	[-4689.84, 17660.94]
treat	1535.48	637.09	2.41	0.02	[286.81, 2784.16]
black	-2592.38	897.20	-2.89	0.00	[-4350.86, -833.89]
age	39.34	45.46	0.87	0.39	[-49.76, 128.44]
educ	-740.54	1083.74	-0.68	0.49	[-2864.63, 1383.55]
educ2	60.08	56.32	1.07	0.29	[-50.3, 170.47]
earn74	-0.03	0.16	-0.18	0.85	[-0.35, 0.29]
blackXearn74	0.18	0.17	1.06	0.29	[-0.15, 0.5]
u74	1316.03	1170.47	1.12	0.26	[-978.05, 3610.11]
u75	-1167.69	946.95	-1.23	0.22	[-3023.68, 688.3]

Table 4: Regression outputs using STATA

VARIABLES	coef	se	tstat	pval	ci
earn78	.	.	.	.	. - .
treat	1,535.48	637.09	2.41	0.02	283.32 - 2,787.64
black	-2,592.38	897.20	-2.89	0.00	-4,355.77 - -828.98
age	39.34	45.46	0.87	0.39	-50.00 - 128.68
educ	-740.54	1,083.74	-0.68	0.49	-2,870.56 - 1,389.48
educ2	60.08	56.32	1.07	0.29	-50.61 - 170.78
earn74	-0.03	0.16	-0.18	0.85	-0.35 - 0.29
blackXearn74	0.18	0.17	1.06	0.29	-0.15 - 0.50
u74	1,316.03	1,170.47	1.12	0.26	-984.45 - 3,616.51
u75	-1,167.69	946.95	-1.23	0.22	-3,028.85 - 693.48
Constant	6,485.55	5,701.83	1.14	0.26	-4,721.02 - 17,692.12

3 Question 3: Analysis of Experiments

3.1 Neyman's Approach

(a) First, note that

$$\begin{aligned}\mathbb{E}[T_{DM}] &= \mathbb{E}[\bar{Y}_1 - \bar{Y}_0] = \mathbb{E}[\bar{Y}_1] - \mathbb{E}[\bar{Y}_0] \\ &= \mathbb{E}[Y_i|T_i = 1] - \mathbb{E}[Y_i|T_i = 0] = \mathbb{E}[Y_i(1)] - \mathbb{E}[Y_i(0)] \\ &= \frac{1}{n}\sum_{i=1}^{n} Y_i(1) - \frac{1}{n}\sum_{i=1}^{n} Y_i(0) =: \tau_{ATE}\end{aligned}$$

I compute an unbiased estimator of the ATE (code in appendices), and my estimate is $\tau_{ATE} = 1,794$ using both R and STATA.

(b) I construct an asymptotically conservative 95% confidence interval for the ATE (code in appendices), and the resulting interval is $[474.01, 3114.68]$ (using both R and STATA).

3.2 Fisher's Approach

(a) Using Fisher's permutation approach, I estimate p-values for the sharp null hypothesis of no treatment effect of 0.01 (using R) and 0.0090 (using STATA). These may differ because the `permts` function in R rounds the p-value to the nearest hundredth.

Using the Kolgomorov-Smirnov test, I estimate p-values for the null hypothesis of equal distribution functions of 0.046 (using R) and 0.046 (using STATA).

(b) Confidence intervals: I calculate a 95% confidence interval of $[578.65, 3012.87]$ using R and $[459.07, 3129.61]$ using STATA. See the code in the respective appendices.

3.3 Power Calculations

(a) See figures below, and appendices for code.

(b) Using both R and STATA, I calculate the minimum sample size needed to detect a \$1,000 increase in earnings with a power of 0.80 as $1,440$. See appendices for code.

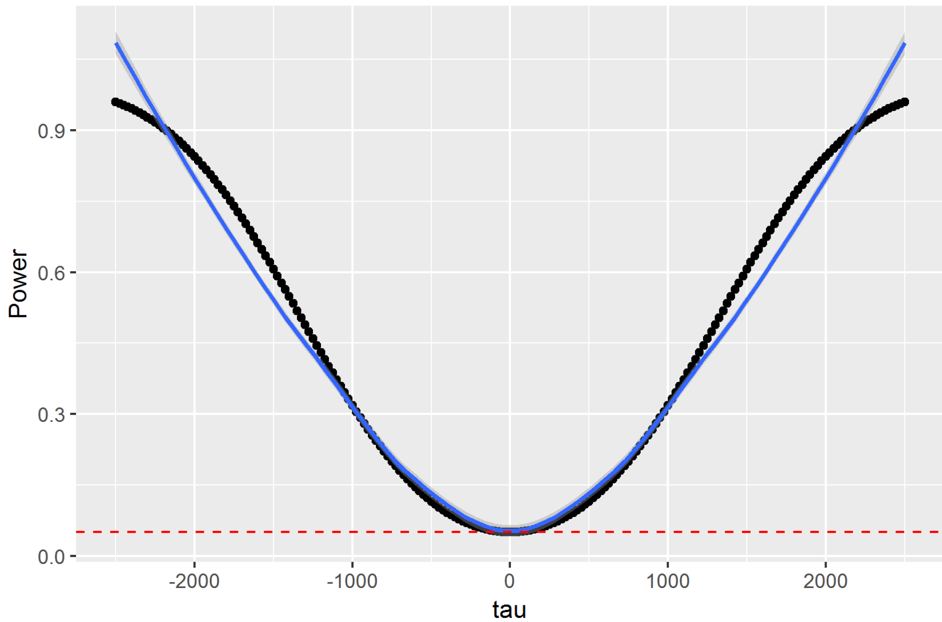

Figure 1: Power Function: R

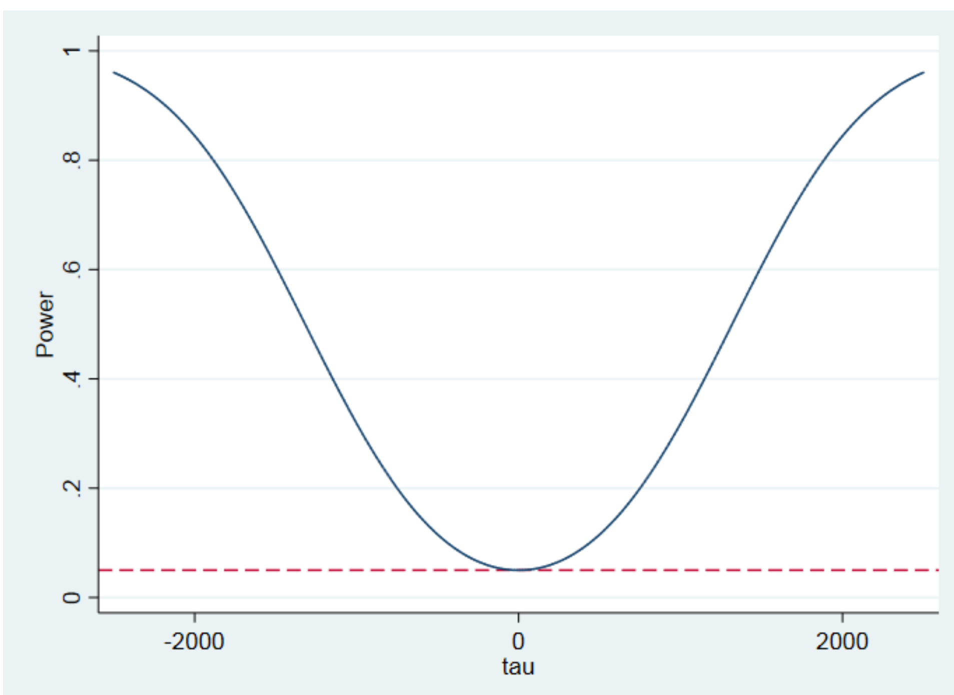

Figure 2: Power Function: Stata

4 Appendix 1: Matlab Code

```
% Author: Paul R. Organ
% Purpose: ECON 675, PS1
% Last Update: Sept 18, 2018

% read in data
M = readtable('LaLonde_1986.csv');

% generate new variables
M.educ2 = M.educ.^2;
M.blackXearn74 = M.black .* M.earn74;
M.intercept = ones(445,1);

n=445;

% define matrices to work with
vars = {'intercept', 'treat', 'black', 'age', 'educ', 'educ2',...
        'earn74', 'blackXearn74', 'u74', 'u75'};
X = M{:,vars};
W = eye(n);
y = M{:,'earn78'};

% define matrix for inverse
A = X'*W*X

% point estimate for beta using symmetric inverse
beta_symm = inv(A)*X'*W*y

% Cholesky decomposition
L = chol(A, 'lower');

% compute Cholesky inverse
% https://www.mathworks.com/matlabcentral/answers/371694-find-inverse
% -and-determinant-of-a-positive-definite-matrix
U = L\eye(10);
Ainv = L'\U;

% point estimate for beta using Cholesky inverse
beta_chol = Ainv*X'*W*y

% estimates for V
% first calculate residuals (eps = y - Xbeta)
eps_symm = y-X*beta_symm;
eps_chol = y-X*beta_chol;

% Cholesky for Variance
Uv = (1/n).*L\eye(10);
Ainvu = L'\Uv;

% estimate V(W)
v_symm = (sum(eps_symm.^2)/(n-10))*inv(A)
v_chol = (sum(eps_symm.^2)/(n-10))*Ainv
```

```
% standard errors
se_symm = sqrt(diag(v_symm))
se_chol = sqrt(diag(v_chol))

% t-tests
beta_null = zeros(10,1)
tvals_symm = zeros(10,1);
tvals_chol = zeros(10,1);
for i = 1:10
    ej = zeros(10,1);
    ej(i,1) = 1;

    tvals_symm(i) = (ej'*beta_symm-ej'*beta_null)/se_symm(i);
    tvals_chol(i) = (ej'*beta_chol-ej'*beta_null)/se_chol(i);
end
tvals_symm
tvals_chol

% p values
pvals_symm = 2*(1-tcdf(abs(tvals_symm), n-10))
pvals_chol = 2*(1-tcdf(abs(tvals_chol), n-10))

% confidence intervals
conf_int_symm = [beta_symm-1.96.*se_symm beta_symm+1.96.*se_symm]
conf_int_chol = [beta_chol-1.96.*se_chol beta_chol+1.96.*se_chol]

% table for output
out_symm = [beta_symm se_symm tvals_symm pvals_symm conf_int_symm];
out_chol = [beta_chol se_chol tvals_chol pvals_chol conf_int_chol];
out_symm = array2table(out_symm,...
    'VariableNames',{'Coef','se','t_stat','p_val','CI_low','CI_high'},...
    'RowNames',{'Constant','treat','black','age','educ',...
    'educ2','earn74','blackXearn74','u74','u75'})
out_chol = array2table(out_chol,...
    'VariableNames',{'Coef','se','t_stat','p_val','CI_low','CI_high'},...
    'RowNames',{'Constant','treat','black','age','educ',...
    'educ2','earn74','blackXearn74','u74','u75'})

% write to Excel for conversion to LaTeX
% https://www.ctan.org/tex-archive/support/excel2latex/?lang=en
writetable(out_symm, 'q2Matlab.xlsx', 'Sheet', 'symm', 'WriteRowNames',true)
writetable(out_chol, 'q2Matlab.xlsx', 'Sheet', 'chol', 'WriteRowNames',true)
```

5 Appendix 2: R Code

```
#########################################################################################
# Author: Paul R. Organ
# Purpose: ECON 675, PS1
# Last Update: Sept 24, 2018
#########################################################################################
# Preliminaries
options(stringsAsFactors = F)

# packages
require(tidyverse) # data cleaning and manipulation
require(magrittr)  # syntax
require(xtable)    # regression table output
require(perm)      # permutation tests
require(boot)      # bootstrapping
require(ggplot2)   # plots

set.seed(22)
select = dplyr::select
setwd('C:/Users/prorgan/Box/Classes/Econ_675/Problem_Sets/PS1')

#########################################################################################
# read in data
df <- read_csv('LaLonde_1986.csv')

#########################################################################################
# Question 2: Implementing Least-Squares Estimators

# add necessary variables
df %<>% mutate(educ2 = educ*educ, blackXearn74 = black * earn74)

# run regression
reg <- lm(earn78 ~ treat + black + age + educ + educ2 +
            earn74 + blackXearn74 + u74 + u75, data = df)

# start table
q2 <- xtable(reg)
colnames(q2) <- c('coefficient', 'std_error', 't_stat', 'p_val')

# Z-value for 95%
z = qnorm(.025, lower.tail=F)

# add confidence interval
q2$conf_int <-
  paste0('[',round(q2$coefficient-z*q2$std_error,2),', ',
         round(q2$coefficient+z*q2$std_error,2),']')

# output table for Latex
xtable(q2)

#########################################################################################
## Question 3: Analysis of Experiments
# 3.1) Neyman's Approach:
```

```r
# 3.1a) Average Treatment Effect
N1 = sum(df$treat==1)
N0 = sum(df$treat==0)

sumY1 = sum(df$earn78[df$treat==1])
sumY0 = sum(df$earn78[df$treat==0])

Ybar1 = sumY1/N1
Ybar0 = sumY0/N0

ATE = Ybar1-Ybar0

# 3.1b) t-test
S1 = (1/(N1-1))*var(df$earn78[df$treat==1])
S0 = (1/(N0-1))*var(df$earn78[df$treat==0])

se <- sqrt(S1+S0)

Tstat = ATE / se
pval = 2*pnorm(-abs(Tstat))

CI_31b = paste0('[',round(ATE-z*sqrt(S1+S0),2),', '
           ,round(ATE+z*sqrt(S1+S0),2),']')

# Canned version for comparison
t.test(earn78 ~ treat, data = df)

##########################################################################################
# 3.2) Fisher's Approach
# 3.2a) p-Value
# Fisher
fisher_1 <- permTS(earn78 ~ treat, data = df,
                   alternative = 'two.sided', method = 'exact.mc',
                   control = permControl(nmc=999,p.conf.level=.95))
fisher_1

# Kolgomorov-Smirnov
earn78_0 <- df$earn78[df$treat==0]
earn78_1 <- df$earn78[df$treat==1]
ks.test(earn78_0, earn78_1, alternative = 'two.sided', exact = T)

# 3.2b) Confidence Interval
# Imputation assuming ATE is constant
# (Generating Yi(1) and Yi(0) for each i, assuming ATE estimate is constant)
Y1_imp <- (df$treat==1) * df$earn78 + (df$treat==0) * (df$earn78 + ATE)
Y0_imp <- (df$treat==1) * (df$earn78 - ATE) + (df$treat==0) * df$earn78

# define statistic (difference in means)
boot_T <- function(x, ind) {
  mean(Y1_imp[df$treat[ind]==1]) - mean(Y0_imp[df$treat[ind]==0])
}

# run bootstrap using defined statistic
boot_results <- boot(df, R = 999, statistic = boot_T,
```

```r
                    sim = "permutation", stype = "i")

# construct 95% confidence interval
CI_32b = paste0('[', quantile(boot_results$t,0.025), ', ',
                quantile(boot_results$t,0.975), ']')

################################################################################
# 3.3) Power Calculations
# 3.3a) graphing power function

# testing tau_0 = 0. plot tau on either side of 0
df_p <- data.frame(tau = seq(-2500, 2500, 25))

# calculate propability of rejection under each alternative tau
df_p %<>% mutate(prob_rej = pnorm(z-tau/se, lower.tail=F) +
                   pnorm(z+tau/se, lower.tail=F))

plot <- ggplot(df_p, aes(x = tau, y = prob_rej)) +
  geom_point() + geom_smooth() +
  ylab('Power') + xlab('tau') +
  geom_hline(yintercept=0.05,linetype='dashed',color='red')
ggsave('power_R.png', plot, width = 6, height = 4, units = 'in')

# 3.3b) determining minimum sample size
df_n <- data.frame(n = seq(100,5000,5))

# given: probablity of treatment is 2/3
p <- 2/3

# effect we want to detect
tau_0 <- 1000

# variances of observed data
V1 <- var(df$earn78[df$treat==1])
V0 <- var(df$earn78[df$treat==0])

# simulate with different sample sizes
df_n %<>% mutate(n1 = n*p,
                 n0 = n-n1,
                 std_err = sqrt(V1/n1 + V0/n0),
                 power = pnorm(z-tau_0/std_err, lower.tail=F) +
                   pnorm(z+tau_0/std_err, lower.tail=F))

# find sample size st power is at least .8
min_n <- min(df_n$n[df_n$power>=0.8])

################################################################################
```

6 Appendix 3: Stata Code

```
*****************************************************************************
* Author: Paul R. Organ
* Purpose: ECON 675, PS1
* Last Update: Sept 18, 2018
*****************************************************************************
clear all
set more off
capture log close

cd "C:\Users\prorgan\Box\Classes\Econ 675\Problem Sets\PS1"
log using ps1.log, replace
set seed 22

* load data
insheet using "LaLonde_1986.csv", clear

*****************************************************************************
* Question 2: Implementing Least-Squares Estimators
* generate new variables
gen educ2 = educ^2
gen blackXearn74 = black * earn74

* run regression
reg earn78 treat black age educ educ2 earn74 blackXearn74 u74 u75

* output table to include in LaTeX
outreg2 using q2outreg.tex, side stats(coef se tstat pval ci) ///
 noaster noparen nor2 noobs dec(2) replace

*****************************************************************************
* Question 3: Analysis of Experiments
* 3.1) Neyman's Approach
* manually
sum earn78 if treat==0
local N0 = r(N)
local mu0 = r(mean)
local sd0 = r(sd)
local V0 = r(Var)/(`N0'-1)

sum earn78 if treat==1
local N1 = r(N)
local mu1 = r(mean)
local sd1 = r(sd)
local V1 = r(Var)/(`N1'-1)

local tau = `mu1'-`mu0'
local v = sqrt(`V1'+`V0')
local T = `tau'/`v'
local pval = 2*normal(-abs(`T'))

local ci_low  = `tau' + invnormal(.025)*`v'
local ci_high = `tau' + invnormal(.975)*`v'
```

```
* rounding
local mu0 = round(`mu0', .01)
local mu1 = round(`mu1', .0001)
local sd0 = round(`sd0', .01)
local sd1 = round(`sd1', .0001)
local tau = round(`tau', .01)
local T = round(`T', .01)
local pval = round(`pval', .0001)
local ci_low = round(`ci_low', .01)
local ci_high = round(`ci_high', .01)

di "Control: `mu0' (`sd0') -- N=`N0'"
di "Treatment: `mu1' (`sd1') -- N=`N1'"
di "Difference: `tau'"
di "Test = `T' -- p-val = `pval'"
di "CI: [`ci_low', `ci_high']"

* canned version for comparison:
ttest earn78, by(treat) unequal

* 3.2) Fisher's Approach
* 3.2a) p-Value
* Fisher permutation
permute treat diffmean=(r(mu_2)-r(mu_1)), reps(999) nowarn: ttest earn78, by(treat)

* Kolgomorov-Smirnov
ksmirnov earn78, by(treat) exact

* 3.2b) confidence interval
* define imputed variables
gen y1_imp = earn78
replace y1_imp = earn78 + `tau' if treat==0

gen y0_imp = earn78
replace y0_imp = earn78 - `tau' if treat==1

* define program to bootstrap
program define SUTVAdiff, rclass
        sum y1_imp if treat==1
        local temp1 = r(mean)
        sum y0_imp if treat==0
        local temp0 = r(mean)
        return scalar SUTVAdiff = `temp1' - `temp0'
end

* run bootstrap
bootstrap diff = r(SUTVAdiff), reps(999): SUTVAdiff

* 3.3) Power Calculations
* 3.3a) graphing power function
local a = 0.05
local Z = invnormal(1-`a'/2)
```

```
* Plot power functions and save
twoway (function y = 1 - normal(x/`v'+`Z') + normal(x/`v'-`Z'), range(-2500 2500)), ///
            yline(`a', lpattern(dash)) yti("Power") xti("tau")
graph save power_Stata.png, replace

* 3.3b) determining minimum sample size
mata:
        y = st_data(., "earn78"); t = st_data(., "treat")

        p = 2/3
        Z = 1.96
        tau = 1000

        V1 = quadvariance(select(y,t:==1))
        V0 = quadvariance(select(y,t:==0))

        N = (0::980)*5 + J(981,1,100)

        N1 = N*p
        N0 = N-N1

        std_errors = J(981,1,0)
        for(i = 1; i<=981; i++){
                std_errors[i] = sqrt(V1/N1[i]+V0/N0[i])
        }

        powers = J(981,1,0)
        for(i = 1; i<=981; i++){
                powers[i] = 1-normal(tau/std_errors[i]+Z)+normal(tau/std_errors[i]-Z)
        }
end

* display sample sizes and corresponding power to detect 1000 increase in earnings
mata: N, powers
********************************************************************************
```