Computing Equilibrium in a Model with School Competition and Educational Vouchers

Dennis Epple and Richard E. Romano

June 18, 2007

Introduction

These notes summarize the approach to computing equilibrium in our model of private and public school competition.[1] The program is written for Gauss and is named School_Competition.gss. It is a text file and can be edited by any text editor. The model in this program is solved using the algorithm in Nlinsys1.dne. Before running School_Competition.gss, do a search and replace. Replace the following which appears five times in the program:
d:\myfolder
with the drive and folder on your computer to which you wish your output directed. In addition, replace
d:\gssmastr
with the drive and folder where you placed file Nlinsys1.dne.

The program is currently set to solve for equilibrium with six private schools and no voucher. To solve for equilibrium with a flat rate voucher, set *doact=1* where indicated near the beginning of the program. The program will then solve for equilibria for vouchers up to $2,000. As explained below, the program is calibrated assuming a half student per household (the US average). Hence, the $2,000 voucher in the program is per half student and corresponds to a per student voucher of $4,000.

Model

Let J denote the number of schools, and index schools in order of ascending quality. Let k, θ, η, and I be, respectively, the vectors of school sizes, peer qualities, shadow prices of ability, and inputs. Let v and t respectively denote the voucher and tax rate. Place [k, θ, η, I, v, t] into a (4J+2) vector denoted x. Equilibrium is computed by solving (4J+2) nonlinear simultaneous equations for a fixed point, x. We now turn to defining these equations.

To define the shadow value of peer quality, we will need the slope of an indifference curve in the (θ,p) plane. The utility function of type (b,y):

1. $U=(y(1-t)-p)\theta^{\gamma}I^{\omega}b^{\beta}$

The slope of an indifference curve in the (θ,p) plane is given by:

[1] The model is in “Educational Vouchers and Cream Skimming,” and these notes use the notation there.

2. $$\left.\frac{\partial p}{\partial \theta}\right|_{U=\bar{U}} = \frac{\gamma(y(1-t)-p)}{\theta}$$

We will also need the boundary loci that delineate admission spaces. The boundary loci are derived from the indifference loci. The indifference locus between schools i and j is the locus of types indifferent between the schools when price in each equals effective marginal cost. An indifference locus is then:

3. $$\begin{gathered}(y(1-t)+v-emc_i(b))q_i b^{\beta} = (y(1-t)+v-emc_j(b))q_j b^{\beta} \\ \Rightarrow \tilde{y}_{ij}(b,x) = \frac{(emc_i(b)-v)q_i-(emc_j(b)-v)q_j}{(q_i-q_j)(1-t)} \quad i,\, j=1,\ldots,J-1;\, i\neq j\end{gathered}$$

where effective marginal cost is:

4. $$emc_j(b) = V'(k_j)+I_j+\eta_j(\theta_j-b)$$

The boundary loci are then determined by:[2]

5. $$\begin{gathered}y_0(b,x)=y_{\min}; \quad y_J(b,x)=y_{\max} \\ y_i(b)=\max[y_{i-1}(b), \min{}_{j>i}\tilde{y}_{ij}(b)] \quad i=1,\ldots,J-1\end{gathered}$$

where the support of y is $[y_{min}, y_{max}]$.

School sizes and peer qualities are:

6. $$k_j = \int_0^{\infty}\int_{y_{j-i}(b,x)}^{y_j(b,x)} f(y,b)dbdy \quad j=1,\ldots,J$$

7. $$\theta_j = \frac{1}{k_j}\int_0^{\infty}\int_{y_{j-i}(b,x)}^{y_j(b,x)} bf(y,b)dbdy \quad j=1,\ldots,J$$

Using (2), the shadow price of peer quality is:

8. $$\eta_j = \int_0^{\infty}\int_{y_{j-i}(b,x)}^{y_j(b,x)} \frac{\gamma[y(1-t)-p_j(b,y)]}{\theta_j} f(y,b)dbdy \quad j=1,2,\ldots,J,\ j\neq pub; \quad \eta_{pub}=0$$

Approximating using price equal to effective marginal cost, we substitute (4) into (8):

9. $$\begin{aligned}\eta_j &= \frac{1}{k_j}\int_0^{\infty}\int_{y_{j-i}(b,x)}^{y_j(b,x)} \frac{\gamma[y(1-t)-V'(k_j)-\eta_j(\theta_j-b)-I_j]}{\theta_j} f(y,b)dbdy \\ &= \frac{\gamma(1-t)}{k_j\theta_j}\int_0^{\infty}\int_{y_{j-i}(b,x)}^{y_j(b,x)} y\ f(y,b)dbdy - \frac{\gamma[V'(k_j)+I_j]}{\theta_j} \quad j=1,2,\ldots,J,\ j\neq pub; \quad \eta_{pub}=0\end{aligned}$$

The first-order condition for inputs is:

10. $$\eta_j = \frac{\gamma I_j}{\omega\theta_j} \Rightarrow I_j = \frac{\omega\eta_j\theta_j}{\gamma} \quad j=1,2,\ldots,J,\ j\neq pub; \quad I_{pub}=I_{exog}$$

[2] See the appendix to these notes for details.

Public school expenditure, I_{pub} is set exogenously. The voucher is set to an exogenously determined value v_{exog}:

11. $v = v_{exog}$

The tax rate must pay the cost of public education plus the voucher. The number of public schools, m, is the integer value that minimizes the total cost of serving the public school population.

12. $$t\overline{y} = k_{pub} I_{pub} + \min_m \left[m \left(F + V \left(\frac{k_{pub}}{m} \right) \right) \right] + v(1 - k_{pub}) \quad \textit{for } m \in \{1, 2, ...\}$$

The system of equations to be solved is then given by (6), (7), (9), (10), (11), and (12).

We now rewrite the integrals in (6), (7), and (9) to facilitate computation.

Substitute $f(b,y) = f_1(b) f_2(y/b)$ into (6):

13. $$k_j \;=\; \int_0^{b_x} f_1(b) \int_{y_{j\text{-}1}(b,x)}^{y_j(b,x)} f_2(y/b)\; dy db$$

We assume that the [ln(y),ln(b)] is normally distributed:

14. $$\begin{bmatrix} \ln y \\ \ln b \end{bmatrix} = N \left[\begin{pmatrix} \mu_y \\ \mu_b \end{pmatrix}, \begin{pmatrix} \sigma_y^2 & \rho \sigma_y \sigma_b \\ \rho \sigma_y \sigma_b & \sigma_b^2 \end{pmatrix} \right]$$

Then rewrite (13) as:

15. $$k_j \;=\; \int_0^{\infty} f_1(b) \left[\Phi \left(\frac{\ln(y_j(b,x)) \text{ - } \mu_y(b)}{\sigma_{y/b}} \right) - \Phi \left(\frac{\ln(y_{j-1}(b,x)) \text{ - } \mu_y(b)}{\sigma_{y/b}} \right) \right] db$$

where $\mu_y(b) = \mu_y - \rho(\sigma_y / \sigma_b)\; (\ln \; (b) - \mu_b)$ and $\sigma_{y/b} = \sigma_y (1-\rho^2)^{.5}$

To simplify computation we let:

16. $$K_j \;=\; \int_0^{\infty} f_1(b) \Phi \left(\frac{\ln(y_j(b,x)) \text{ - } \mu_y(b)}{\sigma_{y/b}} \right) db$$

Then, from (15) and (16):

17. $k_j = K_j - K_{j-1}$

Using the same approach to rewrite (7), we let:

18. $$\Theta_j \;=\; \frac{1}{k_j} \int_0^{\infty} b f_1(b) \Phi \left(\frac{\ln(y_j(b,x)) \text{ - } \mu_y(b)}{\sigma_{y/b}} \right) db$$

Then

19. $\theta_j = \Theta_j - \Theta_{j-1}$

Similarly, the integral of after-tax income is:

22.
$$Y_j = (1-t)\int_0^\infty \int_0^{y_j(b,x)} y\ f(y,b)dydb$$
$$= (1-t)e^{\mu_y(b)+\frac{\sigma^2_{y/b}}{2}} \int_0^\infty\ f_1(b)\Phi\left(\frac{\ln\ (y_j(b,x)) - \mu_y(b) - \sigma^2_{y/b}}{\sigma_{y/b}}\right)db$$

We rewrite (9) as:

23. $$\eta_j = \frac{\gamma(1-t)(Y_j - Y_{j-1})}{k_j\theta_j} - \frac{\gamma[V'(k_j) + I_j]}{\theta_j} \quad j = 2,\ldots,J; \quad \eta_1 = 0$$

Program

We will use superscript v to denote a vector and m for a matrix. Let $b^v = [b_1, b_2, \ldots, b_L]$ be a row vector of L ordinates to be used for numerical integration. Integration is by Gauss-Legendre quadrature. We first discuss evaluation of the integral in (17).

Let $\Gamma_j(b)$ be the integrand in (17). Using Simpson's rule, the integral will be approximated by the sum of L rectangles. Rectangle i has height $\Gamma_j(b_i)$. Let w_i be the width of the base of the rectangle, hence the area of the rectangle is $\Gamma_j(b_i)w_i$. Let $w^v = [w_1, w_2, \ldots, w_K]$ be the column vector of widths associated with b^v. In the program w^v=*(diffb/2)*wvec*. [3]

Then the integral in (17) is approximated by:

24. $$\sum_{i=1}^{L} \Gamma_j(b_i)\cdot w_i = \Gamma_j(b^v)\cdot w^v$$

The integrals for the J schools can be evaluated more compactly by placing the $\Gamma_j(b^v)$ into a JXL matrix:

25. $$\Gamma^m = \begin{bmatrix} \Gamma_1(b^v, x) \\ \ldots \\ \Gamma_J(b^v, x) \end{bmatrix}.$$

The vector K^v is then obtained as:

26. $K^v = \Gamma^m \cdot w^v$.

We now detail calculation of the integrand in (17). We begin with indifference loci: $\tilde{y}^v_j = \tilde{y}_j(b^v, x)$ is a *1XL* row vector of points on the indifference locus between schools j and $j+1$. Similarly, $y^v_j = y_j(b^v, x)$ is a *1XL* row vector of points on the boundary locus $y_j(b,x)$. Procedure (proc) *bcross* calculates the indifference loci (denoted *iloc*) and

---

[3] The vector of weights *wvec* is normalized to sum to 2. Thus, when the integration is over an interval of width different from 2, the *wvec* are multiplied by the one half the width of the support of the variable over which integration is to be conducted. In this application, we set *diffb* = $(b_{max}-b_{min})$ = (150,000 - .0000001).

the boundary loci (denoted *yofb*) and returns a *JXL* matrix of upper boundary loci of the *J* schools:

27. $$y^m = \begin{bmatrix} y_1(b^v, x) \\ \ldots \\ y_J(b^v, x) \end{bmatrix}.$$

Taking y^m as input, procedure *kproc* calculates k^v. We denote as K^v the *J*-dimensional vector of K_j in (23). In *kproc*, the following three 1XL row vectors are calculated. The names used in the program to denote these vectors are indicated in parentheses:

$\mu_y(b^v) = \mu_y - \rho(\sigma_y / \sigma_b)\ (\ln\ (b^v) - \mu_b)$ (muyofb)

$\phi(\ln b^v) = \dfrac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{(\ln(b^v)-\mu_b)^2}{2\sigma_b^2}}$

(bdenb)

$f_1(b^v) = \phi(\ln b^v)./\,b^v$ (denb)

where ./ denotes element-by-element division. The following JXL dimensional matrix then corresponds to Γ^m in (25):

$$f_1(b^v).*\Phi\left(\frac{\ln\ (y^m) - \mu_y(b^v) - \sigma^2_{y/b}}{\sigma_{y/b}}\right)$$

where $\Phi\left(\dfrac{\ln\ (y^m) - \mu_y(b^v) - \sigma^2_{y/b}}{\sigma_{y/b}}\right)$ is a JXL matrix, and .* denotes element-by-element multiplication. The vector K^v is then:

28. $$K^v = f_1(b^v).*\Phi\left(\frac{\ln\ (y^m) - \mu_y(b^v) - \sigma^2_{y/b}}{\sigma_{y/b}}\right)*w^v \qquad \text{(cumk)}$$

Similarly:

29. $$\Theta^v = \phi(\ln b^v).*\Phi\left(\frac{\ln\ (y^m) - \mu_y(b^v) - \sigma^2_{y/b}}{\sigma_{y/b}}\right)*w^v \qquad \text{(cumth)}$$

From these, procedure *kproc* calculates the J-dimensional vectors k^v (denoted *kvout*) and θ^v (denoted *thvout*) using (18) and (20).

Implementing the analogous approach with Equation (22), we obtain the vector of after-tax incomes:

30. $$Y^v = (1-t)e^{\mu_y(b^v)+\frac{\sigma^2_{y/b}}{2}}.*\ f_1(b^v)\Phi\left(\frac{\ln\ (y_j(b^v, x)) - \mu_y(b^v) - \sigma^2_{y/b}}{\sigma_{y/b}}\right)*w^v \quad \text{(cuminc)}$$

Using this result Equation (23), *kproc* calculates the vector η^v (etaout). Procedure *kproc* returns the 3-dimensional column vector [k^v, θ^v, η^v].

Procedure *func* defines the equations to be solved, naming the resulting *3J+2* vector *fun*. Proc *func* calls the proc's discussed above (*bcross* and *kproc*). The first 3J equations in *fun* correspond to (6), (7), and (8). The next *J* equations correspond to (10). The final two equations correspond to (11) and (12).

Proc *func* is called by the algorithm for solving nonlinear simultaneous equations. This algorithm is in file Nlinsys1.dne. When a solution to the equations is found for a given number of schools, profit is calculated for each private school. If the sum of profits is positive, entry occurs. Proc *newx0* is called, and new starting values are created. The starting values are based on the solution just obtained.

Profits are calculated by Monte Carlo simulation in procedure *profn*. A large sample *(y,b)* is drawn. Sample size is set by *smpl*. This calculation requires determining the price that a school charges to each student. That price equates utility in the chosen school to utility at *p=emc* in the next-best alternative school. Hence, in *profn*, utility of each element of the sample is calculated in every school with price set equal to effective marginal cost. The utility of each type in the type's next-best alternative school is calculated and placed in vector *ordstat2*. The price charged to each type is then calculated, followed by calculation of profit for each school.

Welfare and achievement in an equilibrium are compared to those in a benchmark allocation using Monte Carlo simulation in procedure *welfn*. For each *(y,b)* type, utility and achievement are calculated in the benchmark allocation. Compensating variation and achievement for the equilibrium being studied are then calculated relative to the benchmark allocation.

Computation

The algorithm uses Newton's method. Newton's method is fast, but can be fragile. The fragility arises in part because Newton's method may take a step sufficiently large that it moves to a portion of the parameter space where one or more functions is undefined, or one or more of the functions is so "flat" that the matrix of derivatives is singular. This fragility tends to become more pronounced when there are a large number of schools. The program contains some features that help facilitate convergence.

1. Constraining Parameter Values

In the course of searching, numerical search routines may choose parameter values that violate constraints on the signs or magnitudes of parameter values. It is useful to transform parameters to impose constraints that parameters must satisfy. Three types of constraints are imposed in our program. Each type is illustrated below.

Suppose a parameter must be non-negative, such as the θ_j in our model. Let x_u denote a parameter that must be non-negative. Let

$$x_t = \mathrm{Ln}(x_u)$$

Conduct the numerical search over parameter x_t, which can take any values on the real line. Regardless of the value of x_t that is chosen, x_u will be non-negative. We refer to x_t as the transformed parameter and x_u the untransformed parameter (i.e., the parameter of interest). In our program, we use the above to assure that θ_j, I_j, and the tax rate are non-negative.

Suppose a parameter must lie in the interval (0,1), such as the k's in our model. Let x_u be a parameter that must satisfy such a constraint. Then the appropriate transformed parameter is:

$$x_t = \mathrm{Ln}(x_u / 1 - x_u))$$

For any value of x_t on the real line, x_u satisfies the desired constraints. In our program, we use the above transformation to assure that all k_j, and the η_j for private schools, are in the interval (0,1).

The proc *trans* performs the transformations described above. The proc *untrans* reverses the transformations.

2. Adjusting Step Size

The program contains a parameter named *nlfac*. This parameter is the fraction of a Newton step that is taken on a given iteration. For example, if *nlfac*=.05, then the step that is actually taken is 5% of the Newton step. After each 100 function evaluations, the program adjusts the step size to *nlfac=nladj*nlfac*, where *nladj* is greater than one. For example, if *nladj =1.05*, then *nlfac* will be increased by a factor of 1.05 after each 100 function evaluations, until *nlfac* reaches a value of 1.0.

3. Incremental change

If it is possible to move incrementally from a known equilibrium to a desired equilibrium, this is often the best way to proceed. The do loop near the end of the program can be used to do such calculations. For example, the do loop is currently set to incrementally increase the voucher. (This do loop is activated by setting *doact=1* near the beginning of the program.) This do loop uses the result on each cycle through the loop to start the next cycle. The do loop can be amended to incrementally change other features of the model. For example, holding the voucher constant, one might want to explore the effects of varying a cost function parameter (e.g. *fixcst*). This can be done by incrementing the parameter while holding the voucher constant.

Dictionary of Variables

Variable	Symbol Above	Program Name
Number of Schools	J	nsch
# Ordinates for Integration	L	ordnum
Exogenous voucher setting	v_x	vouex
Exogenous public expenditure	I_{pub}	iexog
Lower support of y	y_{min}	ylolim
Upper support of y	y_{max}	yuplim
Lower support of b	b_{min}	blolim
Upper support of b	b_{max}	buplim

Miscellaneous

1. In the U.S., there is approximately one student for each two households. Thus, the program assumes one half student per household. Price, average cost, marginal cost, and the voucher are then per half student and must be multiplied by two to convert to a per student basis in year 2000 dollars.

2. The "custodial" cost function is quadratic:

C(k)= F+V(k) = fixcst +cfac2*kv +cfac*kv^2/2

School Boundary Loci when Indifference Loci Cross

Let $I_{ij}(b)$ be the locus of households indifferent between schools i and j (which may, but need not, be adjacent in quality). Let $y_i(b)$ be the upper boundary locus of school i. Let schools be indexed in ascending order of quality.

If indifference loci don't cross,

1. $y_i(b) = I_{i,i+1}(b)$.

If indifference loci cross, then the $y_i(b)$ can be obtained by solving the following in ascending order of school quality:

2. a) $y_0(b) = \underset{j>0}{\mathrm{Min}}\ I_{0j}(b)$

b) $y_i(b) = \mathrm{Max}\ [y_{i-1}(b), \underset{j>i}{\mathrm{Min}}\ I_{ij}(b)] \quad i > 0$

Figure 1 illustrates Equations (2) with a three-school example. The admission space of School 0 is region OABC. The admission space of School 1 is ABD. The admission space of School 2 is the region above CBD. Equation (2) includes line segment BC as part of the admission space of School 1. This is innocuous since this line segment has zero measure, but it is convenient computationally to include this line seqment as part of School 1.

The following results provide the foundation for Equations (2).

Result 1: No point in the interior of School i can be on the indifference locus between School i and any higher-theta school.

Proof: Let (b',y') in Figure 2 be interior to School i and on the indifference locus between School i and School k with k > i. Let (b',y") be in the neighborhood of (b',y') with y" > y'. Then (b',y") prefers School k to School i contradicting that (b',y') is in the interior of i. ∎

It follows that an indifference locus between School i and School k with k > i must lie on or above the upper boundary locus of School i.

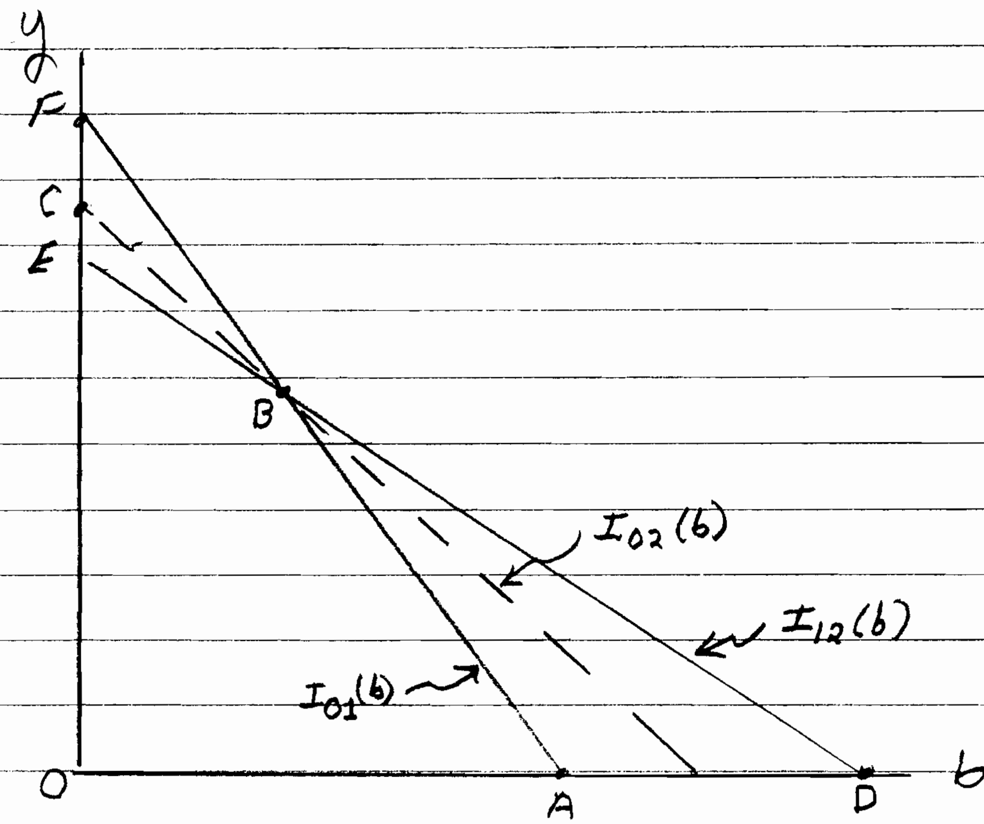
Result 2: A point on the upper boundary locus of School i cannot lie below all indifference loci between School i and School k, k > i.

Proof: Let A denote a point (household) on the upper boundary locus of School i. Now A must at least weakly prefer School i to all others to be on the upper bounary of i. If the preference is strict, then we have a contradiction to the claim that A is on the boundary of i. Hence, A is indifferent between i and some higher-theta school. ■

The above two claims establish that any point on the the upper boundary locus of School i is on the lower envelope of the indifference loci between School i and all higher schools, i.e., on $\min_{j>i} I_{ij}(b)$.

The lower envelope of the indifference loci between School i and all higher schools may also contain points that are not on the upper indifference locus of School i. For example, the lower envelope of the indifference locus between School 1 and School 2 in Figure 1 includes segment BE. Points on BE are interior to School 0. This follows from a single-crossing argument similar to that in Figure 2. Since any point on BF is indifferent between Schools 0 and 1, a point on BC with the same ability and lower income will prefer School 0 to School 1. Similarly, any point on BE with the same ability will have lower income still and will prefer School 0 to School 2. This logic applies to show that any point on BE strictly prefers School 0 to either School 1 or School 2. In short, line segment BE is not on the upper boundary locus of School 1 or any other school. The Max operator in Equation (2b) removes this line segment by requiring that the upper boundary locus of School 1 lie on or above the upper boundary locus of School 0.

Figure 1



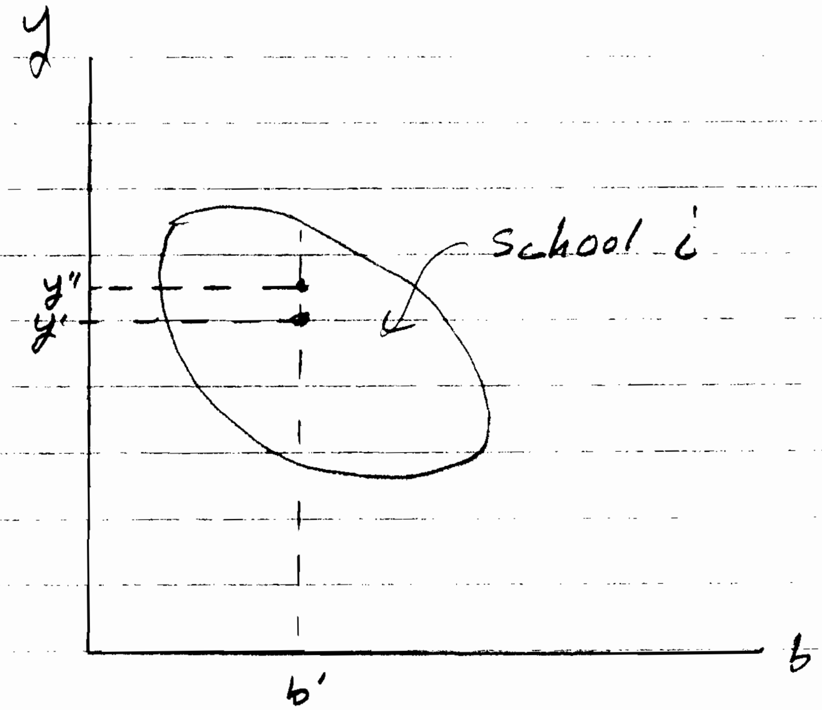


Figure 2