SOC-GA 2332 Intro to Stats Lab 13

Risa Gelles-Watnick

11/26/2025

Part 0: Logistics

- **Problem Set 4** due tonight, Dec. 5th, 11:59pm
- **Replication Part II** due Dec. 15th, 5:00pm
- Any final course feedback is very welcome!

Part 1: Matching

- For the following parts on causal inference, we will use the Early Childhood Longitudinal Study dataset.
- We will examine the effect of going to a Catholic school (`catholic = 1`), as opposed to a public school (`catholic = 0`), on students' standardized math score (`c5r2mtsc_std`). The pre-treatment covariates are:
  - `race_white`: Is the student white (1) or not (0)?
  - `p5hmage`: Mother's age
  - `w3income`: Family income
  - `p5numpla`: Number of places the student has lived for at least 4 months
  - `w3momed_hsb`: Is the mother's education level high-school or below (1) or some college or more (0)?

```r
## import data
ecls <- read.csv("data/ecls.csv")

## covariates variable name vector
ecls_cov <- c('race_white', 'p5hmage', 'w3income', 'p5numpla', 'w3momed_hsb')
```

Check if sample is balanced

- To check if the sample is balanced or not, we first examine the difference in means by treatment status for outcome variable and covariates.

```r
## check difference in mean outcomes by school type
ecls %>%
  group_by(catholic) %>%
  summarise(n_students = n(),
            mean_math = mean(c5r2mtsc_std),
            std_error_math = sd(c5r2mtsc_std) / sqrt(n_students))
```

```
## # A tibble: 2 x 4
##   catholic n_students mean_math std_error_math
##      <int>      <int>     <dbl>          <dbl>
## 1        0       9568   -0.0306         0.0104
## 2        1       1510    0.194          0.0224
```

```
## two Sample t-test: (H0: mean math scores do not differ by school types)
with(ecls, t.test(c5r2mtsc_std ~ catholic))
```

```
##
##  Welch Two Sample t-test
##
## data:  c5r2mtsc_std by catholic
## t = -9.1069, df = 2214.5, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.2727988 -0.1761292
## sample estimates:
## mean in group 0 mean in group 1
##     -0.03059583      0.19386817
```

```
## check difference in means for pre-treatment covariates by school types
## summarise group means for covariates
ecls %>%
  group_by(catholic) %>%
  select(one_of(ecls_cov)) %>%
  summarise_all(funs(mean(., na.rm = T)))
```

```
## # A tibble: 2 x 6
##   catholic race_white p5hmage w3income p5numpla w3momed_hsb
##      <int>      <dbl>   <dbl>    <dbl>    <dbl>       <dbl>
## 1        0      0.556    37.6   54889.     1.13       0.464
## 2        1      0.725    39.6   82074.     1.09       0.227
```

```
## Two sample t-test for every covariate
## lapply: a build-in loop that apply the t-test function along the name vector
lapply(ecls_cov, function(v){
  t.test(ecls[, v] ~ ecls[, 'catholic'])
})
```

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  ecls[, v] by ecls[, "catholic"]
## t = -13.453, df = 2143.3, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.1936817 -0.1444003
## sample estimates:
## mean in group 0 mean in group 1
##       0.5561246       0.7251656
##
##
## [[2]]
##
##  Welch Two Sample t-test
```

```
##
## data:  ecls[, v] by ecls[, "catholic"]
## t = -12.665, df = 2186.9, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -2.326071 -1.702317
## sample estimates:
## mean in group 0 mean in group 1
##        37.56097        39.57516
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  ecls[, v] by ecls[, "catholic"]
## t = -20.25, df = 1825.1, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -29818.10 -24552.18
## sample estimates:
## mean in group 0 mean in group 1
##        54889.16        82074.30
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  ecls[, v] by ecls[, "catholic"]
## t = 4.2458, df = 2233.7, p-value = 0.00002267
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.02150833 0.05842896
## sample estimates:
## mean in group 0 mean in group 1
##        1.132669        1.092701
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  ecls[, v] by ecls[, "catholic"]
## t = 18.855, df = 2107.3, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.2122471 0.2615226
## sample estimates:
## mean in group 0 mean in group 1
##       0.4640918       0.2272069
```

- As we can see, the difference in mean for math score and for covariates are statistically significant.

Choose and execute a matching algorithm

- Remember, propensity scores are a one-number summary of all the different covariates' values for each individual. They are the probability of being treated conditional on pre-treatment covariates.
- *Question:* When do we use propensity score matching?
- *Question:* Can propensity scores fix violations of the ignorability/no ommited variables assumption?
- Assumptions of propensity score matching
  - Ignorability of treatment assignment (no unobserved confounders)
  - Balance (correctly specified propensity score formula)
  - Overlap
  - SUTVA
- The first step in propensity score matching should *always* be deciding which variables you want to match on, and which ones are most important to achieve balance in.
  - Prioritize balancing variables that are most predictive of the outcome (most impactful confounders)
  - Generally, you want to include all covariates in your propensity score formula unless you have a very large number of them
  - *Question:* What variables seem most important to match on in this data?
- To create a balanced sample from the original, unbalanced dataset, we need to choose and execute a matching algorithm in order to created a balanced dataset to estimate ATE. The package `MatchIt` estimates the propensity score in the background and then matches observations based on the method of your choice.
- In this example we use ***nearest neighbor*** matching, which matches units based on some measure of distance. The default and most common measure is the propensity score difference, which is the difference between the propensity scores of each treated and control unit.

```r
## MatchIt does not allow missing values, so we need to remove observations with NAs
ecls_nomiss <- ecls %>%
  select(c5r2mtsc_std, catholic, all_of(ecls_cov)) %>%
  na.omit()

## nearest neighbor matching (see documentation for different matching methods)
mod_match <- matchit(catholic ~ race_white + w3income + p5hmage + p5numpla + w3momed_hsb, # start with
                     method = "nearest", # what method should the package use to match propensity score
                     estimand = "ATT", # what estimand is the package looking for
                     data = ecls_nomiss,
                     replace = FALSE) # matching without replacement
```

- Notice that in this example, we matched without replacement.
  - *Question:* What does it mean to match with replacement vs. matching without?
  - *Question:* What might be the pro/cons of matching with replacement vs. without?

Create matched dataset

- Using the `matchit` function, we obtained a `matchit` object (`mod_match`) that can be used to create a dataframe that contains only the matched observations.
  - Note that in our case, this final dataset is smaller than the original: it contains 2,704 observations, which contains 1,352 original treated units, and the other 1,352 control units that match the treated units one on one.
  - The estimated effect is therefore ATT.
  - The final dataset contains a variable called `distance`, which is the propensity score.

    - Matching ideally requires a common support in propensity.

```r
## to create a dataframe containing only the matched observations
dta_m <- match.data(mod_match)
```

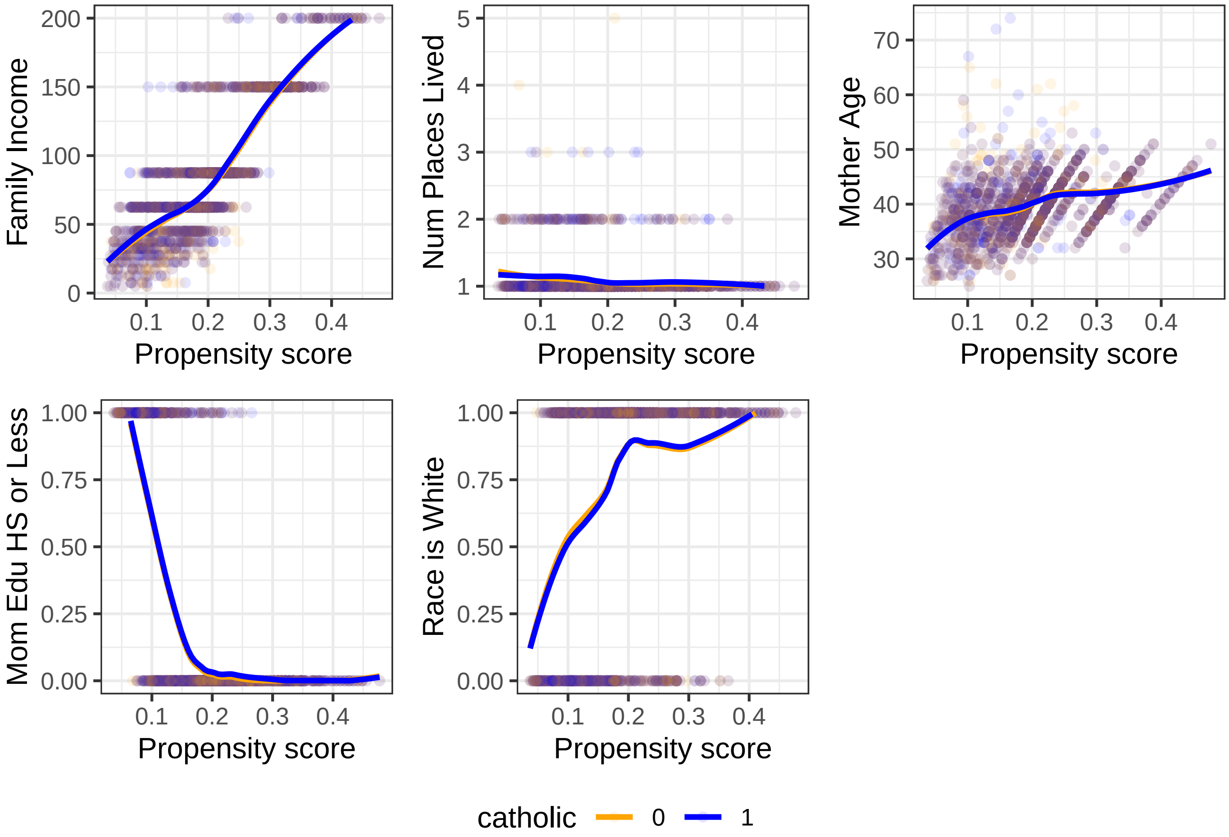
Examine covariates after matching

- After matching, it is useful to plot the mean of each covariate against the estimated propensity score, color-coded by treatment status. If matching is done well, the treatment and control groups will have (near) identical means of each covariate at each value of the propensity score.

```r
## a plotting function that plots the distribution of propensity score of a given covariate
fn_bal <- function(dta, variable, yname) {
  dta$variable <- dta[, variable]
  if (variable == 'w3income') {
    dta$variable <- dta$variable / 10^3
    }
  dta$catholic <- as.factor(dta$catholic)
  support <- c(min(dta$variable), max(dta$variable))
  plot <- ggplot(dta, aes(x = distance, y = variable, color = catholic)) +
    geom_point(alpha = 0.1, size = 1.3) +
    geom_smooth(method = "loess", se = F) +
    scale_color_manual(values=c("orange", "blue")) +
    xlab("Propensity score") +
    ylab(yname) +
    theme_bw() +
    ylim(support)

  return(plot)
}

## plot and arrange the plots
ggarrange(
  fn_bal(dta_m, "w3income", "Family Income"),
  fn_bal(dta_m, "p5numpla", "Num Places Lived"),
  fn_bal(dta_m, "p5hmage", "Mother Age"),
  fn_bal(dta_m, "w3momed_hsb", "Mom Edu HS or Less"),
  fn_bal(dta_m, "race_white", "Race is White"),
  common.legend = T,
  legend = "bottom")
```



```
## you can also check difference-in-means in matched data
dta_m %>%
  group_by(catholic) %>%
  select(one_of(ecls_cov)) %>%
  summarise_all(funs(mean))
```

```
## # A tibble: 2 x 6
##   catholic race_white p5hmage w3income p5numpla w3momed_hsb
##      <int>      <dbl>   <dbl>    <dbl>    <dbl>       <dbl>
## 1        0      0.747    39.6   81404.     1.08       0.215
## 2        1      0.741    39.6   82569.     1.09       0.223
```

```
## two sample t-test
lapply(ecls_cov, function(v) {
  t.test(dta_m[, v] ~ dta_m$catholic)
})
```

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = 0.35243, df = 2701.8, p-value = 0.7245
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.02700440  0.03883872
## sample estimates:
## mean in group 0 mean in group 1
##       0.7470414       0.7411243
```

```
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -0.21331, df = 2702, p-value = 0.8311
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.4372485  0.3514496
## sample estimates:
## mean in group 0 mean in group 1
##         39.5503         39.5932
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -0.64787, df = 2701.9, p-value = 0.5171
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -4690.731  2360.845
## sample estimates:
## mean in group 0 mean in group 1
##        81403.99        82568.94
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -1.339, df = 2699.5, p-value = 0.1807
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.038278301  0.007213213
## sample estimates:
## mean in group 0 mean in group 1
##        1.076183        1.091716
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -0.51108, df = 2701.5, p-value = 0.6093
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.03935185  0.02307966
## sample estimates:
```

```
## mean in group 0 mean in group 1
##       0.2152367       0.2233728
```

- We can also explore how well our matching worked with a table displaying standard mean differences and variance ratios for treated and control groups along key variables.

```
# creating table showing balance statistics

# pulling variable names
vars <- summary(mod_match)$sum.matched %>% rownames()

# transforming summary into tibble and adding back rownames
matched_tibble <- summary(mod_match)$sum.matched %>%
  as_tibble() %>%
  mutate(Variable = vars) %>%
  select("Variable",
         "Means Treated",
         "Means Control",
         "Std. Mean Diff.",
         "Var. Ratio")

# creating measure of ratio of standard deviations
sum_tibble <- matched_tibble %>%
  rename(var_ratio = "Var. Ratio") %>%
  mutate(sdr = sqrt(var_ratio)) %>%
  rename("Var. Ratio" = "var_ratio",
         "Std. Dev. Ratio" = "sdr")

# creating pretty table
sum_tibble %>%
  kbl(caption = "Balance statistics for propensity score model",
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: Balance statistics for propensity score model

Variable	Means Treated	Means Control	Std. Mean Diff.	Var. Ratio	Std. Dev. Ratio
distance	0.1927431	0.1927395	0.0000426	1.000038	1.000019
race_white	0.7411243	0.7470414	-0.0135090	NA	NA
w3income	82568.9356509	81403.9926036	0.0248481	1.011370	1.005669
p5hmage	39.5931953	39.5502959	0.0081970	1.003571	1.001784
p5numpla	1.0917160	1.0761834	0.0507349	1.062727	1.030887
w3momed_hsb	0.2233728	0.2152367	0.0195342	NA	NA

- It's a subjective decision what threshold to use to decide when you've gotten an appropriately balanced data set. You can look through the literature to find what threshold others in the field use, but it also depends how important you think it is to get specific variables in your model right.
- In this case our matched dataset looks pretty good! Normally you would probably have to iterate many times between checking balance statistics and tweaking your propensity score formula until you get a satisfactory balance.
  - Remember it's ok (indeed, necessary) to tweak your propensity scores based on how well they balance your other covariates (this is not p-hacking). What you *don't* want to do is tweak your

    propensity scores based on your outcome variable.
  - You can do anything that makes your propensity score formula achieve better balance (interact variables, exponentiate variables, log variables, remove variables completely from the formula, etc.)
    - Just because you remove a variable from your propensity score formula doesn't mean you don't need to achieve balance in it.
  - Tweaking the propensity score formula is mostly guesswork and iteration; it's hard to know what changes will effect. For example, if you don't have a good balance in a specific variable, changing how that specific variable is formulated in your propensity score formula won't necessarily affect the balance of that variable.
- You can also use machine learning packages like `SuperLearner` or machine learning-based arguments in the `MatchIt` and `WeightIt` packages to specify your propensity score models for you. *Question:* Why might this be a bad idea?
  - I would always try these models as well and see if they give you a better balance than when you manually try to get the propensity score formula right. Moral of the story, try everything and pick the model that achieves the best balance!

Estimate Treatment Effects

- Using matched dataset, we can now estimate ATT using two different methods.
  - We can either use the results of a two sample t-test and calculate the difference in means or regress mean math score on school types
    - Note that since we are using the matched dataset, the distribution of the covariates, in our case, are balanced in both the treatment and control group
    - Therefore, whether control variables are included or not in the linear model, in our case, would not affect the ATT estimate
- *Question:* What might be the advantage of using a linear regression with covariates included rather than a t-test? It will allow you to further adjust for imbalance in any covariates where you couldn't achieve your ideal balance. And it also reduces your standard error. If you matched with replacement, you also need to use a regression because you need to weight your observations based on how many times they're included in the data.

```r
## 1. two sample t-test
with(dta_m, t.test(c5r2mtsc_std ~ catholic))
```

```
##
##  Welch Two Sample t-test
##
## data:  c5r2mtsc_std by catholic
## t = 4.4523, df = 2682.3, p-value = 0.000008843
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.08686896 0.22360634
## sample estimates:
## mean in group 0 mean in group 1
##       0.3649055       0.2096679
```

```r
## mean in group 0: 0.3673451
## mean in group 1: 0.2096679
## ATT = 0.2096679 - 0.3673451 = -0.1576772

## 2. OLS model
## no covariates
lm_treat1 <- lm(c5r2mtsc_std ~ catholic, data = dta_m)
```

```
## with covariates
lm_treat2 <- lm(c5r2mtsc_std ~ catholic + race_white + p5hmage +
                  I(w3income / 10^3) + p5numpla + w3momed_hsb, data = dta_m)
## display models
stargazer(lm_treat1, lm_treat2, type="text",
          star.char = c("+", "*", "**", "***"),
          star.cutoffs = c(0.1, 0.05, 0.01, 0.001))
```

```
##
## =====================================================================
##                                    Dependent variable:
##                    -------------------------------------------------
##                                       c5r2mtsc_std
##                               (1)                      (2)
## ---------------------------------------------------------------------
## catholic                   -0.155***                -0.154***
##                             (0.035)                  (0.033)
##
## race_white                                           0.345***
##                                                      (0.039)
##
## p5hmage                                              0.012***
##                                                      (0.003)
##
## I(w3income/103)                                      0.003***
##                                                      (0.0004)
##
## p5numpla                                              -0.047
##                                                      (0.055)
##
## w3momed_hsb                                         -0.287***
##                                                      (0.041)
##
## Constant                    0.365***                -0.502***
##                             (0.025)                  (0.148)
##
## ---------------------------------------------------------------------
## Observations                 2,704                    2,704
## R2                           0.007                    0.108
## Adjusted R2                  0.007                    0.106
## Residual Std. Error    0.907 (df = 2702)        0.860 (df = 2697)
## F Statistic         19.823*** (df = 1; 2702) 54.510*** (df = 6; 2697)
## =====================================================================
## Note:                                     *p<0.1; **p<0.05; ***p<0.01
```

- *Question:* How would we interpret this coefficient?

Part 4: Instrumental Variables

- We'll look at an example from Causal Inference: The Mixtape by Scott Cunningham
- In this example we use data from NLS Young Men Cohort of the National Longitudinal Survey to consider the returns to schooling in terms of income. Card (1995) wants to estimate:

$$Y_i = \alpha + \beta_1 S_i + \beta_2 X_i + \epsilon_i$$

- where Y is log earnings, S is years of schooling, X is a matrix of observed covariates and *epsilon* is an error term containing unobserved endogenous covariates, for example, ability. Ability, we might expect, is correlated with income as well as years of schooling. Therefore schooling is biased.
- Card (1995) proposes an instrumental variables strategy and instruments schooling by a college-in-the-county dummy variable. The assumption is that if there is a nearby 4-year college, it will increase the likelihood of going to college.
  - The key assumptions here is that having a college in the county is distributed quasi-randomly, and that the only way having a college in the county increases earnings is through its effect on likelihood of going to college.

Estimating causal effects using IV designs

- One of the most common and intuitive estimators is two-stage least squares, with the instrument denoted as Z_i
- We estimate the first-stage (Effect of college-in-county on years of schooling):

$$S_i = \gamma + \rho Z_i + \eta X_i + u_i$$

* And plug the fitted values into the second-stage regression (Effect of years of schooling on earnings):

$$Y_i = \alpha^{iv} + \beta_1^{iv} \hat{S}_i + \beta_2^{iv} X_i + v_i$$

* This can be done manually in R by regressing with the predicted values, or using the `ivreg` function.

```r
## function to read the data from github
read_data <- function(df)
{
  full_path <- paste("https://raw.github.com/scunning1975/mixtape/master/",
                     df, sep = "")
  df <- read_dta(full_path)
  return(df)
}

## read data
card <- read_data("card.dta")

## define variable
## Endo = endogenous variable, Exo = exogenous variable, Inst = Instrument
attach(card)
Endo_educ <- educ
Exo_ <- cbind(exper, black, south, married, smsa)
Inst <- nearc4

## OLS
ols_reg <- lm(lwage ~ Endo_educ + Exo_)

## 2SLS
iv_reg <- ivreg(lwage ~ Endo_educ + Exo_ | Exo_ + Inst)

## how coef estimates
stargazer(ols_reg, iv_reg, type="text",
          star.char = c("+", "*", "**", "***"),
          star.cutoffs = c(0.1, 0.05, 0.01, 0.001))
```

```
##
## ======================================================================
##                                          Dependent variable:
##                                  --------------------------------------
##                                                 lwage
##                                            OLS            instrumental
##                                                             variable
##                                            (1)                (2)
## ----------------------------------------------------------------------
## Endo_educ                               0.071***             0.124*
##                                          (0.003)            (0.050)
##
## Exo_exper                               0.034***            0.056**
##                                          (0.002)            (0.020)
##
## Exo_black                               -0.166***           -0.116*
##                                          (0.018)            (0.051)
##
## Exo_south                               -0.132***          -0.113***
##                                          (0.015)            (0.023)
##
## Exo_married                             -0.036***          -0.032***
##                                          (0.003)            (0.005)
##
## Exo_smsa                                0.176***            0.148***
##                                          (0.015)            (0.031)
##
## Constant                                5.063***            4.162***
##                                          (0.064)            (0.850)
##
## ----------------------------------------------------------------------
## Observations                              3,003              3,003
## R2                                        0.305              0.251
## Adjusted R2                               0.304              0.250
## Residual Std. Error (df = 2996)           0.370              0.384
## F Statistic                     219.153*** (df = 6; 2996)
## ======================================================================
## Note:                                      *p<0.1; **p<0.05; ***p<0.01
```

- Never-taker $S_i = 0, \forall Z_i$
- Always-taker $S_i = 1, \forall Z_i$
- Complier v.s. Defier
- Recall that IV estimates are **LATE**
  - *Question:* Who is the inferential group in this design? Who can you make causal claims about?
- Assumptions of instrumental variable designs:
  - Ignorability of instrument (instrument is unconfounded)
  - Exclusion (instrument only affects outcome through treatment)
  - Monotonicity (no defiers)

- Non-zero correlation between instrument and treatment (this is testable!)
- SUTVA