lab2_2024_v1

February 8, 2024

1 Lab 2: Measuring Upward Mobility Using the National Longitudinal Survey

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

Name: Shreya Chaturvedi

Email: shreyachaturvedi@hks.harvard.edu

HUID: 31575036

Lab: Thursday 3pm @ Belfer L1

Date: February 8, 2024

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 R commands needed for this lab are contained in Table 2. You should have these commands next to you as you work through the lab.

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 how social mobility has changed over time. Finally, you will explore why mobility statistics based on a statistical model are especially useful for characterizing the geography of upward mobility across areas.

1.2 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.

    1. 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.

2. 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%?

[1]:
```
#clear the workspace
rm(list=ls()) # removes all objects from the environment

#Install and load haven package
if (!require(haven)) install.packages("haven"); library(haven)
library(statar, lib="~/Rpackages")
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)
if (!require(ggplot2)) install.packages("ggplot2"); library(ggplot2)
if (!require(statar)) install.packages("statar"); library(statar)
```

```
Loading required package: haven

Loading required package: tidyverse

  Attaching core tidyverse packages                tidyverse
2.0.0
  dplyr     1.1.3       readr     2.1.4
  forcats   1.0.0       stringr   1.5.0
  ggplot2   3.4.4       tibble    3.2.1
  lubridate 1.9.3       tidyr     1.3.0
  purrr     1.0.2
  Conflicts
tidyverse_conflicts()
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()
  Use the conflicted package
(<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

[2]:
```
#Load stata data set
download.file("https://raw.githubusercontent.com/ekassos/ec50_s24/main/nlsy97.
  ↪dta", "nlsy97.dta", mode = "wb")
nlsy1 <- read_dta("nlsy97.dta")

# QUESTION 1 Code
fraction_college <- mean(nlsy1$child_college[nlsy1$mother_education <= 12], na.
  ↪rm = TRUE)
print(fraction_college)
```

```
[1] 0.1818182
```

Question 1 Answer

(Answer here; include your images if needed.)

In the nlsy97 data, 18.18% of children went on to receive a college degree or higher when their mothers had a high school education or less. This estimate I think is reasonably close to the 20.9% result from the Chetty et al (2018) paper.

2. Now we will start by generating a few new variables, following what we did in Lab 1:
    1. Generate percentile ranks for *kid_income*, normalized so that highest rank is 100
    2. Generate percentile ranks for *parent_income*, normalized in the same way

[3]:
```
# QUESTION 2 Code
nlsy1$kid_rank <- rank(nlsy1$kid_income)
nlsy1 <- nlsy1[order(nlsy1$kid_income),]
nlsy1$kid_inc_rank <- ((nlsy1$kid_rank) / max(nlsy1$kid_rank)) * 100

nlsy1$parent_rank <- rank(nlsy1$parent_income)
nlsy1 <- nlsy1[order(nlsy1$parent_income),]
nlsy1$parent_inc_rank <- ((nlsy1$parent_rank) / max(nlsy1$parent_rank)) * 100
```

Question 2 Answer

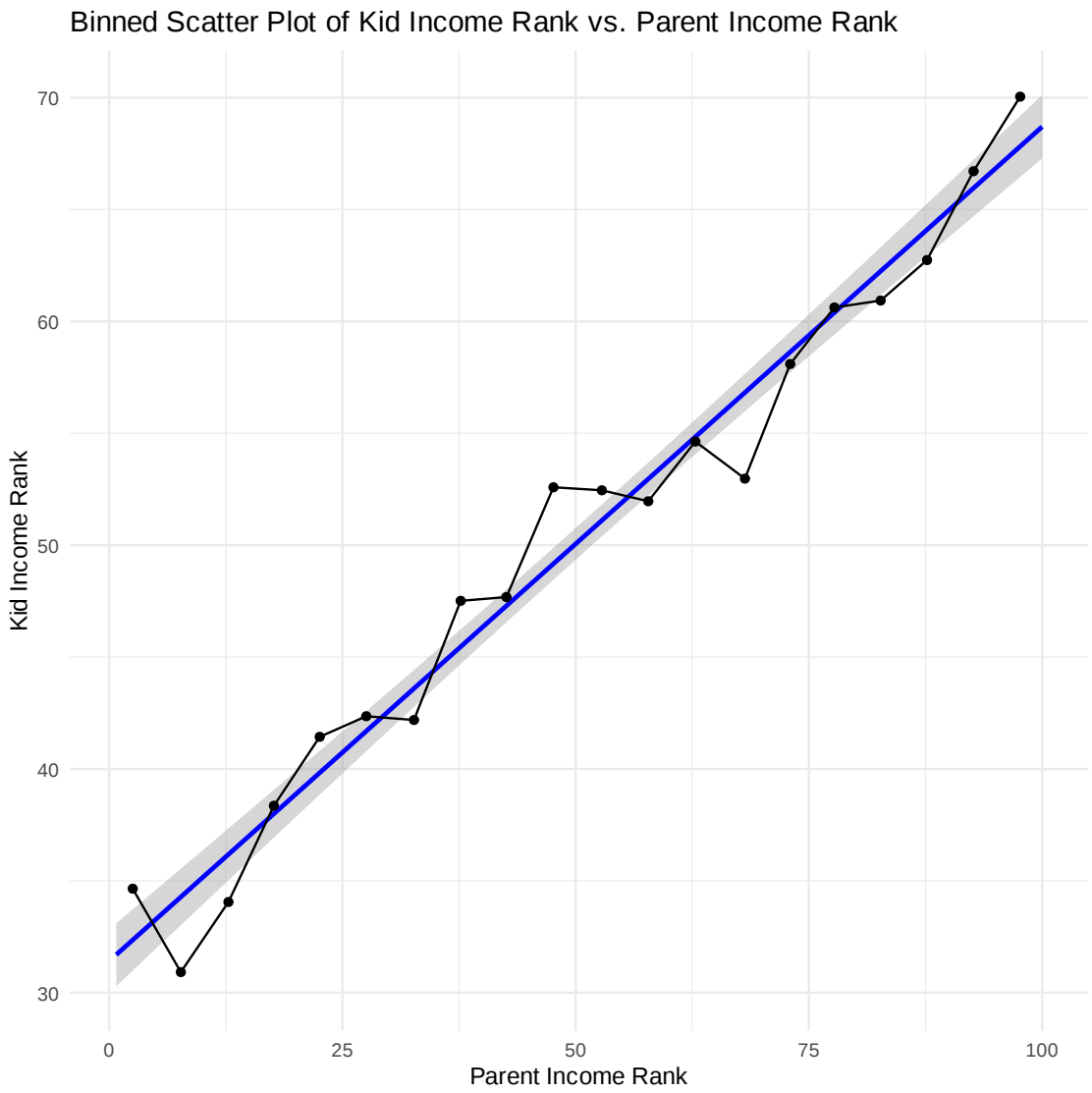
(Answer here; include your images if needed.)

Code given above to generate percentile ranks of parents and kids incomes.

3. Visualize the relationship between child (y-axis) and parent (x-axis) income ranks using a binned scatter plot, including your graph in your solutions. In your judgement, are child and parent income ranks *linearly* related or *non-linearly* related? Explain clearly what you see in your graphical analysis that leads you to your conclusions.

[4]:
```
ggplot(data = nlsy1, aes(x = parent_inc_rank, y = kid_inc_rank)) +
geom_smooth(method = "lm", color = "blue") +  # Adds a linear regression line
stat_binmean(n = 20, geom = "line") +
stat_binmean(n = 20, geom = "point") +
    labs(title = "Binned Scatter Plot of Kid Income Rank vs. Parent Income␣
 ↪Rank",
       x = "Parent Income Rank",
       y = "Kid Income Rank") +
 theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Question 3 Answer

(Answer here; include your images if needed.)

Based on this graph, I would conclude that the relationship between parent income and kid income is approximately linear. This is because, with the exception of a few outlier points, the relationship between kid income rank and parent income rank closely follows the linear regression line drawn throughout the range.

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]:
```r
# QUESTION 4 Code
model1 <- lm(kid_inc_rank ~ parent_inc_rank, data=nlsy1)
print(model1)
coefficients1 <- coef(model1)
intercept1 <- coefficients1["(Intercept)"]
beta1 <- coefficients1["parent_inc_rank"]
```

```
Call:
lm(formula = kid_inc_rank ~ parent_inc_rank, data = nlsy1)

Coefficients:
    (Intercept)  parent_inc_rank
        31.4183           0.3728
```

Question 4 Answer

(Answer here; include your images if needed.)

The intercept is 31.43 and the slope is 0.3748 for the linear regression between kids income rank and parents income rank. In other words, all else the same, a one percent increase in parents income rank is associated with a 0.37 percent increase in kids income rank.

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.
    1. **Statistic 1:** Predicted child income rank from the rank-rank regression in question 4 evaluated at $\text{Rank}_{\text{parent}} = 25$, which Chetty et al. (2014) report as 41.3.
    2. **Statistic 2:** Relative Mobility, which Chetty et al. (2014) report as 34.1.
    3. **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%.
    4. **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.

[6]:
```r
# QUESTION 5 Code

# Part A
predicted_kid_inc_rank = intercept1 + 25*beta1
print(paste("Statistic 1: Predicted Child Income Rank: ",␣
 ↪predicted_kid_inc_rank))

#Part B
relative_mobility <- beta1 * 100
print(paste("Statistic 2: Relative Mobility: ", relative_mobility))

#Part C
```

```
bottom_quintile_parent <- nlsy1$parent_inc_rank <= 20
top_quintile_kid <- nlsy1$kid_inc_rank > 80
probability_bottom_to_top <- mean(top_quintile_kid[bottom_quintile_parent], na.
 ↪rm = TRUE) * 100
print(paste("Statistic 3: Bottom to Top Probability:  ",␣
 ↪probability_bottom_to_top))

# Part D: Calculate the fraction of children outearning their parents
fraction_outearning <- mean(nlsy1$kid_income > nlsy1$parent_income, na.rm =␣
 ↪TRUE) * 100
print(paste("Statistic 4: Fraction Outearning:  ", fraction_outearning))
```

```
[1] "Statistic 1: Predicted Child Income Rank:  40.7380307621951"
[1] "Statistic 2: Relative Mobility:  37.2790721046002"
[1] "Statistic 3: Bottom to Top Probability:  7.37033666969973"
[1] "Statistic 4: Fraction Outearning:  50.8931826467371"
```

Question 5 Answer

(Answer here; include your images if needed.)

Calculations above.

6. Repeat your calculations in Questions 2, 4, and 5a,b,c,d using the Lab 1 **nls6679.dta** data from the 1966 and 1979 National Longitudinal Survey. Is mobility higher for children born in the 1980s or for children born in 1948-1964? Does it depend on what statistic you use? Explain.

[7]:
```
# QUESTION 6 Code
nlsy2 <- read_dta("nls6679.dta")

nlsy2$kid_rank <- rank(nlsy2$kid_income)
nlsy2 <- nlsy2[order(nlsy2$kid_income),]
nlsy2$kid_inc_rank <- ((nlsy2$kid_rank) / max(nlsy2$kid_rank)) * 100

nlsy2$parent_rank <- rank(nlsy2$parent_income)
nlsy2 <- nlsy2[order(nlsy2$parent_income),]
nlsy2$parent_inc_rank <- ((nlsy2$parent_rank) / max(nlsy2$parent_rank)) * 100
```

[8]:
```
model2 <- lm(kid_inc_rank ~ parent_inc_rank, data=nlsy2)
print(model2)
coefficients2 <- coef(model2)
intercept2 <- coefficients2["(Intercept)"]
beta2 <- coefficients2["parent_inc_rank"]
```

```
Call:
lm(formula = kid_inc_rank ~ parent_inc_rank, data = nlsy2)
```

```
Coefficients:
    (Intercept)  parent_inc_rank
        32.6917           0.3462
```

[9]:
```r
# Part A
predicted_kid_inc_rank2 = intercept2 + 25*beta2
print(paste("Statistic 1: Predicted Child Income Rank: ",␣
 ↪predicted_kid_inc_rank2))

#Part B
relative_mobility2 <- beta2 * 100
print(paste("Statistic 2: Relative Mobility: ", relative_mobility2))

#Part C
bottom_quintile_parent <- nlsy2$parent_inc_rank <= 20
top_quintile_kid <- nlsy2$kid_inc_rank > 80
probability_bottom_to_top2 <- mean(top_quintile_kid[bottom_quintile_parent], na.
 ↪rm = TRUE) * 100
print(paste("Statistic 3: Bottom to Top Probability:  ",␣
 ↪probability_bottom_to_top2))

# Part D: Calculate the fraction of children outearning their parents
fraction_outearning2 <- mean(nlsy2$kid_income > nlsy2$parent_income, na.rm =␣
 ↪TRUE) * 100
print(paste("Statistic 4: Fraction Outearning:  ", fraction_outearning2))
```

```
[1] "Statistic 1: Predicted Child Income Rank:  41.3478298452581"
[1] "Statistic 2: Relative Mobility:  34.6245888673475"
[1] "Statistic 3: Bottom to Top Probability:  7.78145695364238"
[1] "Statistic 4: Fraction Outearning:  50.3972194637537"
```

[10]:
```r
stats_df <- data.frame(
  Variable = c("Predicted Kid Inc Rank", "Relative Mobility", "Probability␣
 ↪Bottom to Top", "Fraction Outearning"),
  nlsy1 = c(predicted_kid_inc_rank, relative_mobility,␣
 ↪probability_bottom_to_top, fraction_outearning),
  nlsy2 = c(predicted_kid_inc_rank2, relative_mobility2,␣
 ↪probability_bottom_to_top2, fraction_outearning2)
)

# Print the data frame
print(stats_df)
```

```
                 Variable     nlsy1     nlsy2
1    Predicted Kid Inc Rank 40.738031 41.347830
2         Relative Mobility 37.279072 34.624589
```

```
3 Probability Bottom to Top  7.370337  7.781457
4       Fraction Outearning 50.893183 50.397219
```

Question 6 Answer

(Answer here; include your images if needed.)

Based on these 4 statistics, it is clear that whether upwards mobility increased or decreased between the two periods depends on the statistic we choose to measure it. Specifically,

- For Predicted Kid Income Rank, since (41.34 > 40.73), there is higher mobility in the nlsy2 dataset.
- For Relative Mobility, since (37.28 > 34.62), there is higher mobility in the nlsy1 dataset.
- For Probability Bottom to Top, since (7.78 > 7.37), there is slightly higher mobility in the nlsy2 dataset.
- For Fraction Outearning, since (50.89 > 50.40), there is slightly higher mobility in the nlsy1 dataset.

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.

    1. Instead of using a regression, calculate the simple arithmetic mean of $\text{Rank}_{\text{child}}$ for children with $\text{Rank}_{\text{parent}}$ 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.

    2. 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.

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

    4. 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)?

[11]:
```
# QUESTION 7 Code

# PART A
mean_kid_inc_rank <- mean(nlsy1$kid_inc_rank[nlsy1$parent_inc_rank >= 20 &␣
 ↪nlsy1$parent_inc_rank <= 30], na.rm = TRUE)
print(paste("Mean Child Income Rank: ", mean_kid_inc_rank))

#PART B
```

```r
set.seed(31575036)
sampled_nlsy1 <- nlsy1[sample(nrow(nlsy1), 50), ]
model3 <- lm(kid_inc_rank ~ parent_inc_rank, data = sampled_nlsy1)
coefficients3 <- coef(model3)
intercept3 <- coefficients3["(Intercept)"]
beta3 <- coefficients3["parent_inc_rank"]
predicted_kid_inc_rank3 = intercept3 + 25*beta3
print(paste("Predicted Child Income Rank: ", predicted_kid_inc_rank3))

#PART C
#print(sum(sampled_nlsy1$parent_inc_rank >= 20 & sampled_nlsy1$parent_inc_rank
 ↪<= 30, na.rm = TRUE))
average_kid_inc_rank <-
 ↪mean(sampled_nlsy1$kid_inc_rank[sampled_nlsy1$parent_inc_rank >= 20 &
 ↪sampled_nlsy1$parent_inc_rank <= 30], na.rm = TRUE)
print(paste("Average Child Rank for Parent Rank between 20 and 30: ",
 ↪average_kid_inc_rank))
```

```
[1] "Mean Child Income Rank:  41.9323368324966"
[1] "Predicted Child Income Rank:  39.442244867716"
[1] "Average Child Rank for Parent Rank between 20 and 30:  39.0780465540849"
```

Question 7 Answer

(Answer here; include your images if needed.)

A. The simple arithmetic mean of children's income rank for parents with income ranks between 20 and 30 in the NLS 1997 data is 41.93. Compared to the Statistic 1 calculated for a parent at the 25th percentile, which was 40.738, this indicates that children from parents within the 20th to 30th percentile range have, on average, higher mobility ($41.93 > 40.73$). This comparison highlights that the simple arithmetic mean can provide a different insight into mobility, in this case, suggesting a slightly higher upward mobility for this group.

B. Absolute Mobility is 39.44 (code above).

C. There are onlyy two observations in my subsample, with the average child rank = 39.08

D. Both the estimates (39.44, 39.08) are roughly equal with the absolute mobility very slightly higher and therefore closer to the population estimate of 41.3.

8. Create an annotated/commented do-file, .ipynb Python Notebook, or .R file that can replicate all your analyses above. This will be the final code that you submit on Gradescope. The motivation for using do-files and .R files is described on the next page, which has been adapted from training materials used by Innovations for Poverty Action (IPA) and the Abdul Latif Jameel Poverty Action Lab (J-PAL).

1.3 How to submit your assignment

Step 1 Access the lab assignment under the "Assignments" tab on Canvas

Step 2 Access Gradescope from Canvas
Step 3 Access the lab assignment on Gradescope
Step 4 Upload your files *Check What files to submit to confirm what files you need to submit.*
Step 5 What you'll see after submitting your lab assignment
Step 6 Check your submitted files
Step 7 You'll receive an email confirmation as well

1.4 What files to submit

If you're using Python Notebook to write your R code, and a document editor to write your answers
If you're using a Python Notebook to write your R code AND to write your answers

1.5 WHAT ARE DO-FILES AND .R FILES AND WHY DO WE NEED ONE?

Let's imagine the following situation - you just found out you have to present your results to a partner– all the averages you produced and comparisons you made. Suppose you also found out that the data you had used to produce all these results was not completely clean, and have only just fixed it. You now have incorrect numbers and need to re-do everything.

How would you go about it? Would you reproduce everything you did for Lab 1 from scratch? Can you do it? How long would it take you to do? Just re-typing all those commands into Stata or R in order and checking them would take an hour.

An important feature of any good research project is that the results should be reproducible. For Stata and R the easiest way to do this is to create a text file that lists all your commands in order, so anyone can re-run all your Stata or R work on a project anytime. Such text files that are produced within Stata or linked to Stata are called do-files, because they have an extension .do (like intro_exercise.do). Similarly, in R, these files are called .R files because they have an extension of .R. These files feed commands directly into Stata or R without you having to type or copy them into the command window.

An added bonus is that having do-files and .R files makes it very easy to fix your typos, re-order commands, and create more complicated chains of commands that wouldn't work otherwise. You can now quickly reproduce your work, correct it, adjust it, and build on it.

Finally, do-files and .R files make it possible for multiple people to work on a project, which is necessary for collaborating with others or when you hand off a project to someone else.

2 Figure 1

2.1 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 and have adjusted it for inflation.

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 ($2015), averaged across 2013 and 2015 and adjusted for inflation	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_income*	Parents' income ($2015), averaged across 1997-1998 and adjusted for inflation	5,486	$68,537	$68,060	$0	$628,462
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	*male*	Child is male	5,486	0.499	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.

2.2 TABLE 2: R Commands

R command

Description

```
#clear the workspace
rm(list=ls()) # removes all objects from the environment

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

#Load stata data set
download.file("https://raw.githubusercontent.com/ekassos/ec50_s24/main/nlsy97.dta", "nlsy97.dta
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 downloads and loads in nls6679.dta.

```
#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.

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

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

```
#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.

```
#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


# 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))
```

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.

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.