Spring 2023

Gregory Bruich, Ph.D.
Dept. of Economics, Harvard University

Lab 2: Measuring Upward Mobility Using the National Longitudinal Survey

Methods/concepts: predicted values, predicted effects, conditional probabilities, unconditional probabilities, statistical models for small data sets

LAB DESCRIPTION

This lab uses an extract from the National Longitudinal Survey of Youth called **nlsy97.dta** to quantify intergenerational mobility. For more details on the variables included in these data, see Table 1. A list and description of each of the Stata and R commands needed for this lab are contained in Table 2 and Table 3, respectively.

You will explore how mobility statistics estimated in this sample of 5,486 individuals compare with the same statistics calculated in Chetty et al. (2014) and Chetty et al. (2017) using full population tax data. You will then use the National Longitudinal Survey data to explore differences in social mobility by race and gender. Finally, you will explore why mobility statistics based on a statistical model are especially useful for characterizing the geography of upward mobility across areas.

QUESTIONS

1. In historical data (Card et al. 2018; Derenoncourt 2019) and developing countries (Alesina et al. 2021, Asher et al. 2021), intergenerational mobility has been measured using data on educational attainment rather than parent and child income. For example, Alesina et al. (2021) define upward mobility in Africa as the fraction of children who complete primary school if their parents did not. Educational mobility is a useful place to start to remind us of some of the tools we learned in Lab 1.

    a. Using the **nlsy97.dta** data, what *fraction* of children whose mothers had a high school education or less went on to receive a college degree or higher? *Hint:* Calculate the arithmetic average of the *indicator variable* *child_college* for observations with *mother_education* less than or equal to 12. You'll use this trick again in question 5c,d.

    b. Using data from the Census Bureau for a much larger sample, I calculate that 20.9% of children whose mothers had a high school education or less went on to receive a college degree or higher (Online Table 7, Chetty et al. (2018)). In your judgement, is your estimate close to 20.9%?

2. Now we will start by generating a few new variables, following what we did in Lab 1:

    a. Generate percentile ranks for *kid_income*, normalized so that highest rank is 100

    b. Generate percentile ranks for *parent_inc*, normalized in the same way

3. Visualize the relationship between child and parent income ranks in two ways, commenting on which is a more useful summary of the data and including your graphs in your solutions:

    a. Scatter plot of the individual level data

    b. Bin scatter plot

4. Estimate a *linear regression* of kid income ranks on parent income ranks. What is the intercept in this regression? What is the estimated slope? (For today's lab, don't worry about the standard errors, R^2, or anything else, other than the values of the estimated coefficients.)

5. Compare the following measures of upward mobility in these survey data with those calculated using full population tax data (Chetty et al. 2014). Refer back to the Lab 2 video as needed.

    a. **Statistic 1:** Predicted child income rank from the rank-rank regression in question 4 evaluated at $Rank_{parent} = 25$, which Chetty et al. (2014) report as 41.3.

    b. **Statistic 2:** Relative Mobility, which Chetty et al. (2014) report as 34.1.

    c. **Statistic 3:** Probability that a child born to parents in the bottom fifth of the income distribution reaches the top fifth of the income distribution, which Chetty et al. (2014) report as 7.5%.

    d. **Statistic 4:** fraction of children who make more in (inflation adjusted) dollars than their parents, which Chetty et al. (2017) report as 50% for children born in the 1980s. To adjust for inflation, note that $1 in 1997 would be worth $1.4767 in 2015.

6. Repeat your calculations in (4)-(5) separately for Black men (i.e., `female == 0 & black == 1`) and White men (i.e., `female == 0 & white == 1`). Is mobility higher for Black men or White men? Does it depend on what statistic you use? Explain.

7. Why is Statistic 1: Absolute Mobility at the 25th Percentile estimated using a "linear statistical model" for purposes of constructing the Opportunity Atlas? This question is meant to help demonstrate the key advantage of the linear statistical model: by finding the central tendency of the data, this method provides precise, stable estimates even in small samples. Professor Chetty illustrated this lesson using the graph shown in Figure 1 below. Include your answers to 5a and 7a-d **both** in your lab write up **and** submit them also to this Google Form.

    a. Instead of using a regression, calculate the simple arithmetic mean of $Rank_{child}$ for children with $Rank_{parent}$ between 20 and 30 in the full NLS 1997 data. How does this "binned average" compare with statistic 1 that you calculated in question 5a?

       In Figure 1 from Professor Chetty's lecture, the binned average corresponded to the red dot. Statistic 1 instead corresponded to the yellow dot.

    b. Calculate Statistic 1: Absolute Mobility at the 25th Percentile using a linear regression for a *random sample* of 50 observations. Set the "seed" using your Harvard ID number as in Lab 1 to make the analysis replicable.

    c. Now calculate the mean of $Rank_{child}$ for children with $Rank_{parent}$ between 20 and 30 (if possible: there may not be any observations in this range) for the small randomly selected sample of 50 observations.

    d. Which set of estimates — statistic 1 estimated using a linear regression in question 7b or the average you calculated in question 7c — are closer to the full population estimate of 41.3 reported by Chetty et al. (2014)?

8. The files to submit for this lab are:

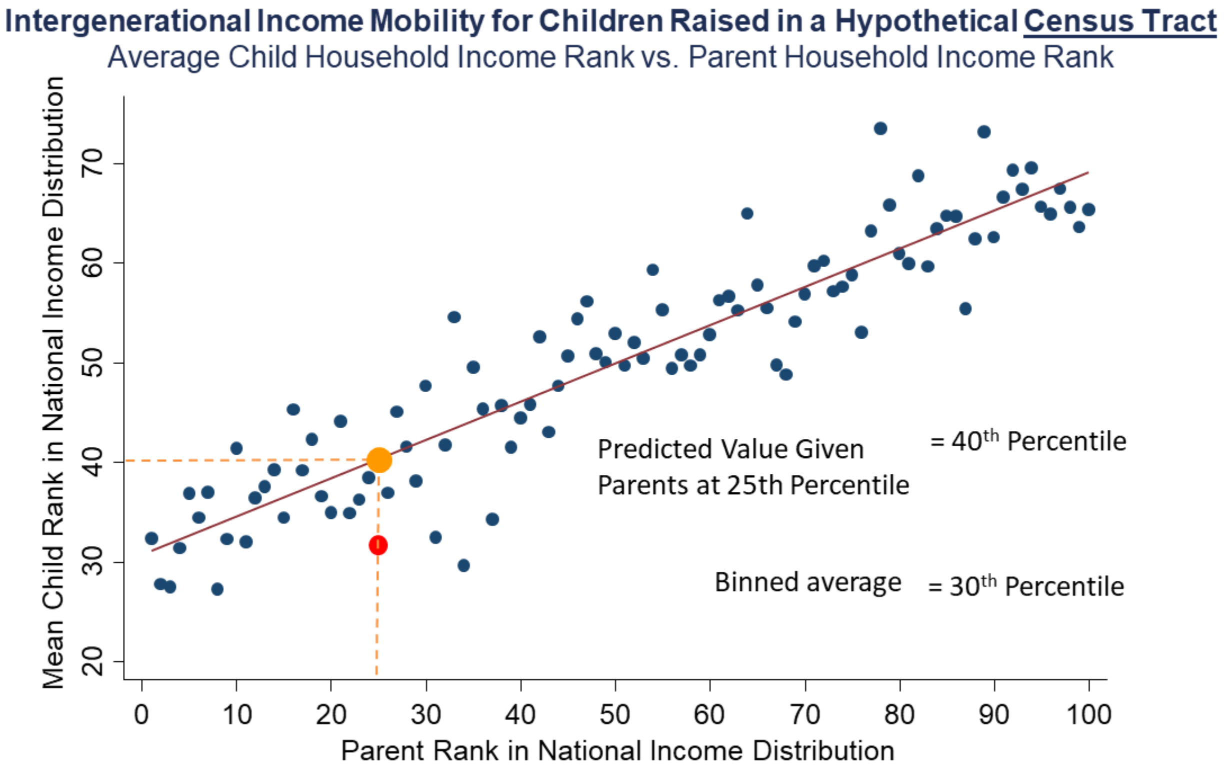
   a. Your well annotated do-file/.R file replicating all your analyses above (with enough comments that a principal investigator on a research project would be able to follow and understand what each step of the code is doing). You can submit this to Gradescope.

   b. For Stata users, a log-file with the log showing the output generated by your final do-file. You can submit this file to the same gradescope assignment as the do-file.

   c. A PDF version of the solutions to the above questions. For graphs, you can save them as .png files and insert them into the document. You can submit this file to the same Gradescope assignment as the do-file and log-file. (Please do not submit a word document: we can only read PDFs in Gradescope).

   d. Your answers to questions 7a-d and 5a to this Google Form so that we can compare the results across students in the class.

FIGURE 1



DATA DESCRIPTION, FILE: nlsy97.dta

The data consist of $N = 5{,}486$ children from the National Longitudinal Survey of Youth 1997, born between 1980 and 1984. The sample and income definitions are chosen to match Chetty et al. (2014) as closely as possible. I measure the income of the children in 2013 and 2015, when they have grown up and are in their early 30s. I measure their parents' income in 1997 and 1998 in the first two waves of the survey.

TABLE 1

Variable Definitions

	Variable	Description	Obs.	Mean	St. Dev.	Min	Max
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	*id_num*	Individual identifier	5,486	n/a	n/a	3	9022
2	*kid_income*	Child's income, averaged across 2013 and 2015	5,486	$70,500	$59,552	0	$329,331
3	*incarcerated*	Child was incarcerated at least once by 2015	5,486	0.0995	0.299	0	1
4	*child_education*	Child's years of education in 2017	5,486	13.77	3.002	5	20
5	*child_college*	Child has college degree in 2017	5,486	0.295	0.456	0	1
6	*child_sat*	Child's SAT score (math plus verbal)	2,456	1,187	198.9	600	1,600
7	*parent_inc*	Parents' income, averaged across 1997 and 1998	5,486	$46,413	$46,089	0	$425,586
8	*mother_education*	Mother years of education	5,486	12.67	2.490	5	20
9	*father_education*	Father years of education	5,486	12.70	2.362	5	20
10	*female*	Child is female	5,486	0.501	0.500	0	1
11	*black*	Child is Black	5,486	0.265	0.441	0	1
12	*hispanic*	Child is Hispanic	5,486	0.199	0.399	0	1
13	*white*	Child is White	5,486	0.600	0.490	0	1
14	*region*	Census Region of residence in 1997, defined as: 1 = Northeast (CT, ME, MA, NH, NJ, NY, PA, RI, VT) 2 = North Central (IL, IN, IA, KS, MI, MN, MO, NE, OH, ND, SD, WI) 3 = South (AL, AR, DE, DC, FL, GA, KY, LA, MD, MS, NC, OK, SC, TN, TX, VA, WV) 4 = West (AK, AZ, CA, CO, HI, ID, MT, NV, NM, OR, UT, WA, WY)	5,486	2.655	0.987	1	4
15	*age2015*	Child's age in 2015	5,486	32.96	1.399	31	35
16	*cohort*	Child's year of birth	5,486	1982	1.399	1980	1984

Note: Child's SAT score is missing (indicated by a period "." in Stata) for 55% of observations in these data.

TABLE 2
Stata Commands

STATA command	Description
*clear the workspace clear all version 17 cap log close *change working directory and open data cd "C:\Users\gbruich\Ec 50\Lab 2\" use nlsy97.dta, clear *Display all variables in the data describe *Report detailed information on all variables codebook	This code shows how to clear the workspace, change the working directory, and open a Stata data file. To change directories on either a mac or windows PC, you can use the drop down menu in Stata. Go to file -> change working directory -> navigate to the folder where your data is located. The command to change directories will appear; it can then be copied and pasted into your .do file. The describe and codebook commands will report information on what is included in the data set loaded into memory.
*Summary stats for one variable sum yvar *Summary stats for observations with wvar<=55 sum yvar if wvar <= 55 *Observations with wvar<=55 and dvar equal to 1 sum yvar if wvar <= 55 & dvar == 1 *Observations with wvar<=55 or dvar equal to 1 sum yvar if wvar <= 55 | dvar == 1 *Observations with wvar between 45 and 55 sum yvar if wvar <= 55 & wvar >= 45 sum yvar if inrange(wvar, 45,55)	We used these commands in Lab 1. These commands report means and standard deviations for *yvar*. The first line calculates these statistics across the full sample. The other lines illustrate how to calculate these statistics for observations meeting certain criteria: when another variable in the data is less than or equal to 55; when one variable is less than or equal to 55 and a separate variable is equal to 1; when either one variable is less than or equal to 55, or a separate variable is equal to 1, or both. The last block of code shows how to calculate the mean of yvar for observations with wvar between 45 and 55. One way is to use the greater than and less than operators along with the & symbol. The second way uses the inrange() function in stata.
*Create new indicator variable gen dvar= 0 replace dvar = 1 if wvar > 1.4767*xvar	These commands show how to generate a new indicator variable called dvar. In the this example, dvar equals 1 if another variable wvar is greater than 1.4767 times a variable xvar.
*Create variable in percentile ranks *Start by rank ordering the data based on yvar egen yvar_rank = rank(yvar) *Get maximum rank, automatically stored as r(max) sum yvar_rank *Store maximum rank as a scalar variable scalar max_rank = r(max) *Normalize rank so that maximum is 100 replace yvar_rank = 100* yvar_rank / max_rank	These commands show how to convert a variable yvar into percentile ranks, normalized so that the highest rank is 100. We start using egen and the rank() function to generate a new variable that rank orders yvar. Then to normalize the variable, we divide it by the maximum rank and multiply by 100. The maximum rank is saved temporarily as r(max) after the sum command. I store it as a "scalar variable" called max_rank, and use that variable in the denominator in the last line.
twoway (scatter yvar xvar) (lfit yvar xvar) graph export figure1.png, replace	This pair of commands first draws a scatter plot of *yvar* against *xvar*. The second line saves the graph as a .png file.

*install bin scatter command ssc install binscatter *Bin scatter plot – connected dots binscatter yvar xvar, linetype(connect) graph export figure2_connected.png, replace *Bin scatter plot – linear best fit line binscatter yvar xvar, linetype(lfit) graph export figure2_linear.png, replace	We used these commands in Lab 1. These commands show how to create binned scatter plots. The first line installs the command from the SSC. The second block of code shows how to create a binned scatter plot where a variable yvar is along the y-axis and a variable xvar is along the x-axis. It will connect the dots with a line. The third block of code shows how to create a binned scatter plot where a variable yvar is along the y-axis and a variable xvar is along the x-axis. It will also plot a linear best fit line. The commands beginning with “graph export” save the graphs as .png files.
*Estimate linear regression regress yvar xvar *Linear regression for observations with wvar<=55 regress yvar xvar if wvar <= 55 * Linear regression for wvar<=55 and dvar equal to 1 regress yvar xvar if wvar <= 55 & dvar == 1 * Linear regression for wvar<=55 or dvar equal to 1 regress yvar xvar if wvar <= 55 | dvar == 1	These commands report estimated regression coefficients from a regression of *yvar* on *xvar*. The first line estimates the regression using the full sample. The other lines illustrate how to restrict the regression to observations meeting certain criteria: when another variable in the data is less than or equal to 55; when one variable is less than or equal to 55 and a separate variable is equal to 1; when either one variable is less than or equal to 55, or a separate variable is equal to 1, or both.
*Set seed to make reproducible set seed 505050505 *Randomly select 50 observations to keep sample 50, count	These commands show how to randomly select 50 observations to keep, dropping the rest. We start by setting the “seed”. Then the sample command with the “, count” option will keep the specified number of observations (here 50), dropping the rest of the observations from memory.
*start a log file log using lab2.log, replace *commands go here *close and save log file log close	These commands show how to start and close a log file, which will save a text file of all the commands and output that appears on the command window in stata.

TABLE 3
R Commands

R command	Description

#clear the workspace
rm(list=ls()) # removes all objects from the environment
cat('\014') # clears the console
 	
#Install and load haven package
if (!require(haven)) install.packages("haven"); library(haven)
 	
#Change working directory and load stata data set
setwd("C:/Users/gbruich/Ec 50/Lab 2")
nlsy <- read_dta("nlsy97.dta")
	This sequence of commands shows how to open Stata datasets in R. The first block of code clears the work space. The second block of code installs and loads the “haven” package. The third block of code changes the working directory to the location of the data and loads in nlsy97.dta. To change the working directory in R Studio, you can also use the drop down menu. Go to session -> set working directory -> choose working directory.
	
	The easiest way to open a Stata data set in R Studio is to use the drop down menu. Go to file, then import data set, and finally browse to locate the file you want to open. This option will be available after you install the haven package.

#Summary stats for one variable
mean(nlsy$yvar, na.rm=TRUE)
 	
#Summary stats for observations with wvar<=55
#Subset data
new_df <- subset(nlsy, wvar <= 55)
 	
#Report mean
mean(new_df$yvar, na.rm=TRUE)
 	
#Alternatively, do it all at once using the with() function
with(subset(nlsy, wvar <= 55), mean(yvar, na.rm=TRUE))
 	
#Observations with wvar<=55 and dvar equal to 1
with(subset(nlsy, wvar <= 55 & dvar == 1), mean(yvar, na.rm=TRUE))
 	
#Observations with wvar<=55 or dvar equal to 1
with(subset(nlsy, wvar <= 55 | dvar == 1), mean(yvar, na.rm=TRUE))
 	
#Observations with wvar between 45 and 55
with(subset(nlsy, wvar <= 55 & wvar >= 45), mean(yvar, na.rm=TRUE))
 	
#Alternatively, use between() function from tidyverse
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)
with(subset(nlsy, between(wvar, 45,55)), mean(yvar, na.rm=TRUE))
	We used these commands in Lab 1. These commands report mean of *yvar*. The first line calculates these statistics across the full sample.
	
	The other lines illustrate how to calculate these statistics for observations meeting certain criteria: when another variable in the data is less than or equal to 55; when one variable is less than or equal to 55 and a separate variable is equal to 1; when either one variable is less than or equal to 55, or a separate variable is equal to 1, or both.
	
	The subset() function will pick out only the observations in a data frame that meet certain criteria. One way to proceed is to create a new data frame and then apply the mean() function to yvar in this new data frame. The second way to proceed is to do it all at once using the with() function. The with() function in R takes two arguments: a data frame and an expression. The data frame argument is nlsy and the expression applies the mean() function to the variable yvar: mean(yvar).
	
	The last block of code shows how to calculate the mean of yvar for observations with wvar between 45 and 55. One way is to use the greater than and less than operators along with the & symbol. The second way uses the between() function from the tidyverse library in R.

#Create new indicator variable nlsy$dvar <- ifelse(nlsy$wvar > 1.4767*nlsy$xvar, 1, 0)	We used similar commands in Lab 1 to generate a new indicator variable. In the this example, dvar equals 1 if another variable wvar is greater than 1.4767 times a variable xvar. dvar equals 0 otherwise

```
# Install and load ggplot2 package
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)
if (!require(ggplot2)) install.packages("ggplot2"); library(ggplot2)

# Draw scatter plot with linear fit line
ggplot(nlsy) + geom_point(aes(x = xvar1, y = yvar)) +
  geom_smooth(aes(x = xvar, y = yvar), method = "lm", se = F)

#Save graph as figure1a.png
ggsave("figure1a.png")
```

These commands show how to draw a scatter plot of *yvar* against *xvar1*. The *geom_smooth* part of the code adds an OLS regression line. The last line saves the graph as a .png file.

```
#install ggplot and statar packages
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)
if (!require(ggplot2)) install.packages("ggplot2"); library(ggplot2)
if (!require(statar)) install.packages("statar"); library(statar)

#Bin scatter plot – connected dots
ggplot(nlsy, aes(x = xvar  , y = yvar)) +
  stat_binmean(n = 20, geom = "line") +
  stat_binmean(n = 20, geom = "point")

#Save graph
ggsave("binscatter_connected.png")

#Bin scatter plot – linear best fit line
ggplot(nlsy, aes(x = xvar , y = yvar)) +
  stat_smooth(method = "lm", se = FALSE) +
  stat_binmean(n = 20, geom = "point")

#Save graph
ggsave("binscatter_bestfitline.png")
```

We used these commands in Lab 1 to create binned scatter plots. The first lines install packages, including the statar package so that we can use the stat_binmean() function with ggplot.

The second block of code shows how to create a binned scatter plot where a variable yvar is along the y-axis and a variable xvar is along the x-axis. It will connect the dots with a line.

The third block of code shows how to create a binned scatter plot where a variable yvar is along the y-axis and a variable xvar is along the x-axis. It will also plot a linear best fit line.

```
#Estimate linear regression
mod1 <- lm(yvar ~ xvar, data=nlsy)
mod1

#Linear regression for observations with wvar<=55
new_df <- subset(nlsy, wvar <= 55)
mod2 <- lm(yvar ~ xvar, data= new_df)
mod2

#Linear regression for wvar<=55 and dvar equal to 1
new_df <- subset(nlsy, wvar <= 55 & dvar == 15)
mod3 <- lm(yvar ~ xvar, data= new_df)
mod3

#Linear regression for wvar<=55 or dvar equal to 1
new_df <- subset(nlsy, wvar <= 55 | dvar == 1)
mod4 <- lm(yvar ~ xvar, data= new_df)
mod4
```

These commands report estimated regression coefficients from a regression of *yvar* on *xvar*. The first line estimates the regression using the full sample.

The other lines illustrate how to restrict the regression to observations meeting certain criteria: when another variable in the data is less than or equal to 55; when one variable is less than or equal to 55 and a separate variable is equal to 1; when either one variable is less than or equal to 55, or a separate variable is equal to 1, or both.

```
#Create variable in percentile ranks
#Start by rank ordering the data based on yvar
nlsy$yvar_rank <- rank(nlsy$yvar)

#Store the maximum rank
max_rank <- max(nlsy$yvar_rank)

#Normalize rank so that maximum is 100
nlsy$yvar_rank <- 100*nlsy$yvar_rank / max_rank
```

We used these commands in Lab 1 to convert a variable yvar into percentile ranks, normalized so that the highest rank is 100. We start using the rank() function to generate a new variable that rank orders yvar. Then to normalize the variable, we divide it by the maximum rank and multiply by 100. The code uses the max() function in R in the denominator to do the normalization.

# Create Function that will Calculate Percentile Ranks with NAs 	
 	
#Define function for percentile ranking
percentile_rank<-function(variable){
 	
#Convert to ranks, taking care of potential missing values
 r <- ifelse(is.na(variable), NA, rank(variable, ties.method = "average"))
 	
#Return percentile rank = rank normalized so max is 100
100*r/max(r, na.rm = T)
}
 	
#Example using Function to Define ranks
nlsy$yvar_rank <-with(nlsy, percentile_rank(yvar))
	Unfortunately, the rank() function does not work as desired for data with missing values (NAs). But we can create our own function to do what we want that will work as intended in more complex data sets. This second block of code shows how to define a new function called percentile_rank() that will generate percentile ranks that assign missing values to NAs, and returns the percentile rank normalized to have a maximum rank of 100.
	
	The last line shows how to use the function to create the variable yvar_rank. The with() function in R takes two arguments: a data frame and an expression. The data frame argument is nlsy and the expression applies the new function we wrote to the variable yvar: percentile_rank(yvar).

#Set seed so that simulations are replicable
HUID <- 505050505
set.seed(HUID)
 	
#install tidyverse package
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)
 	
#Create data frame with random sample of size 50
sample50 <- sample_n(nlsy, 50)
	These commands show how to randomly select 50 observations to keep in a new data frame called sample50. We start by setting the "seed".
	
	Then the sample_n() command will keep the specified number of observations (here 50) from a data frame.

sink(file="lab2_log.txt", split=TRUE)
sink()
	The first line starts a log file. The last line closes and saves the log file.