Homework #2

Professor: Pat Kline

Students: Christina Brown, Sam Leone, Peter McCrory, Preston Mui

1. Standard Errors for Composite Statistics

Subproblem A

The Delta Method (or the So-Called "Delta Method," so called by Dr. James Powell) says that, if we have a random vector θ that is asymptotically normal with

$$\sqrt{N}(\hat{\theta} - \theta_0) \xrightarrow{d} N(0, \Sigma)$$

then for some $g(\theta)$ that is continuously differentiable at $\theta = \theta_0$ with Jacobian matrix

$$G_0 = \frac{\partial g(\theta_0)}{\partial \theta'}$$

we have that $g(\theta)$ is also asymptotically normal with

$$\sqrt{N}(g(\hat{\theta}) - g(\theta_0) \xrightarrow{d} N(0, G_0' \Sigma G_0)$$

In the context of Fehr and Goette (2007), observe that

$$\theta = \begin{pmatrix} \bar{Y_A} \\ \bar{Y_B} \end{pmatrix}$$

that

$$\Sigma = \begin{bmatrix} \hat{\sigma}^2_{\bar{Y_A}} & \hat{\sigma}_{\bar{Y_A}} \hat{\sigma}_{\bar{Y_B}} \\ \hat{\sigma}_{\bar{Y_A}} \hat{\sigma}_{\bar{Y_B}} & \hat{\sigma}^2_{\bar{Y_B}} \end{bmatrix}$$

and that

$$g(\theta) = \frac{\bar{Y_A} - \bar{Y_B}}{\bar{Y_B}}$$

We can find many of these values in the reprinted table. $\bar{Y_A} = 4131.33$, $\bar{Y_B} = 3005.75$, $\hat{\sigma}^2_{\bar{Y_A}} = (2669.21)^2$, and $\hat{\sigma}^2_{\bar{Y_B}} = (2054.20)^2$. But we still have to do a little bit of work.

Differentiating g with respect to each of its arguments and evaluating them at θ_0 yields G.

$$G = \begin{pmatrix} \frac{\partial g}{\partial \bar{Y_A}} = \frac{1}{\bar{Y_B}} \\ \frac{\partial g}{\partial \bar{Y_B}} = \frac{-\bar{Y_A}}{\bar{Y_B}^2} \end{pmatrix} \Rightarrow G_0 = \begin{pmatrix} \frac{\partial g(\theta_0)}{\partial \bar{Y_A}} = \frac{1}{3005.75} \\ \frac{\partial g(\theta_0)}{\partial \bar{Y_B}} = \frac{-4131.33}{9034533.06} \end{pmatrix}$$

The table also gives us the standard error $\frac{\hat{\sigma}_{\bar{Y_A}-\bar{Y_B}}}{\sqrt{n}}$, which unlike the sample standard deviations, already accounts for the sample size. Combined with a simple variance identity, it can give us $\hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}}$.

$$VAR(\bar{Y_A} - \bar{Y_B}) = VAR(\bar{Y_A}) + VAR(\bar{Y_B}) - 2COV(\bar{Y_A}, \bar{Y_B})$$

$$\Rightarrow COV(\bar{Y_A}, \bar{Y_B}) = \frac{VAR(\bar{Y_A}) + VAR(\bar{Y_B}) - VAR(\bar{Y_A} - \bar{Y_B})}{2}$$

$$\Rightarrow \hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}} = \frac{(2669.21)^2 + (2054.20)^2 - (42)(519.72)^2}{2}$$

$$\Rightarrow \hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}} = -76.67$$

Now we can set up the matrix algebra to use the Delta Method to compute an *estimate* of the standard error, S.

$$S^2 = \frac{1}{n} G_0' \Sigma G_0 = \frac{1}{n} \begin{pmatrix} \frac{\partial g}{\partial \bar{Y_A}} & \frac{\partial g}{\partial \bar{Y_B}} \end{pmatrix} \begin{bmatrix} \hat{\sigma}^2_{\bar{Y_A}} & \hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}} \\ \hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}} & \hat{\sigma}^2_{\bar{Y_B}} \end{bmatrix} \begin{pmatrix} \frac{\partial g}{\partial \bar{Y_A}} \\ \frac{\partial g}{\partial \bar{Y_B}} \end{pmatrix}$$

$$\Rightarrow S^2 = \frac{1}{n}[(\frac{\partial g}{\partial \bar{Y_A}})[(\frac{\partial g}{\partial \bar{Y_A}})(\hat{\sigma}^2_{\bar{Y_A}}) + (\frac{\partial g}{\partial \bar{Y_B}})(\hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}})] + (\frac{\partial g}{\partial \bar{Y_B}})[(\frac{\partial g}{\partial \bar{Y_A}})(\hat{\sigma}_{\bar{Y_A}}\hat{\sigma}_{\bar{Y_B}}) + (\frac{\partial g}{\partial \bar{Y_B}})(\hat{\sigma}^2_{\bar{Y_B}})]]$$

$$\begin{aligned} \Rightarrow S^2 = \frac{1}{42}[\frac{1}{3005.75}[(\frac{1}{3005.75})((2669.21)^2) + (\frac{-4131.33}{9034533.06})(-76.67)] \\ + \frac{-4131.33}{9034533.06}[(\frac{1}{3005.75})(-76.67) + (\frac{-4131.33}{9034533.06})((2054.20)^2)]] \end{aligned}$$

$$\Rightarrow S = .20$$

Subproblem B

Let's rename g to be η. Now we know that

$$\hat{\eta} \sim N(\eta, S^2)$$

So by the rules of variance,

$$\frac{\hat{\eta}}{.25} \sim N(\frac{\eta}{.25}, \frac{S^2}{(.25)^2})$$

$$\Rightarrow \frac{\hat{\eta}}{.25} \sim N(\frac{\eta}{.25}, \frac{(.20)^2}{(.25)^2})$$

$$\Rightarrow \frac{\hat{\eta}}{.25} \sim N(\frac{\eta}{.25}, .64)$$

The experiment's realization of $\hat{\eta}$ is

$$\hat{\eta} = \frac{\bar{Y_A} - \bar{Y_B}}{\bar{Y_B}} = \frac{4131.33 - 3005.75}{3005.75} = .37$$

Hence producing a 95% confidence interval for $\frac{\hat{\eta}}{.25}$ is straightforward.

$$\frac{\eta}{.25} \in \left[\frac{.37}{.25} \pm 1.96(\sqrt{.64})\right]$$

$$\Rightarrow \frac{\eta}{.25} \in [.23, 2.37]$$

2. Logit MLE

a) Derive the score of the logit-likelihood: First, noting that

$$\frac{\partial \Lambda(X_i'\beta)}{\partial \beta} = \frac{\exp(X_i'\beta)X_i'(1 + X_i'\beta) - \exp(X_i'\beta)\exp(X_i'\beta)X_i'}{(1 + \exp(X_i'\beta))^2} = \frac{\exp(X_i'\beta)X_i'}{(1 + \exp(X_i'\beta))^2}$$

The sum of the individual score of the logit log-likelihood is therefore

$$\begin{aligned} s(\beta) &= \sum_i \frac{Y_i \exp(X_i'\beta)X_i'}{\Lambda(X_i'\beta)(1 + \exp(X_i'\beta))^2} - \frac{(1 - Y_i)\exp(X_i'\beta)X_i'}{(1 - \Lambda(X_i'\beta))(1 + \exp(X_i'\beta))^2} \\ &= \sum_i \Big(Y_i - (1 - Y_i)\exp(X_i'\beta)\Big)\frac{X_i'}{1 + \exp(X_i'\beta)} \\ &= \sum_i \Big(Y_i(1 + \exp(X_i'\beta)) - \exp(X_i'\beta)\Big)\frac{X_i'}{1 + \exp(X_i'\beta)} \\ &= \sum_i \Big(Y_i - \frac{\exp(X_i'\beta)}{1 + \exp(X_i'\beta)}\Big)X_i' \end{aligned}$$

b) The moment condition identifying β_{ML} is

$$E\left[\Big(Y_i - \frac{\exp(X_i'\beta)}{1+\exp(X_i'\beta)}\Big)X_i'\right] = 0$$
$$E\left[E[Y_i - \frac{\exp(X_i'\beta)}{1+\exp(X_i'\beta)}|X_i]X_i'\right] = 0$$

the interpretation here is that the "residual", that is, $E[Y_i] - \Lambda(X_i'\beta)$, is uncorrelated with X_i. $\Lambda(X_i'\beta)$ can be thought of as the "predicted value" for Y_i, since $P(Y_i = 1|X_i) = \Lambda(X_i'\beta)$.

c) The moment conditions identifying β_{NLLS} are

$$\begin{aligned} 0 &= E\left[-2(Y_i - \Lambda(X_i'\beta))\frac{\exp(X_i'\beta)}{(1+\exp(X_i'\beta))^2}X_i'\right] \\ &= E\left[E[(Y - \Lambda(X_i'\beta))|X_i]\frac{\exp(X_i'\beta)}{(1+\exp(X_i'\beta))^2}X_i'\right] \end{aligned}$$

d) Under which conditions does β_{NLLS} coincide with β_{ML}? The moment conditions will coincide when $E[Y_i - \Lambda(X_i'\beta)|X_i'] = 0$ for all X_i; that is, they coincide when the logit model is correctly specified, and $\Lambda(X_i'\beta)$ is the actual conditional expectation function of Y_i given X_i.

e) Under proper specification of the model, the asymptotic variance of the estimator is the additive inverse of the inverse of the expectation of the Hessian of the log likelihood.[1] As was derived in part a), the score (i.e. the gradient of the log-likelihood) is

$$s(X_i,\beta) = \left(Y_i - \frac{\exp(X_i'\beta)}{1+\exp(X_i'\beta)}\right)X_i'$$

Thus, the Hessian matrix is

$$\begin{aligned} \nabla_\beta s(X_i,\beta) &= -\frac{\exp(X_i'\beta)(1+\exp(X_i'\beta)) - \exp(X_i'\beta)^2}{(1+\exp(X_i'\beta))^2}X_iX_i' \\ &= -\frac{\exp(X_i'\beta)}{(1+\exp(X_i'\beta))^2}X_iX_i' \end{aligned}$$

And so, the asymptotic variance of β_{ML} under correct specification is given by

$$\sqrt{N}(\beta_{ML} - \beta) \xrightarrow{d} \mathcal{N}(0, H(\beta)^{-1})$$

with

$$H(\beta) = E\left[\frac{\exp(X_i'\beta)}{(1+\exp(X_i'\beta))^2}X_iX_i'\right]$$

[1] I just wanted to write that all out.

f) Let $V_s = E\left[s(X_i,\beta)s(X_i,\beta)'\right]$, the outer product of the score. Under mispecification, the asymptotic variance of β_{ML} is as follows:

$$\sqrt{N}(\beta_{ML}-\beta) \xrightarrow{d} \mathcal{N}\left(0, H(\beta)^{-1}V_sH(\beta)^{-1}\right)$$

with $H(\beta)$ defined as in the previous problem and

$$V_s = E\left[\left(Y_i - \frac{\exp(X_i'\beta)}{1+\exp(X_i'\beta)}\right)^2 X_iX_i'\right]$$

g) Suppose the data are independent across but not necessarily within clusters. Propose a cluster robust estimator of the asymptotic variance of β_{ML}

One idea is to construct a similar estimator to the one we have for cluster robust standard errors for OLS:

$$\frac{J}{J-K}\frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\exp(X_i'\beta)}{(1+\exp(X_i'\beta))^2}X_i'X_i\right)^{-1}\left(\frac{1}{N}\sum_{j=1}^{J}X_ju_ju_j'X_j'\right)\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\exp(X_i'\beta)}{(1+\exp(X_i'\beta))^2}X_i'X_i\right)^{-1}$$

where

$$u_j = \left[\left(Y_s - \frac{\exp(X_s'\beta)}{(1+\exp(X_s'\beta))^2}\right)\right]_{s\in j}.$$

That is, u_j is the vector of "residuals" for observations s in cluster j.

3. Matlab Probit DGP

a) (Matlab program attached)

b) The ML point estimates of $\hat{\beta}^{con} = (\hat{\alpha},\hat{b})$ are $(-0.0175, 0.9159)$. The standard errors are $(0.3030, 0.3607)$, respectively.

c) Score test: Following Wooldridge (2010), page 570, the LM statistic is the ESS from the following regression

$$\frac{\hat{u}_i}{\sqrt{\hat{G}_i(1-\hat{G}_i)}} = \alpha\frac{\hat{g}_i}{\sqrt{\hat{G}_i(1-\hat{G}_i)}}x_i + \gamma\frac{\hat{g}_i}{\sqrt{\hat{G}_i(1-\hat{G}_i)}}z_i$$

where x_i is the regressor matrix in the unconstrained regression (constant and X) and z_i is the vector of X_i^2. The ESS from this regression was 0.2963, which has a p-value of 0.58621. So, one does not reject the null that $b_2 = 0$.

d) The unrestricted model yields point estimates of $\hat{\beta}^{unc} = (\hat{\alpha},\hat{b},\hat{b_2}) = (-0.0435, 0.9175, 0.0428)$. The Wald test will test the null $g(\beta) = 0$ where

$$g(\beta) \equiv (0,0,1)\cdot\beta = b_2$$
$$G(\beta) \equiv \frac{\partial g(\beta)}{\partial\beta} = (0,0,1)$$

and the Wald statistic is given by

$$N \cdot \hat{b_2} \cdot \left(G \cdot \frac{1}{\sqrt{N}} H^{-1} \cdot G' \right)^{-1} \cdot \hat{b_2}$$

where H is the average Hessian at the ML estimate. Because we know that the model is correctly specified, I use the inverse Hessian instead of the sandwich estimator. The Wald evaluates to 7.4592, which has a p-value of 0.0063 under the χ^2 distribution with d.f. 1, so one rejects the null that $b_2 = 0$. This is starkly different from the score test result, which did not reject the null. This makes sense, as the Wald tends to reject more than the LM test. If one bumps the number of observations up (say, to 5000), both tests reject the null.

4. Clustered DGP in Stata

a) Regressing Y_{ic} on D_{ic}, we get the results in table 1.

Table 1: Clustered DGP

	(1) Y_ic, Regular SE	(2) Y_ic, Robust SE	(3) Y_ic, Clustered SE
d_ic	1.010***	1.010***	1.010***
	(0.03237)	(0.03244)	(0.09508)
Constant	0.0394*	0.0394*	0.0394
	(0.02278)	(0.02037)	(0.10761)
R^2	0.089	0.089	0.089
Observations	10000	10000	10000

Standard errors in parentheses
* p<0.10, ** p<0.05, *** p<0.01

The basic SE are 0.03237. The standard errors are slightly larger (0.03244) when we correct for heteroskedasticity in column 2 and significantly larger (0.09508) when we account for the intracluster correlation of observations from the same cluster.

b) Changing σ_η^2 to 0. Regressing Y_{ic} on D_{ic}, we get the results in table :

The standard errors are fairly similar for columns 1 and 2 to what we had in A. They are 0.02732 for the regular SE, 0.027731 for robust SE. The clustered SE (0.02420) are similar to the magnitude of the non-clustered version.

c) Comment on your differences in the answers to a) and b).
The SE are fairly similar in columns 1 and 2 for parts a) and b) though they are slightly smaller in part b as a result of the lower variance of y_{ic} in part b. In part b since there is no intra-cluster correlation in the coefficient on d_{ic} the clustered standard errors are of a similar magnitude to the non-clustered version (0.02420).

Table 2: Clustered DGP

	(1) Y_ic, Regular SE	(2) Y_ic, Robust SE	(3) Y_ic, Clustered SE
d_ic	0.992***	0.992***	0.992***
	(0.02732)	(0.02731)	(0.02420)
Constant	-0.00105	-0.00105	-0.00105
	(0.01911)	(0.01923)	(0.09650)
R^2	0.117	0.117	0.117
Observations	10000	10000	10000

Standard errors in parentheses

* p<0.10, ** p<0.05, *** p<0.01

d) In table 3, we see that the simulation rejects the null that the coefficient on d_{ic}=1 57% of the time when we don't cluster our standard errors and about 5% (which we would expect) when we cluster. So the non-clustered version is rejecting too often and the clustered version is rejecting at about the rate of p which we set.

Table 3: Monte Carlo Simulations

	(1)				
	count	mean	sd	min	max
reject	1000	.567	.4957386	0	1
reject_cluster	1000	.055	.2280943	0	1

e) Here we have that the regression with non-clustered standard errors rejects the test that the coefficient equals the average of the cluster level η's 3% of the time and the clustered standard errors are never rejecting, so in this case clustering is causing us not reject often enough.

Table 4: Monte Carlo Simulations

	(1)				
	count	mean	sd	min	max
reject	1000	.03	.1706726	0	1
reject_cluster	1000	0	0	0	0

f) When treatment is at the cluster level like in the case of the set up for part d then clustering standard errors will produce the accurate standard errors. However, in e, when the treatment is at the individual level but those individuals are within a cluster, for example a school or village then clustering the errors will actually cause our standard errors to be larger than we would want, leading to us to not reject enough. For the purpose of the internal validity of the study, using non-clustered standard errors when treatment is at the individual level is appropriate.

5. CPS and WLS

a) See results in column 1 of table 5.

Table 5: Wage Regression (Weighted)

	(1) Log wage	(2) Log wage (weighted by cell size)	(3) Log wage (weighted by 1/cell var)
agecat==2	0.314***	0.314***	0.314***
	(0.00746)	(0.00073)	(0.00047)
agecat==3	0.458***	0.458***	0.458***
	(0.00731)	(0.00071)	(0.00046)
agecat==4	0.494***	0.494***	0.494***
	(0.00771)	(0.00075)	(0.00048)
agecat==5	0.472***	0.472***	0.472***
	(0.00966)	(0.00094)	(0.00061)
agecat==6	0.295***	0.295***	0.295***
	(0.01651)	(0.00161)	(0.00104)
educcat==2	0.265***	0.265***	0.265***
	(0.00754)	(0.00073)	(0.00047)
educcat==3	0.415***	0.415***	0.415***
	(0.00764)	(0.00074)	(0.00048)
educcat==4	0.771***	0.771***	0.771***
	(0.00783)	(0.00076)	(0.00049)
Sex	-0.261***	-0.261***	-0.261***
	(0.00458)	(0.00045)	(0.00029)
Constant	2.153***	2.153***	2.153***
	(0.00995)	(0.00097)	(0.00063)
R^2	0.297	0.978	0.978
Observations	56182	56182	135056

Standard errors in parentheses

* p<0.10, ** p<0.05, *** p<0.01

b) Collapsed results used in analysis below.

c) See column 2 of table 5. The coefficients are identical (as we would expect) but the standard errors are much smaller when we weight by bin size.

d) If wages are *iid* within a cell, then the variance of each cell is σ^2/N_c where σ^2 is the population variance and N_c is the number of observations in the cell.

e) Formula above is used to calculate the variance of the cell means.

f) See results in column 3 of table 5.

g) With an RSS of 0.133 and 38 degrees of freedom we are not able to reject this model at the 5% level. In other words we cannot reject our model explains the data up to a sampling error.

h) See results in table 7.

i) With an RSS of 0.009 and 15 degrees of freedom we cannot reject this goodness of fit of this model at the 5% level either.

j) See fig. 1. The model does a very nice job of fitting the data. The place it has a bit of trouble fitting is for older more educated individuals. The model overestimates the wages of 65+ males with some college and underestimates the wages for males with a BA or more.

k) A simple model which passes the chi-squared test is just lnwage on sex. We cannot reject that this model explains all of the variation in lnwage up to a sampling error.

Table 6: Wage Regression (Weighted)

	(1) Log wage
agecat==2	0.301***
	(0.00103)
agecat==3	0.414***
	(0.00103)
agecat==4	0.424***
	(0.00103)
agecat==5	0.407***
	(0.00106)
agecat==6	0.355***
	(0.00068)
educcat==2	0.109***
	(0.00076)
educcat==3	0.238***
	(0.00083)
educcat==4	0.757***
	(0.00081)
Sex	-0.192***
	(0.00054)
<=25 × Dropout	0
	(.)
<=25 × HS	0.0576***
	(0.00078)
<=25 × Some College	0.00573***
	(0.00084)
<=25 × BA+	-0.135***
	(0.00087)
25-35 × Dropout	0.00900***
	(0.00084)
25-35 × HS	0.135***
	(0.00071)
25-35 × Some College	0.152***
	(0.00078)
25-35 × BA+	0
	(.)
35-45 × Dropout	-0.0677***
	(0.00084)
35-45 × HS	0.0784***
	(0.00070)
35-45 × Some College	0.154***
	(0.00078)
35-45 × BA+	0
	(.)
45-55 × Dropout	-0.0676***
	(0.00086)
45-55 × HS	0.112***
	(0.00071)
45-55 × Some College	0.158***
	(0.00079)
45-55 × BA+	0
	(.)
55-65 × Dropout	-0.00166*
	(0.00090)
55-65 × HS	0.0884***
	(0.00076)
R^2	0.998
Observations	135056

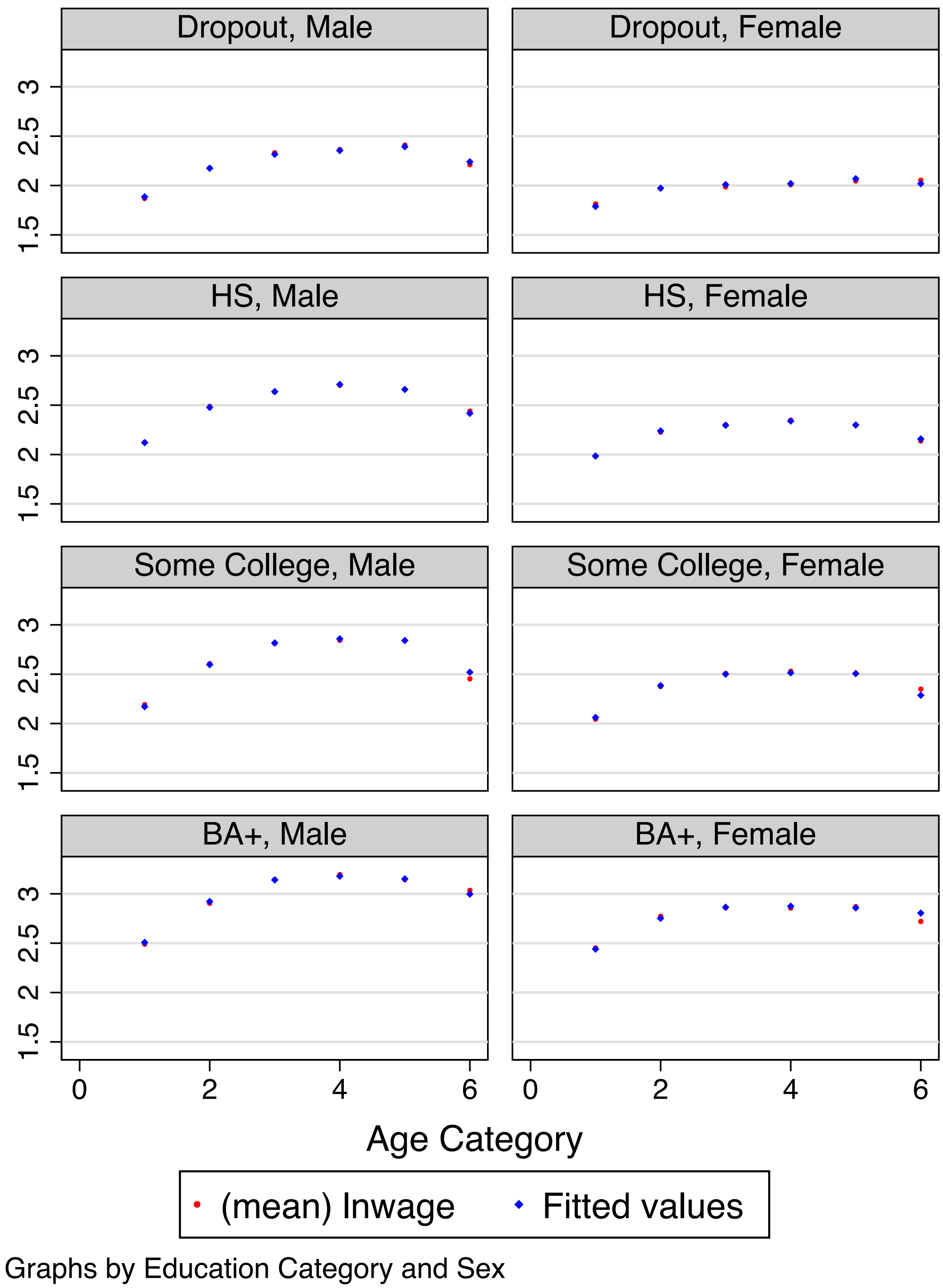
Standard errors in parentheses
* p<0.10, ** p<0.05, *** p<0.01

Table 7: Wage Regression (Weighted) cont

	(1) Log wage
55-65 × Some College	0.167***
	(0.00084)
55-65 × BA+	0
	(.)
>65 × Dropout	0
	(.)
>65 × HS	0
	(.)
>65 × Some College	0
	(.)
>65 × BA+	0
	(.)
<=25 × Male	0
	(.)
<=25 × Female	0.124***
	(0.00055)
25-35 × Male	-0.0212***
	(0.00054)
25-35 × Female	0
	(.)
35-45 × Male	0.0836***
	(0.00054)
35-45 × Female	0
	(.)
45-55 × Male	0.112***
	(0.00054)
45-55 × Female	0
	(.)
55-65 × Male	0.101***
	(0.00058)
55-65 × Female	0
	(.)
>65 × Male	0
	(.)
>65 × Female	0
	(.)
Dropout × Male	0
	(.)
Dropout × Female	-0.0302***
	(0.00026)
HS × Male	0.0672***
	(0.00020)
HS × Female	0
	(.)
Some College × Male	0.0410***
	(0.00020)
Some College × Female	0
	(.)
BA+ × Male	0
	(.)
BA+ × Female	0
	(.)
Constant	2.078***
	(0.00057)
R^2	0.998
Observations	135056

Standard errors in parentheses
* p<0.10, ** p<0.05, *** p<0.01

Figure 1: Predicted and actual log wages by subgroup



6. **Random Coefficient Binary Choice Model:**

a) As was derived in the lecture notes, the simulated log likelihood can be derived as follows. Denote the Logistic cdf with Λ and its pdf with λ.

First observe that

$$Pr(Y_i = 1|X_i, b_i) = Pr(\epsilon_i > -b_i x_i) = 1 - Pr(\epsilon_i < -b_i x_i) = 1 - \Lambda(-b_i x_i) = \Lambda(b_i x_i)$$

Thus, assuming that b_i and ϵ_i are independent:

$$Pr(Y_i = 1|X_i) = \int \Lambda(bx_i)\frac{1}{\sigma}\phi(\frac{b_i - \mu}{\sigma})db$$

The likelihood of this model for a single observation is

$$L(Y_i, X_i; \mu, \sigma) = \left(\int \Lambda(bx_i)\frac{1}{\sigma}\phi(\frac{b_i - \mu}{\sigma})db\right)^{Y_i}\left(1 - \int \Lambda(bx_i)\frac{1}{\sigma}\phi(\frac{b_i - \mu}{\sigma})db\right)^{1-Y_i}$$

Thus, for **fixed** random draws $\{u_i\}_{m=1}^{M}$ from $\mathcal{N}(0,1)$, the simulated likelihood is

$$\hat{L}_M(Y_i, X_i, \mu, \sigma) = \left(\frac{1}{M}\sum_{m=1}^{M}\Lambda(\underbrace{(\mu + \sigma u_{im})}_{\equiv b}X_i)\right)^{Y_i}\left(1 - \frac{1}{M}\sum_{m=1}^{M}\Lambda(\underbrace{(\mu + \sigma u_{im})}_{\equiv b}X_i)\right)^{1-Y_i}$$

Clearly, taking logs (of the product of all the likelihoods) allows us to recover the simulated log-likelihood of the full sample.

Let's differentiate with respect to μ and σ!

For μ:

$$\frac{\partial}{\partial\mu}\frac{1}{N}\sum_{i}^{N} Y_i log\left(\frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)\right) + (1 - Y_i)log\left(1 - \frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)\right)$$
$$= \frac{1}{N}\sum_{i}^{N}\left[\frac{Y_i X_i \frac{1}{M}\sum_{m=1}^{M}\lambda((\mu + \sigma u_{im})X_i)}{\frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)} - \frac{(1 - Y_i)X_i\frac{1}{M}\sum_{m=1}^{M}\lambda((\mu + \sigma u_{im})X_i)}{1 - \frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)}\right]$$

For σ:

$$\frac{\partial}{\partial\sigma}\frac{1}{N}\sum_{i}^{N} Y_i log\left(\frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)\right) + (1 - Y_i)log\left(1 - \frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)\right)$$
$$= \frac{1}{N}\sum_{i}^{N}\left[\frac{Y_i X_i \frac{1}{M}\sum_{m=1}^{M}\lambda((\mu + \sigma u_{im})X_i)u_{im}}{\frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)} - \frac{(1 - Y_i)X_i\frac{1}{M}\sum_{m=1}^{M}\lambda((\mu + \sigma u_{im})X_i)u_{im}}{1 - \frac{1}{M}\sum_{m=1}^{M}\Lambda((\mu + \sigma u_{im})X_i)}\right]$$

b) See *max_sml.m* file.

c) See *max_sml.m* file.

d) These results are invariant to initial conditions:

$$\begin{bmatrix} \mu \\ ln(\sigma) \end{bmatrix} = \begin{bmatrix} 1.6011 \\ 0.5139 \end{bmatrix}$$

e) Let $\hat{\theta}_{MSL}$ denote the estimated parameters from the method of simulated likelihood and the true parameters $\theta = [\mu, \sigma]'$. Under proper specification, we know that

$$\sqrt{N}(\hat{\theta}_{MSL} - \theta) \to \mathcal{N}(\mathbf{0}, \mathbf{H}(\theta)^{-1})$$

By the delta-method, with $g(\theta) \equiv [\mu, ln(\sigma)]'$, we have that

$$\sqrt{N}(g(\hat{\theta}_{MSL}) - g(\theta)) \to \mathcal{N}(\mathbf{0}, \mathbf{G}\mathbf{H}(\theta)^{-1}\mathbf{G})$$

where

$$\mathbf{G} \equiv \frac{\partial g(\theta)}{\partial \theta} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sigma} \end{bmatrix}$$

When we calculate the square root of the diagonal elements of $\frac{1}{N}\mathbf{H}(\hat{\theta})^{-1}$, we derive the standard errors of our estimates of μ and $\ln(\sigma)$: $(0.2633, 0.3884)$.

f) To calculate the standard errors for the incorrect model, we replace $\mathbf{H}(\theta)^{-1}$ with $\mathbf{H}(\theta)^{-1}\mathbf{V_s}\mathbf{H}(\theta)^{-1}$, where $\mathbf{V_s}$ is the population variance-covariance matrix of the scores—defined by the the criterion function we are maximizing.

Again, we use the analogy principle to replace the asymptotic variance terms with the simulated likelihood, sample averages. This yields standard errors of μ and $ln(\sigma)$: $(0.4839, 0.6417)$.

Appendix: Code for Various Problems

```
% Function that calculates the negative log-likelihood of data input

function negLogLikelihood = Q(Y,X,b)

    N = length(X);
    K = length(b);

    YHat = zeros(length(X),1);
    for k = 1:K
        YHat = YHat + X.^(k - 1) * b(k);
    end

    negLogLikelihood = - sum(...
        Y.*log(normcdf(YHat)) + (1-Y).*log((1 - normcdf(YHat))));

end
```

```
% Pset 2, question 3
% Group members: Christina Brown, Sam Leone, Peter McCrory, Preston Mui

% Set seed
rng(1234);
N = 500;

% Generate X and e
X = randn(N,1);
e = randn(N,1);
Y = (X + 0.1*X.^2 + e) > 0;

% Part B: estimate misspecified Probit by ML

    beta_con = fminunc(@(b)Q(Y,X,b),[0 0]);

    % calculate standard errors

    % score outer product (meat)
    Xmatrix = [ones(N,1),X];
    LHat = Xmatrix * beta_con';
    scoreproduct = zeros(2,2);
    for i = 1:N
        score_i = (Y(i) * normpdf(LHat(i)) / normcdf(LHat(i))...
            - (1 - Y(i)) * normpdf(LHat(i)) / (1 - normcdf(LHat(i))));
        scoreproduct = scoreproduct + score_i * score_i * Xmatrix(i,:)' * Xmatrix(i,:);
    end
    scoreproduct = scoreproduct / N;

    % average hessian (bread)
    hessian = zeros(2,2);
    for i = 1:N
        y = Y(i);
        phi = normpdf(LHat(i));
        Phi = normcdf(LHat(i));

        firstpart = y * phi * (LHat(i) * Phi + phi) / Phi^2;
        secondpart = (1 - y) * phi * (phi - LHat(i) * (1 - Phi)) / ((1 - Phi)^2);
        hessian_i = (firstpart + secondpart) * Xmatrix(i,:)' * Xmatrix(i,:);
        hessian = hessian + hessian_i;
    end
    hessian = hessian / N;

    % report standard errors
    Omega = (1/sqrt(N)) * inv(hessian) * scoreproduct * inv(hessian);
    disp(['Coefficients a, b: '])
    disp(beta_con(1))
    disp(beta_con(2))

    disp('Standard errors a, b: ')
    disp(sqrt(Omega(1,1)))
    disp(sqrt(Omega(2,2)))

% c: Score test on hypothesis that b_2 = 0
% The following follows Wooldridge, pg. 570

    ghat = normpdf(LHat);
    Ghat = normcdf(LHat);
```

```
    % LHS: u_i / sqrt(G * (1 - G))
    aux_lhs = (Y - Ghat) ./ sqrt(Ghat .* (1 - Ghat));

    % RHS: g_i / sqrt(G * (1 - G)) times x and z
    aux_RHS = bsxfun(@times,ghat ./ sqrt(Ghat .* (1 - Ghat)),[Xmatrix X.^2]);

    % Regress, obtain explained sum of squares
    aux_lhs_hat = aux_RHS * inv(aux_RHS' * aux_RHS) * aux_RHS' * aux_lhs;
    ESS = (aux_lhs_hat - mean(aux_lhs))' * (aux_lhs_hat - mean(aux_lhs));

    % The LM statistic has an asymptotic distribution of Chi squared 1 under the null
    disp(['LM Statistic: ' num2str(ESS)])
    disp(['P-value: ' num2str(1-chi2cdf(ESS,1))])

% d: Unconstrained models

    % Calculate unconstrained coefficients
    beta_unc = fminunc(@(b)Q(Y,X,b),[0 0 0])
    Xmatrix = [ones(N,1),X,X.^2];
    LHat = Xmatrix * beta_unc';

    % average (negative) hessian
    hessian = zeros(3,3);
    Phi = normcdf(LHat);
    phi = normpdf(LHat);

    for i = 1:N
        y = Y(i);
        phi = normpdf(LHat(i));
        Phi = normcdf(LHat(i));

        firstpart = y * phi * (LHat(i) * Phi + phi) / Phi^2;
        secondpart = (1 - y) * phi * (phi - LHat(i) * (1 - Phi)) / ((1 - Phi)^2);
        hessian_i = (firstpart + secondpart) * Xmatrix(i,:)' * Xmatrix(i,:);
        hessian = hessian + hessian_i;
    end
    hessian = hessian / N;

% Wald test
    g = beta_unc(1,3);
    gprime = [0 0 1];
    wald = N * g * inv((gprime * (1/sqrt(N)) * inv(hessian) * gprime')) * g;
    pvalue_d = 1 - chi2cdf(wald,1);
    disp(['P-value for Wald test on b2 = 0: ' num2str(pvalue_d)])
```

```
**** Problem Set 1 Econ 244
* Christina Brown

/*Preliminaries*/
clear
set more off
cap clear matrix
set mem 100m
set matsize 800

set seed 1234

cap log close
log using "$ps_244/log_`c(current_date)'.log", replace

******** Problem 4 ************************************************
***** Part a *****

    set obs 100

    gen eta_c=rnormal(1,1)
    gen v_c=rnormal(0,1)
    gen clusterid=_n

    expand 100

    gen e_ic=rnormal(0,1)
    gen dstar_ic=runiform()
    gen d_ic=(dstar_ic>.5)
    gen y_ic=v_c + eta_c*d_ic + e_ic

    eststo clear

    eststo: reg y_ic d_ic
    eststo: reg y_ic d_ic, r
    eststo: reg y_ic d_ic, cl(clusterid)

    esttab using "$ps_244/ps2_4a_regs.tex", b label starlevels(* 0.10 ** 0.05 *** 0.01) se(5) booktabs r2 obslast ///
        title("Clustered DGP") replace mtitles("Y_ic, Regular SE" "Y_ic, Robust SE" "Y_ic, Clustered SE")

***** Part b *****

    clear
    set obs 100

    gen eta_c=1
```

```
    gen v_c=rnormal(0,1)
    gen clusterid=_n

    expand 100

    gen e_ic=rnormal(0,1)
    gen dstar_ic=runiform()
    gen d_ic=(dstar_ic>.5)
    gen y_ic=v_c + eta_c*d_ic + e_ic

    eststo clear

    eststo: reg y_ic d_ic
    eststo: reg y_ic d_ic, r
    eststo: reg y_ic d_ic, cl(clusterid)

    esttab using "$ps_244/ps2_4b_regs.tex", b label starlevels(*
0.10 ** 0.05 *** 0.01) se(5) booktabs r2 obslast ///
        title("Clustered DGP") replace mtitles("Y_ic, Regular SE"
"Y_ic, Robust SE" "Y_ic, Clustered SE")

***** Part d *****

    set seed 1234

    cap program drop dgp
    program define dgp, rclass
        clear
        set obs 100

        gen eta_c=rnormal(1,1)
        gen v_c=rnormal(0,1)
        gen clusterid=_n

        expand 100

        gen e_ic=rnormal(0,1)
        gen dstar_ic=runiform()
        gen d_ic=(dstar_ic>.5)
        gen y_ic=v_c + eta_c*d_ic + e_ic

        reg y_ic d_ic, r
        test d_ic=1
        return scalar reject=(r(p)<.05)

        reg y_ic d_ic, cl(clusterid)
        test d_ic=1
        return scalar reject_cluster=(r(p)<.05)
```

```
    end

    simulate reject=r(reject) reject_cluster=r(reject_cluster), reps(1000): dgp

        eststo clear
        estpost summarize reject reject_cluster
        esttab using "$ps_244/ps2_4d_regs.tex", cells("count mean sd min max") title("Monte Carlo Simulations") replace noobs

***** Part e *****                      *** Need to fix to have eta_c's fixed

    set seed 1234

    clear
    set obs 100

    gen eta_c=rnormal(1,1)
    gen v_c=rnormal(0,1)
    gen clusterid=_n

    su eta_c
    global avg_eta_c=`r(mean)'
    di "`avg_eta_c'"

    expand 100

    cap program drop dgp2
    program define dgp2, rclass

        cap drop e_ic dstar_ic d_ic y_ic
        gen e_ic=rnormal(0,1)
        gen dstar_ic=runiform()
        gen d_ic=(dstar_ic>.5)
        gen y_ic=v_c + eta_c*d_ic + e_ic

        reg y_ic d_ic, r
        test d_ic=1.033752
        return scalar reject=(r(p)<.05)

        reg y_ic d_ic, cl(clusterid)
        test d_ic=1.033752
        return scalar reject_cluster=(r(p)<.05)

    end

```

```
    simulate reject=r(reject) reject_cluster=r(reject_cluster), reps(1000): dgp2

        eststo clear
        estpost summarize reject reject_cluster
        esttab using "$ps_244/ps2_4e_regs.tex", cells("count mean sd min max") title("Monte Carlo Simulations") replace noobs

******** Problem 5 ************************************************
***** Part a *****
use "$ps_244/cps2000.dta" if empstat==10, clear

        drop if incwage>999000|incwage==0 //those without valid earnings
        gen wage=incwage/(uhrswork*wkswork1) //average annual hourly wages
        drop if wage<2|wage>99  //drop outliers
        gen lnwage=ln(wage) //take logs

    *construct age categories
        gen agecat=1 if age<=25
        replace agecat=2 if age>25&age<=35
        replace agecat=3 if age>35&age<=45
        replace agecat=4 if age>45&age<=55
        replace agecat=5 if age>55&age<=65
        replace agecat=6 if age>65

    *construct education categories
        gen educcat=1 if educ99>=1&educ99<=9
        replace educcat=2 if educ99==10
        replace educcat=3 if educ99>=11&educ99<=13
        replace educcat=4 if educ99>=14

        lab var agecat "Age Category"
        lab var educcat "Education Category"

        lab def agecatlbl 1 "<=25" 2 "25-35" 3 "35-45" 4 "45-55" 5 "55-65" 6 ">65"
        lab val agecat agecatlbl

        lab def educcatlbl 1 "Dropout" 2 "HS" 3 "Some College" 4 "BA+"
        lab val educcat educcatlbl

    eststo clear
    eststo: xi: reg lnwage i.agecat i.educcat sex

***** Part b-f *****
```

```
    bys agecat educcat sex: gen count=_N
    su lnwage
    local var `r(Var)'
    gen cell_var=`var'/count
    gen cell_var_inv=1/cell_var

    collapse count lnwage cell_var cell_var_inv age educ99, by(agecat educcat sex)

        lab var agecat "Age Category"
        lab var educcat "Education Category"
        lab val agecat agecatlbl
        lab val educcat educcatlbl

    eststo: xi: reg lnwage i.agecat i.educcat sex [iweight=count]
    eststo: xi: reg lnwage i.agecat i.educcat sex [iweight=cell_var_inv]

        esttab using "$ps_244/ps2_5a.tex", b label starlevels(* 0.10 ** 0.05 *** 0.01) se(5) booktabs r2 obslast style(tex) ///
                title("Wage Regression (Weighted)") replace mtitles("Log wage" "Log wage (weighted by cell size)" "Log wage (weighted by 1/cell var)")

***** Part h *****
    eststo clear
    eststo: xi: reg lnwage i.agecat i.educcat sex agecat#educcat agecat#sex educcat#sex [iweight=cell_var_inv]
    predict lnwage_hat

        esttab using "$ps_244/ps2_5h.tex", b label starlevels(* 0.10 ** 0.05 *** 0.01) se(5) booktabs r2 obslast style(tex) ///
                title("Wage Regression (Weighted)") replace mtitles("Log wage" "Log wage (weighted by cell size)" "Log wage (weighted by 1/cell var)")

    twoway (scatter lnwage agecat, msize(tiny) mcolor(red)) (scatter lnwage_hat agecat, msize(tiny) mcolor(blue)), scheme(s1mono) by(educcat sex, cols(2)) xsize(3)
    graph export "$ps_244/ps2_5j.pdf", replace


cap log close
```

```
%--------------------------------------------------------------------------------
% Preliminaries
%--------------------------------------------------------------------------------
clear all
close all

% Specify globals data and V to be used in the function call of SML
global data V

% Set the seed for the random draws u_im. Grab V
rng(1234)
V = randn(2000,1000);

% Load data
load('logit.mat');
[N,~] = size(data);

%--------------------------------------------------------------------------------
% Unconstrained maximization using fminco
%--------------------------------------------------------------------------------
% Specify starting guess of optimal mu and ln_sigma
x0 = [1,1];
options = optimoptions('fminunc','Algorithm','trust-region', ...
                       'SpecifyObjectiveGradient',true);
fun = @SML;
[x_min, ~, ~, ~,~,x_hessian] = fminunc(fun,x0,options)

mu_val_est = x_min(1)
ln_sig_est = x_min(2)

% Problem 6e, standard errors under correct specification
G = [1,0;0,(1/x_min(2))];
correct_specification_se = sqrt(diag((1/N)*G*inv(x_hessian)*G))

% Problem 6f, standard errors under incorrect specification
% Construct vector of scores
[sim_avg_cdf, sim_avg_pdf, sim_avg_pdf_u_im,X,Y]...
    = sim_avg_vals(mu_val_est,ln_sig_est);

score_v = [(Y.*X.*sim_avg_pdf)./(sim_avg_cdf) -...
           ((1-Y).*X.*sim_avg_pdf)./(1-sim_avg_cdf),...
           (Y.*X.*sim_avg_pdf_u_im)./(sim_avg_cdf) -...
           ((1-Y).*X.*sim_avg_pdf_u_im)./(1-sim_avg_cdf)];

V_s = (1/length(score_v))*score_v'*score_v;

incorrect_spec_se = sqrt(diag((1/N)*G*inv(x_hessian)*V_s*inv(x_hessian)*G))
```

```
function [sml sml_gradient] = SML(x)
    mu_val = x(1);
    ln_sigma_val = x(2);

    global data V

    [sim_avg_cdf sim_avg_pdf sim_avg_pdf_u_im,X,Y,sigma_val]...
    = sim_avg_vals(mu_val,ln_sigma_val);

    % Calculate the negative of the SML:
    sml = - mean(Y.*log(sim_avg_cdf) + (1-Y).*log(1 - sim_avg_cdf));

    % Calculate the gradient
    % Note: dF/dx = dF/dlog(x) * dlog(x)/dx == > dF/dlog(x) = dF/dx * x
    if nargout > 1
    sml_gradient = -[mean((Y.*X.*sim_avg_pdf)./(sim_avg_cdf) -...
                    ((1-Y).*X.*sim_avg_pdf)./(1-sim_avg_cdf));...
                    mean((Y.*X.*sim_avg_pdf_u_im)./(sim_avg_cdf) -...
                    ((1-Y).*X.*sim_avg_pdf_u_im)./(1-sim_avg_cdf))*sigma_val];
    end

end
```

```
function [sim_avg_cdf sim_avg_pdf sim_avg_pdf_u_im,X,Y,sigma_val]...
    = sim_avg_vals(mu_val,ln_sigma_val)

    global data V
    % Number of Observations
    [N, ~] = size(data);

    %X and Y Values
    Y = data(:,1);
    X = data(:,2);

    % Number of simulations
    [~, M] = size(V);

    % Sigma Value
    sigma_val = exp(ln_sigma_val);

    % Value of Simulated Maximum Likelihood
    logistic_input = mu_val*ones(N,M) + sigma_val*V;
    logistic_input = bsxfun(@times,logistic_input,X);

    % Calculate the value of logistic cdf and pdf
    logist_cdf = cdf('Logistic',logistic_input,0,1);
    logist_pdf = pdf('Logistic',logistic_input,0,1);

    % Calculate the average value of the cdf and pdf across simulations.
    % This is used in the numerators and denominators of the SML and gradient
    sim_avg_cdf = mean(logist_cdf,2);
    sim_avg_pdf = mean(logist_pdf,2);
    sim_avg_pdf_u_im = mean(logist_pdf.*V,2);
end
```