

# Final Exam Prep Sessions 1 - 5

MSBA 6440: Causal Inference via Experimentation

*Danny Moncada (monca016)*

*April 27, 2020*

## Session 1-3: Randomized Experiments (lm)

Correlation  $=/ \neq$  Causation

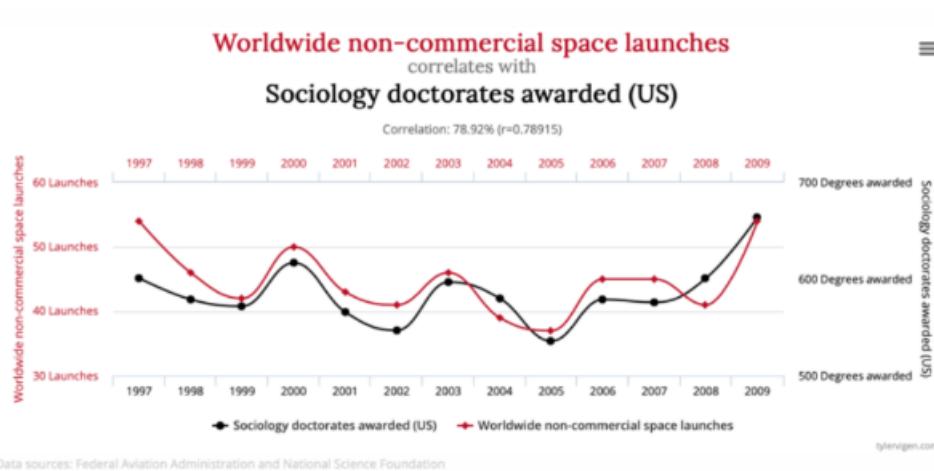


Figure 1: Necessary but not sufficient

## Requirements for Causality

- **Correlation**
  - X must be associated with Y.
- **Temporal Precendence**
  - Variation in X must precede the variation in Y
- **No Endogeneity**
  - No unobserved confounds, no measurement error, no simultaneity, no sample selection.

## Taxonomy of “Threats to Causal Inference”

$$y_i = \alpha + \beta x_i + \gamma z_i + \mu_i$$

- **Omitted Variables**
  - unobserved confounders for X.

- **Simultaneity**
  - X and Y cause each other.
- **Selection**
  - Whether we observe a representative sample.
- **Mis-measurement**
  - Our measure of X is imperfect.



Figure 2: You are interested in the impact of X on Y, but there are other variables that you are not including in your model; e.g. coffee drinking is linked with smoking.

## The Take-Away

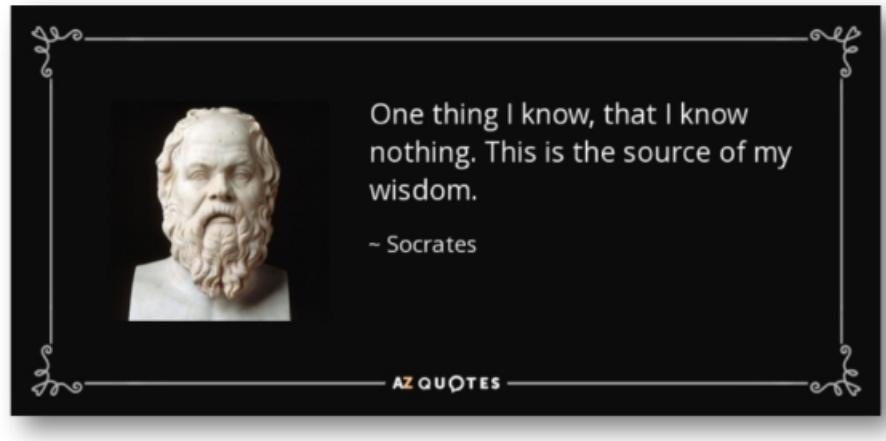


Figure 3: There's no way to control away all other variables, and thus what to control.

## Example: Selection Bias

- **Abraham Wald**

- Hungarian statistician working for the Allies in WW2.
- RAF Asked for Input on How to Armor B-52 Bombers.
- *So, we just place armor where the bullet holes are concentrated, right?*

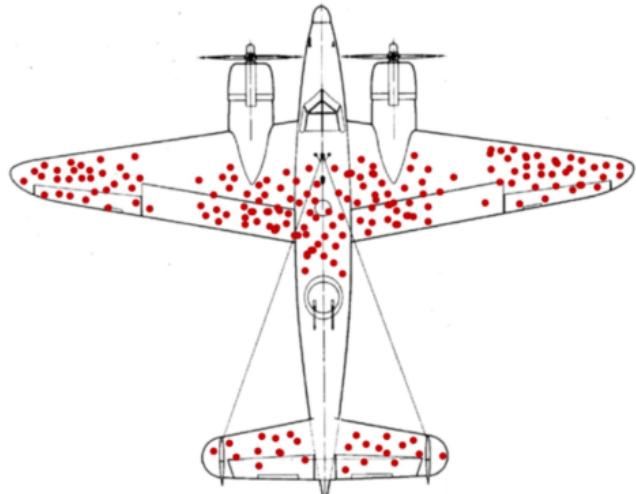


Figure 4: Is our sample representative of the population we're looking to model?

## Example: Simultaneity Bias

- The “Matthew Effect”
  - Status Success Status.
  - A.K.A. “Rich-get-Richer”
  - Or...

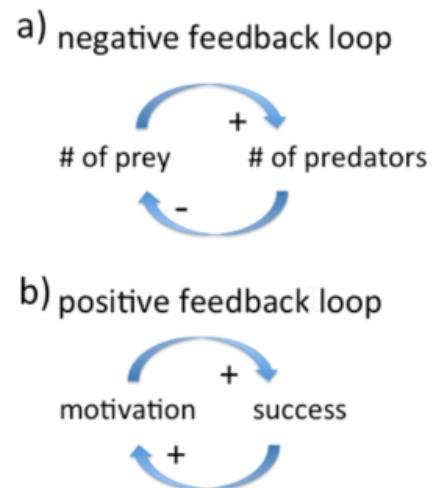


Figure 5: Y may affect X.

## Intuition: Measurement Error Bias

- If the measure of x is ‘noisy’, the estimate of x’s effect on y underestimate the actual effect (if measurement error is random).



Figure 6: We’re interested on the impact of X on Y, but the measures of X are no correct.

# More Generally

- These ‘threats to causal inference’ are formally referred to as sources of endogeneity.



- Formally:  $x$  is correlated with the regression error term.
- Informally:  $x$  is associated with relevant, yet unmodeled factors in the regression.

Figure 7: Something that is in the error term that is not being modeled.

```
# Author: Gordon Burtch and Gautam Ray
# Course: MSBA 6440
# Session: Causality and Endogeneity
# Topic: Simulating Endogeneity

set.seed(100)

## Some examples of what happens when we ignore different kinds of endogeneity

# 1) Measurement Error
# We build our variable X, and then also an erroneously measured version of X.
X <- rnorm(200, mean = 50, sd=7)
X_m <- X + rnorm(200,mean=4, sd=15)

# Now we simulate Y using the true data generating process (accurately measured X)
Y <- 0.5*X + rnorm(200,mean=0,sd=1)

# You can see that the estimate is hugely deflated when we ignore the measurement error.
summary(lm(Y~X))

##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5000 -0.3750  0.0000  0.3750  1.5000
```

```

## -3.3031 -0.6666  0.0320  0.6496  2.8931
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06088    0.59601  -0.102    0.919
## X            0.50114    0.01181  42.423 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.065 on 198 degrees of freedom
## Multiple R-squared:  0.9009, Adjusted R-squared:  0.9004
## F-statistic:  1800 on 1 and 198 DF,  p-value: < 2.2e-16

```

```
summary(lm(Y~X_m))
```

```

##
## Call:
## lm(formula = Y ~ X_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.4062 -1.8631  0.1169  1.6514  7.1989
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.40168    0.68256 29.890 < 2e-16 ***
## X_m         0.08631    0.01211  7.125 1.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.018 on 198 degrees of freedom
## Multiple R-squared:  0.2041, Adjusted R-squared:     0.2
## F-statistic: 50.76 on 1 and 198 DF,  p-value: 1.895e-11

```

# 2) Omitted Variables // Correlated Unobservable

```

# Let's add in a confounder for X that we will "not observe" in our regression and see
# what it does.
Z <- rnorm(200, mean=3, sd=.5) - X
Y <- 0.5*X + 2*Z + rnorm(200,mean=0,sd=1)

# You can see that ignoring Z causes X to be downward biased (because Z is negatively
# correlated with X)
summary(lm(Y~X))

```

```

##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.0211 -1.0238  0.0938  1.0383  3.6412
## 
```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.6731    0.8376  7.967 1.25e-13 ***
## X           -1.5111    0.0166 -91.026 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.497 on 198 degrees of freedom
## Multiple R-squared:  0.9767, Adjusted R-squared:  0.9765
## F-statistic:  8286 on 1 and 198 DF,  p-value: < 2.2e-16

summary(lm(Y~X+Z))

```

```

##
## Call:
## lm(formula = Y ~ X + Z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.57143 -0.78364  0.02501  0.77308  3.02907
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9656    0.7206   1.340 0.181755
## X           0.4844    0.1429   3.389 0.000848 ***
## Z           2.0025    0.1429  14.009 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.062 on 197 degrees of freedom
## Multiple R-squared:  0.9883, Adjusted R-squared:  0.9882
## F-statistic:  8326 on 2 and 197 DF,  p-value: < 2.2e-16

```

## Establishing Causality

# What Should We Do to Establish Causality?

- You Have Two Options:
  - Experiments: Intervene and manipulate  $x$  ourselves, at random so it is uncorrelated with  $e$  by design.
  - Econometrics: Figure out a way to get at ‘good’ variation in  $x$ , which is not correlated with  $e$ .
- It’s always useful to start off by thinking of the ideal experiment.
  - You then run that experiment (or something close).
  - If not, at least you have thought through the problems facing causal inference.
  - Figure out a strategy to **break the correlation between  $x$  and  $e$** .
    - Make  $X$  exogenous/orthogonal to make it uncorrelated with the error term
    - Determine what problems observational data may bring out

## Randomized Experiments

- Randomized Experiments (RCTs) are the Gold Standard for causal inference.
- The researcher intervenes to manipulate  $x$ , at random (e.g., coin toss), such that **it** is nearly guaranteed not to correlate with  $e$ .
  - This gives you temporal precedence (hit  $X$ , then observe changes in  $Y$ ).
- By randomly assigning  $x$ , you are eliminating any possibility of an unobserved confound and any possibility of simultaneity.
  - **This may or may not solve measurement error!**
  - Sample selection may still be a problem (if treatment causes attrition bias, for example).

```
# BMI example
# Code below is used to generate simulated dataset
```

```

set.seed(1000)
gender=sample(0:1,10000,TRUE)
#weight in lb for men and women
weight=rnorm(10000, mean = 180, sd = 40) + gender*rnorm(10000,mean=-30,sd=10)
#height in in for men and women
height=rnorm(10000, mean = 60, sd = 1) + gender*rnorm(10000,mean=-6,sd=0.5)
#using general formula
bmi=(weight*703)/(height*height)+rnorm(10000, mean=0, sd = 3)
Data<-data.frame(gender, weight, height, bmi)
head(Data)

```

```

##   gender   weight   height      bmi
## 1       1 233.7009 53.42704 58.05625
## 2       1 137.1929 55.02970 29.92295
## 3       0 242.2719 60.70448 43.60021
## 4       1 108.3572 53.18934 25.30988
## 5       0 200.4341 59.05139 42.82229
## 6       1 169.1344 55.21242 42.28387

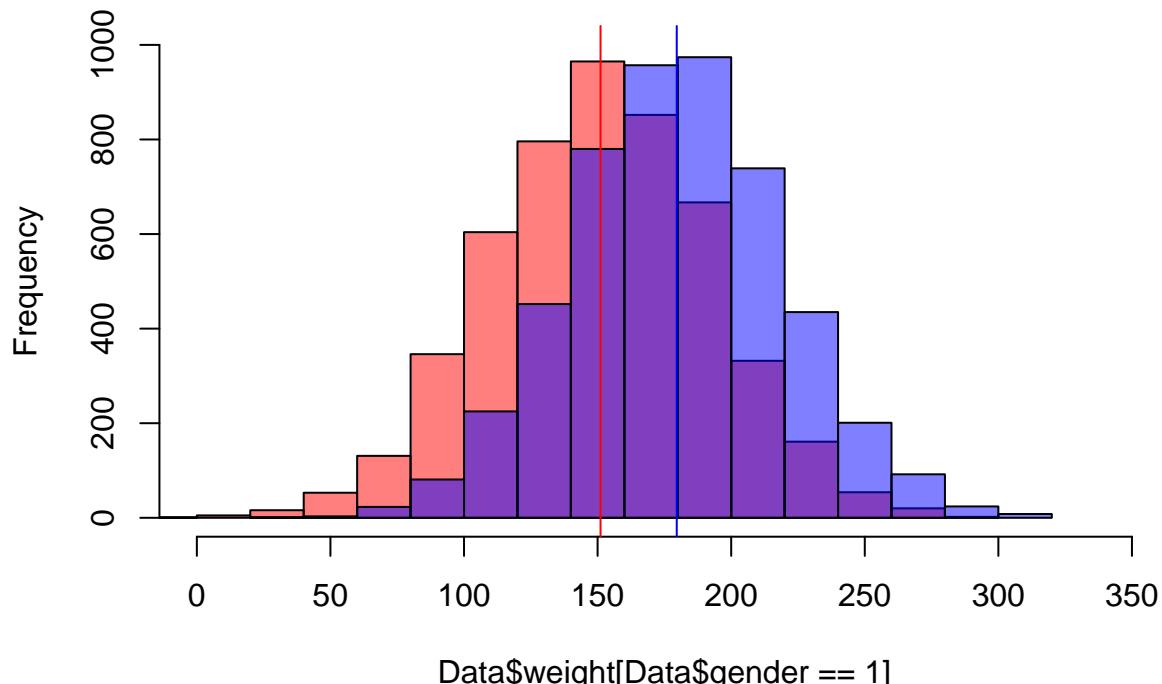
```

```

hist(Data$weight[Data$gender==1],col=rgb(1,0,0,.5),ylim=c(0,1000),xlim=c(0,350))
hist(Data$weight[Data$gender==0],col=rgb(0,0,1,.5),add=T)
abline(v=mean(Data$weight[Data$gender==1]),col="red")
abline(v=mean(Data$weight[Data$gender==0]),col="blue")

```

Histogram of Data\$weight[Data\$gender == 1]



```

# check averages
mean(weight)

## [1] 165.3459

summary(lm(weight~1)) ## run a regression to calculate the mean (run the regression against 1)

##
## Call:
## lm(formula = weight ~ 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -170.462  -28.729    0.436   28.656  148.521
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 165.3459    0.4296  384.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.96 on 9999 degrees of freedom

summary(lm(weight~gender)) #average weight

##
## Call:
## lm(formula = weight ~ gender)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -156.211  -27.464    0.365   27.439  137.001
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 179.6254    0.5734  313.3   <2e-16 ***
## gender     -28.5304    0.8104  -35.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.52 on 9998 degrees of freedom
## Multiple R-squared:  0.1103, Adjusted R-squared:  0.1102 
## F-statistic:  1239 on 1 and 9998 DF,  p-value: < 2.2e-16

## Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 179.6254 0.5734 313.3 <2e-16 ***
## gender     -28.5304 0.8104 -35.2 <2e-16 ***
## Men's average weight is 179 lbs.
## Women's average weight is 28 lbs lighter than men.

summary(lm(height~gender)) #average height

```

```

## 
## Call:
## lm(formula = height ~ gender)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2445 -0.7291 -0.0108  0.7098  3.5688
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 60.00062  0.01515 3959.2 <2e-16 ***
## gender     -5.98404  0.02142 -279.4 <2e-16 ***  
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.071 on 9998 degrees of freedom
## Multiple R-squared:  0.8864, Adjusted R-squared:  0.8864 
## F-statistic: 7.804e+04 on 1 and 9998 DF,  p-value: < 2.2e-16

## Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 60.00062 0.01515 3959.2 <2e-16 ***
## gender     -5.98404 0.02142 -279.4 <2e-16 ***  
## Men's average height is 60 inches
3
```

```

## [1] 3

## Women's average height is ~6 inches shorter.

# BMI as a function of weight and height #
summary(lm(log(bmi)~log(weight)+log(height))) #general formula
```

```

## Warning in log(weight): NaNs produced

## 
## Call:
## lm(formula = log(bmi) ~ log(weight) + log(height))
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67898 -0.05467  0.00309  0.05895  1.50471
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.624136  0.072464  91.41 <2e-16 ***
## log(weight) 1.015946  0.003547 286.45 <2e-16 ***  
## log(height) -2.038372  0.018753 -108.70 <2e-16 *** 
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09997 on 9996 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8921, Adjusted R-squared:  0.8921 
## F-statistic: 4.131e+04 on 2 and 9996 DF,  p-value: < 2.2e-16
```

```

## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.624136 0.072464 91.41 <2e-16 ***
## log(weight) 1.015946 0.003547 286.45 <2e-16 ***
## log(height) -2.038372 0.018753 -108.70 <2e-16 ***

## Intercept: Not meaningful (geometric mean for people with 0 weight and 0 height)
## When we have logs on both sides, we can interpret the coefficient like this:
## for every 1% increase in weight (lbs), we increase BMI by 1%
## for every 10% increase in height (in), we decrease BMI by 18%

summary(lm(log(bmi)~log(weight)*gender+log(height)*gender)) #adjusted for men and women

## Warning in log(weight): NaNs produced

## 
## Call:
## lm(formula = log(bmi) ~ log(weight) * gender + log(height) *
##     gender)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.676662 -0.05460  0.00326  0.05919  1.50986
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.733596  0.344313 19.557 <2e-16 ***
## log(weight) 1.012185  0.005982 169.212 <2e-16 ***
## gender      -0.226349  0.438641 -0.516  0.606    
## log(height) -2.060376  0.083729 -24.608 <2e-16 ***
## log(weight):gender 0.005958  0.007460  0.799  0.424    
## gender:log(height) 0.048578  0.107759  0.451  0.652    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09998 on 9993 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8921, Adjusted R-squared:  0.892  
## F-statistic: 1.652e+04 on 5 and 9993 DF,  p-value: < 2.2e-16

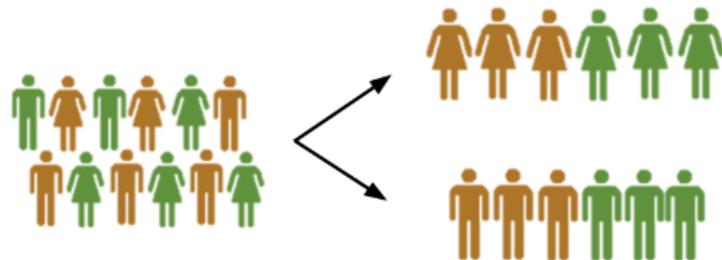
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.733596  0.344313 19.557 <2e-16 ***
## log(weight) 1.012185  0.005982 169.212 <2e-16 ***
## gender      -0.226349  0.438641 -0.516  0.606    
## log(height) -2.060376  0.083729 -24.608 <2e-16 ***
## log(weight):gender 0.005958  0.007460  0.799  0.424    
## gender:log(height) 0.048578  0.107759  0.451  0.652    

## gender: Essentially 0, there is no relationship with gender and BMI
## log(weight): gender: The interaction between log(weight) and gender is insignificant
## gender: log(height): The interaction between log(height) and gender is insignificant

```

## On the Fly vs. Block Randomization

- If we are randomizing in advance, and not on the fly, we typically first “block” on features that would typically be considered confounds...



- Alternatively, we can randomize and check balance on those features. Re-randomize until the treatment is orthogonal.

\* On the fly: randomize people as they come onsite and check at the end if they are balanced on covariates you care about

- Block: do this apriori (before making observations)

### Effect of a Magic Pill

- Imagine we have a magic pill we want to evaluate, to see if it reduces BMI. We randomly assign pill to half of subjects using coin toss for each one.

$$BMI_i = \alpha + \beta_1 MagicPill_i + \mu_i$$

- If  $MagicPill = 1$ , you are called treated, and if it = 0 you are called Control.
  - This is an A/B test.
- Our causal effect:

$$E(BMI|MagicPill = 1) - E(BMI|MagicPill = 0)$$

+ This is a t-test.

```
#### MSBA 6440 ####  
### Gordon Burtch and Gautam Ray###  
### Code for Lecture 2 ###
```

```
## Set working dir - for this example only
```

```
setwd("~/MSBA 2020 All Files/Spring 2020/MSBA 6440 - Causal Inference via Econmtrcs Exprmnt/Week 2 - Des
```

```

# BMI example with randomization

BMI = read.csv("BMI_pill.csv")
#check balance of some observable covariates between treated and control groups.
t.test(data=BMI, height~magicpill)

## 
## Welch Two Sample t-test
##
## data: height by magicpill
## t = -0.30601, df = 376.88, p-value = 0.7598
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.7333199  0.5358093
## sample estimates:
## mean in group 0 mean in group 1
##      56.93160      57.03036

t.test(data=BMI, gender~magicpill)

## 
## Welch Two Sample t-test
##
## data: gender by magicpill
## t = 0.60068, df = 387.92, p-value = 0.5484
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.06863251  0.12901898
## sample estimates:
## mean in group 0 mean in group 1
##      0.5138889      0.4836957

t.test(data=BMI, weight~magicpill)

## 
## Welch Two Sample t-test
##
## data: weight by magicpill
## t = 0.38354, df = 374.55, p-value = 0.7015
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -6.524866  9.687110
## sample estimates:
## mean in group 0 mean in group 1
##      175.5419      173.9607

#Let's see if BMI changes with receipt of the randomly assigned pill.
mp<-lm(log(bmi)~magicpill, data = BMI)
summary(mp)

##

```

```

## Call:
## lm(formula = log(bmi) ~ magicpill, data = BMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.3251 -0.1616  0.0419  0.2416  0.7935
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.55643   0.02514 141.447 < 2e-16 ***
## magicpill   -0.15842   0.03707 -4.273 2.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3695 on 398 degrees of freedom
## Multiple R-squared:  0.04387,    Adjusted R-squared:  0.04147
## F-statistic: 18.26 on 1 and 398 DF,  p-value: 2.413e-05

```

*#Let's see if this changes when we control for some observables (if its randomly assigned  
# then this won't make a difference)*

```

mphw<-lm(log(bmi)~magicpill+log(height) + log(weight), data = BMI)
summary(mphw)

```

```

##
## Call:
## lm(formula = log(bmi) ~ magicpill + log(height) + log(weight),
##      data = BMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.3903 -0.1578  0.0616  0.2290  0.6635
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.94100   1.32834  5.978 5.04e-09 ***
## magicpill   -0.15550   0.03654 -4.255 2.61e-05 ***
## log(height) -1.19644   0.32677 -3.661 0.000285 ***
## log(weight)  0.08745   0.07290  1.200 0.230977
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3641 on 396 degrees of freedom
## Multiple R-squared:  0.07649,    Adjusted R-squared:  0.06949
## F-statistic: 10.93 on 3 and 396 DF,  p-value: 6.509e-07

```

```

mphwg<-lm(log(bmi)~magicpill+log(height) + log(weight) + gender, data = BMI)
summary(mphwg)

```

```

##
## Call:
## lm(formula = log(bmi) ~ magicpill + log(height) + log(weight) +
##      gender, data = BMI)
##
```

```

## Residuals:
##      Min     1Q Median     3Q    Max
## -3.3299 -0.1681  0.0498  0.2275  0.7130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.96344   4.23651   3.768  0.00019 ***
## magicpill   -0.15977   0.03647  -4.381 1.52e-05 ***
## log(height) -3.13220   1.02420  -3.058  0.00238 **
## log(weight)  0.07132   0.07307   0.976  0.32966
## gender      -0.23017   0.11547  -1.993  0.04691 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3627 on 395 degrees of freedom
## Multiple R-squared:  0.08569, Adjusted R-squared:  0.07643
## F-statistic: 9.255 on 4 and 395 DF, p-value: 3.712e-07

stargazer(mp, mphw, mphwg, type="text", column.labels=c("magic pill", "magic pill with controls",
                                                       "magic pill with more controls"))

```

Dependent variable:			
	magic pill	magic pill with controls	magic pill with more controls
	(1)	(2)	(3)
magicpill	-0.158*** (0.037)	-0.155*** (0.037)	-0.160*** (0.036)
log(height)		-1.196*** (0.327)	-3.132*** (1.024)
log(weight)		0.087 (0.073)	0.071 (0.073)
gender			-0.230** (0.115)
Constant	3.556*** (0.025)	7.941*** (1.328)	15.963*** (4.237)
Observations	400	400	400
R2	0.044	0.076	0.086
Adjusted R2	0.041	0.069	0.076
Residual Std. Error	0.370 (df = 398)	0.364 (df = 396)	0.363 (df = 395)
F Statistic	18.260*** (df = 1; 398)	10.933*** (df = 3; 396)	9.255*** (df = 4; 395)
Note:	*p<0.1; **p<0.05; ***p<0.01		

## How Do We Know $\text{Cov}(X,e) = 0$ ?

- That is, how do we know the randomization worked?
- You argue it by construction, typically. As your sample gets larger, the probability of a random correlation between X and e goes to 0 (if your randomization isn't broken).
- You might also provide some suggestive evidence.
  - What difference would you expect to see in the gender proportion between treatment and control groups? What about the difference in average height?

\* There were no systematic difference between the treated and control groups.

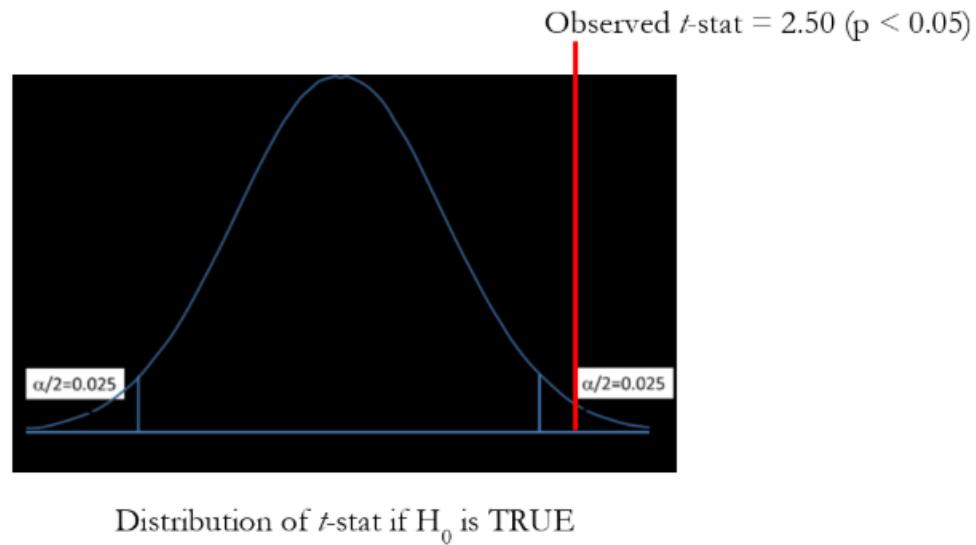
### To Determine Sample Size Requirements

- Significance threshold,  $\alpha$  ( $< 0.05$ )
- Assumed power level,  $\beta$  ( $> 0.80$ )
- Effect size,  $\delta$
- Sample size,  $N$

## What is the Significance Threshold?

- **Type I Error:** rejecting the null when you shouldn't.
- An alpha of 0.05 in a two-tailed test implies that we reject our null only if the observed test statistic lies in the extreme tails of the null distribution, i.e., bottom 2.5% or top 2.5%.
  - For a 1-tailed test, this means the value needs to lie in the top (bottom) 5% of the distribution, at a specific tail.
- When you have a lower threshold for significance, your probability of concluding there's an effect when it's not really there is lower.

# What is the Significance Threshold?



# What is the Power Level?

- **Type II Error:** failing to reject the null when you should.

- Power is  $1 - \Pr(\text{Type II Error})$ . That is, when you have **more power**, your **probability of failing to detect a true effect is smaller**.

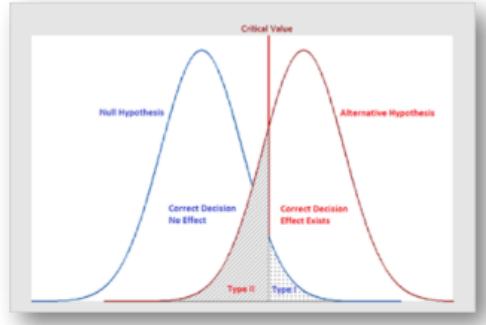


Figure 8: We want probability to be less than 20% for type II error.

# Type I vs Type II Error

		State of the World	
		H0	H1
Decision	H0	Correct Acceptance	Type II Error (b)
	H1	Type 1 Error (a)	Correct Rejection

Figure 9: Top Left, Bottom Right

## Relationship Between Alpha, Beta and N?

- As Sample Size Goes Up...

- Type I Errors Decline.
- Type II Errors Decline.
- You can detect smaller effects.
- How can we calculate required sample size in R?

```
> power_t_test(n=NULL,type=c("two.sample"),alternative="two.sided",power=0.8,sig.level=0.05,delta=0.22)
```

```
Two-sample t test power calculation
```

```
n = 325.297
delta = 0.22
sd = 1
sig.level = 0.05
power = 0.8
alternative = two.sided
```

```
NOTE: n is number in *each* group
```

- Question:  
Is there such a thing as an overpowered study? **NO!**

- Large samples lead to everything being significant... you want to know if the effect is significant.

## Other Concerns: Interference

- Interference: recall that we need to be able to accurately target units with treatment assignment. If there is interference, it means there are treatment spillovers between subjects in the two groups.
- For example, if subjects share pills with others, this would be interference. In other contexts, it might simply amount to sharing information.
- Even being aware that someone else got a treatment and you did not (or vice versa) might change subject behavior. This kills your ability to recover accurate causal estimates.

Figure 10: No interaction between groups

## Other Concerns: Valid Controls

- **Valid Control:** It is not always clear what the appropriate control should be for a given treatment, e.g., no treatment, a placebo (and if so, what placebo)?
- An Example: We want to examine the yield impact of a specific fertilizer, we use a tractor to spread fertilizer where there is no other practicable way to spread fertilizer. What is a valid control?

\* Is showing a charity ad an appropriate control for people we're trying to sell a product to?

# Other Concerns: Response Biases

- Are subjects aware of the experiment?
  - If so, they may change their behavior.
- Social Desirability Bias
  - They may conceal behaviors they think are undesirable, or lie about behavior.
- Demand Effects
  - Doing what they think the experimenter would want them to do.

\* Are subjects thinking about their behavior?

## Formalizing Experiment Design

### Full-Factorial Design:

- If we have two alternative treatments, a full factorial design considers the 2x2 matrix of all permutations.
- This allows you to understand not just the independent effects of these treatments, but how they combine with one another.
- That is, does having one treatment amplify the effect of the other, or attenuate it?

### This is Called an *Interaction Effect*

\* Testing two ads: one group would see no ads, one group would see ad A, one group would see ad B, and group 4 would see A & B, to determine if A has an effect, if B has an effect, or if A&B combined have an effect.

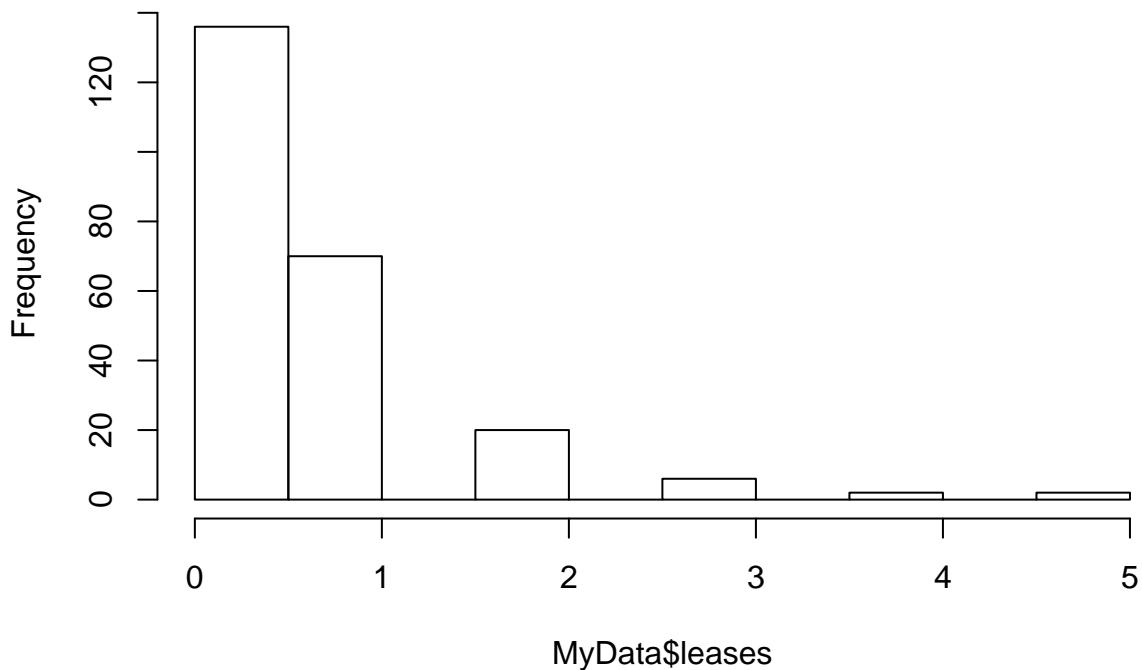
```
setwd("~/MSBA 2020 All Files/Spring 2020/MSBA 6440 - Causal Inference via Econometrics Exprmnt/Week 3 - Analysis")  
  
#### MSBA 6440 ####  
#### Gordon Burtch and Gautam Ray###  
#### Updated Feb 2020 ####  
#### Code for Lecture 3 ####
```

```
# Analyzing Movie Rental Pricing Experiment Data

##### Load Dataset #####
MyData<- read.csv("MovieData-Exp.csv")

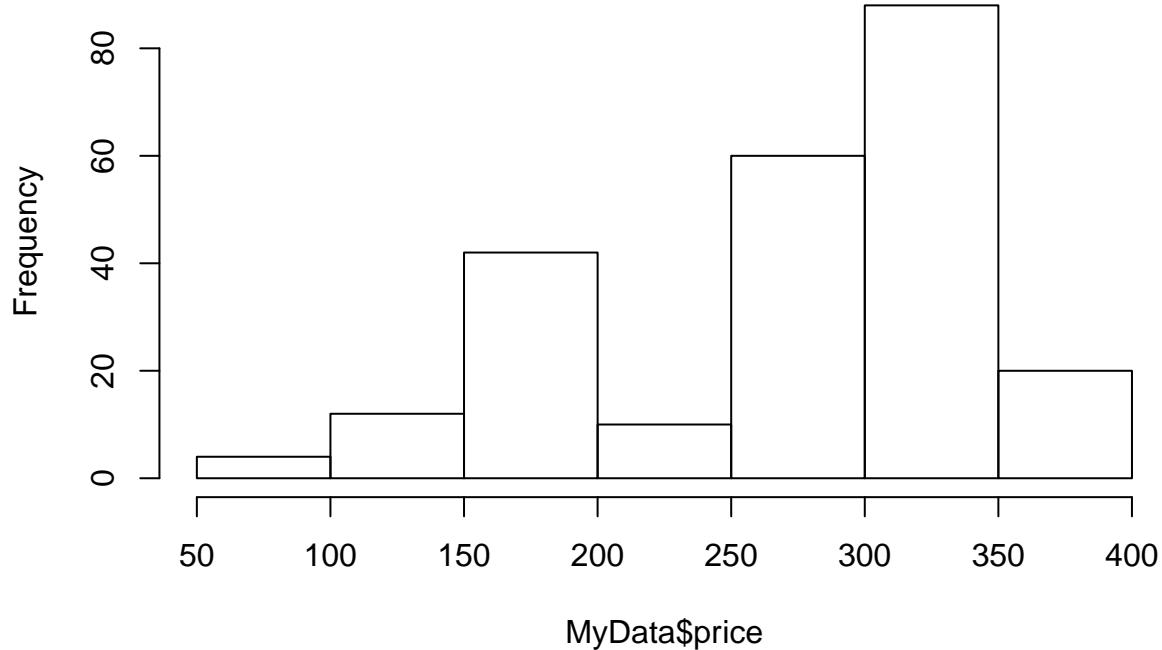
# Descriptive statistics / plots...
hist(MyData$leases)
```

**Histogram of MyData\$leases**



```
hist(MyData$price)
```

## Histogram of MyData\$price



```
# Let's make a treatment dummy to keep things simple for now.  
# This helps us do some easy randomization checks.  
# Let's also construct the discount variable.  
MyData$disc <- MyData$base_price - MyData$price  
summary(MyData[MyData$disc>0,]$disc)  
  
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.  
##    10.00   30.00  50.00    52.88   70.00  120.00  
  
MyData$treated <- (MyData$disc > 0)  
# Let's check randomization...  
t.test(likes~treated,data=MyData)  
  
##  
##  Welch Two Sample t-test  
##  
## data: likes by treated  
## t = 0.060292, df = 233.42, p-value = 0.952  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -465366.0 494747.5  
## sample estimates:  
## mean in group FALSE mean in group TRUE  
##                  2343120              2328429
```

```

# Let's evaluate statistical power now.
# Do we have enough data? Remember, we have 0.5 leases per movie-week on average prior
# to the experiment taking place, for this set of customers.
# Management wants to know about a 20% increase with 90% confidence.
# Thus, we need to detect an increase of 0.10 leases per movie in the week of the experiment.
# That is,  $0.50 * 20\% = 0.10$ . This is our delta parameter.
# 90% confidence implies an alpha of 0.10 ( $1 - 0.9 = 0.1$ ).
# We assume a power of 80% absent other information.

# The first power test tells us what sort of difference we can reliably detect with our current
# sample size... 118 movies per group.
power_t_test(n=118,type=c("two.sample"),alternative="two.sided",power=0.8,sig.level=0.1,delta=NULL)

## 
## Two-sample t test power calculation
##
##          n = 118
##          delta = 0.324651
##          sd = 1
##          sig.level = 0.1
##          power = 0.8
##    alternative = two.sided
##
## NOTE: n is number in *each* group

# The second tells how big a sample we would need to detect the 20% change they hope to find.
power_t_test(n=NULL,type=c("two.sample"),alternative="two.sided",power=0.8,sig.level=0.1,delta=0.1)

## 
## Two-sample t test power calculation
##
##          n = 1237.188
##          delta = 0.1
##          sd = 1
##          sig.level = 0.1
##          power = 0.8
##    alternative = two.sided
##
## NOTE: n is number in *each* group

# Note: we appear to be heavily underpowered to detect the effect management is looking for.
# I would thus caution management about reading too much into results from this experiment.
# I might even advise repeating it with the bigger, requisite sample.

# That said, moving on...
# Let's estimate the treatment effect.

**** OLS of leases on price and log(price) ****#
ols <- lm(leases ~ price, data = MyData)
olslog <- lm(leases ~ log(price), data = MyData)
stargazer(ols,olslog,title="OLS leases on prices and log(price)",type="text",
          column.labels=c("price","log(price)"))

```

```

## 
## OLS leases on prices and log(price)
## =====
##                               Dependent variable:
## 
##                               leases
##          price      log(price)
##          (1)        (2)
## -----
## price                  -0.001
##                         (0.001)
## 
## log(price)             -0.255
##                         (0.177)
## 
## Constant              0.973***   2.043**
##                         (0.229)    (0.994)
## 
## -----
## Observations           236       236
## R2                     0.011     0.009
## Adjusted R2            0.007     0.005
## Residual Std. Error (df = 234) 0.909     0.910
## F Statistic (df = 1; 234)    2.571     2.062
## =====
## Note:                 *p<0.1; **p<0.05; ***p<0.01

**** OLS of leases on price and log(price) with additional controls****
olslogcontrols <- lm(leases ~ log(price) + log(likes), data = MyData)

stargazer(ols,olslog,olslogcontrols,
           title="OLS leases on prices, log(price) and controls",
           type="text",column.labels=c("price","log(price)","with controls"))

## 
## OLS leases on prices, log(price) and controls
## =====
##                               Dependent variable:
## 
##                               leases
##          price      log(price)      with controls
##          (1)        (2)          (3)
## -----
## price                  -0.001
##                         (0.001)
## 
## log(price)             -0.255      -0.266
##                         (0.177)    (0.177)
## 
## log(likes)              0.043
##                         (0.031)
## 
## Constant              0.973***   2.043**   1.506
##                         (0.229)    (0.994)    (1.066)

```

```

## -----
## Observations           236          236          236
## R2                   0.011        0.009        0.017
## Adjusted R2           0.007        0.005        0.008
## Residual Std. Error  0.909 (df = 234)  0.910 (df = 234)  0.909 (df = 233)
## F Statistic           2.571 (df = 1; 234) 2.062 (df = 1; 234) 1.978 (df = 2; 233)
## -----
## Note:                 *p<0.1; **p<0.05; ***p<0.01

# Hmm... something is wrong!
# WAIT!!! We can't just look at price...
# Not all of the variation in price is from our experiment!
# The variation across movies in base-price is endogenous...
# We need to focus just on the price discount treatment itself...

t.test(leases~treated, data=MyData)

## 
## Welch Two Sample t-test
##
## data: leases by treated
## t = -1.8645, df = 217.08, p-value = 0.0636
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.45326122 0.01258325
## sample estimates:
## mean in group FALSE mean in group TRUE
##             0.5084746            0.7288136

ols_treat <- lm(leases ~ treated, data = MyData)
ols_log_discount <- lm(leases ~ log(disc+1), data = MyData)
# Does a positive coefficient make sense? Yes, discount is amount of money removed from price.
stargazer(ols_treat,ols_log_discount,type="text",column.labels=c("Binary","Log Discount"))

## -----
##                               Dependent variable:
##                               -----
##                               leases
##                               Binary    Log Discount
##                               (1)      (2)
## -----
## treated                  0.220*
##                           (0.118)
## 
## log(disc + 1)            0.055*
##                           (0.030)
## 
## Constant                0.508***   0.512***
##                           (0.084)   (0.083)
## 
## -----

```

```

## Observations           236          236
## R2                   0.015        0.014
## Adjusted R2          0.010        0.010
## Residual Std. Error (df = 234)  0.908        0.908
## F Statistic (df = 1; 234)      3.476*       3.401*
## =====
## Note:                  *p<0.1; **p<0.05; ***p<0.01

# What sort of heterogeneity might we look at here? And how?
# Let's check out base price.
ols_moderated_base <- lm(leases ~ treated*base_price, data=MyData)
ols_log_disc_moderated_base <- lm(leases ~ log(disc+1)*base_price, data=MyData)
stargazer(ols_moderated_base,ols_log_disc_moderated_base,
           type="text",
           column.labels=c("Treated Moderated by Base Price","Disc Moderated by Base Price"))

## 
## =====
##                                     Dependent variable:
##                                     -----
##                                     leases
## Treated Moderated by Base Price Disc Moderated by Base Price
## (1)                           (2)
## -----
## treated                      0.477
##                               (0.552)
## 
## log(disc + 1)                0.071
##                               (0.133)
## 
## base_price                   -0.001
##                               (0.001)          -0.001
##                               (0.001)
## 
## treatedTRUE:base_price      -0.001
##                               (0.002)
## 
## log(disc + 1):base_price    -0.0001
##                               (0.0004)
## 
## Constant                     0.695*
##                               (0.383)          0.782**
##                               (0.382)
## 
## Observations           236          236
## R2                   0.021        0.020
## Adjusted R2          0.009        0.007
## Residual Std. Error (df = 232)  0.909        0.909
## F Statistic (df = 3; 232)      1.673        1.551
## =====
## Note:                  *p<0.1; **p<0.05; ***p<0.01

# Why does the treatment effect disappear? Because it's the effect of treatment when
# base price = 0... this never actually occurs in the data!

```

```

# Let's shift the base price variable so it is mean 0.
# Then, the coefficient on treatment's main effect reflects treatment on the average movie.
MyData$log_base_price_demean <- log(MyData$base_price) - mean(log(MyData$base_price))
ols_moderated_dm <- lm(leases ~ treated*log_base_price_demean, data=MyData)
ols_log_disc_moderated_dm <- lm(leases ~ log(disc+1)*log_base_price_demean, data=MyData)
stargazer(ols_moderated_dm,ols_log_disc_moderated_dm,
type="text",column.labels=c("Base Price Moderator De-Meanned"),
"Log Disc Base Price Moderated De-Meanned")

## -----
## =====
##                               Dependent variable:
## -----
##                               leases
##                               Base Price Moderator De-Meanned
##                               (1)          (2)
## -----
## treated                      0.223*
##                               (0.118)
##
## log(disc + 1)                0.055*
##                               (0.030)
##
## log_base_price_demean        -0.141
##                               (0.336)      -0.227
##                               (0.335)
##
## treatedTRUE:log_base_price_demean -0.289
##                               (0.475)
##
## log(disc + 1):log_base_price_demean -0.024
##                               (0.115)
##
## Constant                     0.508***   0.513***
##                               (0.084)     (0.083)
## -----
## Observations                  236       236
## R2                           0.022     0.020
## Adjusted R2                   0.010     0.008
## Residual Std. Error (df = 232) 0.908     0.909
## F Statistic (df = 3; 232)    1.765     1.593
## -----
## Note:                         *p<0.1; **p<0.05; ***p<0.01
##
## =====
## Log Disc Base Price Moderated De-Meanned
## -----


# Nope, the effect doesn't seem to be moderated by baseline price.
# You can try it with a log transformation and you'll come to the same conclusion.
# What can we conclude?
# Nothing! Don't draw conclusions from null results...
# Try doing the same thing with likes...

```

```

MyData$likes_demean <- MyData$likes-mean(MyData$likes)
ols_moderated_likes_dm <- lm(leases ~ treated*likes_demean, data=MyData)
ols_log_disc_moderated__likes_dm <- lm(leases ~ log(disc+1)*likes_demean, data=MyData)
stargazer(ols_moderated_likes_dm,
           ols_log_disc_moderated__likes_dm,
type="text",column.labels=c("Like Moderator De-Meanned", "Log Price, Like Moderator De-Mean"))

## -----
## ====== Dependent variable: leases ======
## -----
##               Like Moderator De-Meanned Log Price, Like Moderator De-Mean
##               (1)                   (2)
## -----
## treated          0.221*
##                   (0.117)
## log(disc + 1)   0.055*
##                   (0.030)
## likes_demean    0.00000**   0.00000*** 
##                   (0.00000)  (0.00000)
## treatedTRUE:likes_demean -0.00000* 
##                   (0.00000)
## log(disc + 1):likes_demean -0.00000* 
##                   (0.00000)
## Constant        0.508***   0.512*** 
##                   (0.083)   (0.082)
## -----
## Observations    236        236
## R2              0.042      0.043
## Adjusted R2     0.029      0.030
## Residual Std. Error (df = 232) 0.899      0.899
## F Statistic (df = 3; 232)       3.356**   3.442**
## ======
## Note: *p<0.1; **p<0.05; ***p<0.01

# We do find that popular movies respond less strongly to the discount treatment.
# This makes some sense... if a movie is really good, "I don't care what it costs!"
# The strength of the moderation is pretty weak, however, in practical terms...
# We are going out to many significant digits... you can try the log transform here,
# But a better option might also be to just rescale the variable (e.g., 1,000's of likes)

```

## Session 4: Matching Techniques (matchit)

```
**** MSBA 6440 ****
**** Gordon Burtch and Gautam Ray****
**** Updated Feb 2020 ****
**** Code for Lecture 4 ****

# Analyzing Training Program Data
# Load Data

setwd("~/MSBA 2020 All Files/Spring 2020/MSBA 6440 - Causal Inference via Ecnmtrcs Exprmnt/Week 4 - Mat

# Read data
MyData<-read.csv("TrainingProgram.csv")

# Estimate Average Treatment Effect
mean(MyData$re78[MyData$treat == 1]) - mean(MyData$re78[MyData$treat == 0])

## [1] 886.3035

ols.model <- lm(re78 ~ treat + age + education +
                 black + hispanic + married + nodegree + re75 , data = MyData)
summary(lm(re78 ~ treat + age + education +
           black + hispanic + married + nodegree + re75 , data = MyData))

## 
## Call:
## lm(formula = re78 ~ treat + age + education + black + hispanic +
##      married + nodegree + re75, data = MyData)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9949   -4401   -1558    3056   54903 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3879.6048  2604.4327   1.490 0.136767  
## treat        806.5111   467.8870   1.724 0.085190 .  
## age          17.3925    36.1876   0.481 0.630934  
## education    175.3231   179.5810   0.976 0.329252  
## black        -1445.5412   801.4514  -1.804 0.071708 .  
## hispanic     98.4215   1046.1070   0.094 0.925069  
## married      71.8648   652.2659   0.110 0.912300  
## nodegree     -470.3970   742.9283  -0.633 0.526828  
## re75         0.1705     0.0467   3.651 0.000281 *** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6153 on 713 degrees of freedom
## Multiple R-squared:  0.0426, Adjusted R-squared:  0.03186 
## F-statistic: 3.966 on 8 and 713 DF,  p-value: 0.0001309
```

```

#Exact Matching
matchit(formula = treat ~ age + education + black +
        married + nodegree + re75 + hispanic, data = MyData, method = "exact")

## 
## Call:
## matchit(formula = treat ~ age + education + black + married +
##         nodegree + re75 + hispanic, data = MyData, method = "exact")
##
## Exact Subclasses: 38
##
## Sample sizes:
##           Control Treated
## All          425     297
## Matched      80      57
## Unmatched   345     240

exact.match <- matchit(formula= treat ~ age + education +
                       black + married + nodegree + re75 +
                       hispanic, data = MyData, method = "exact")

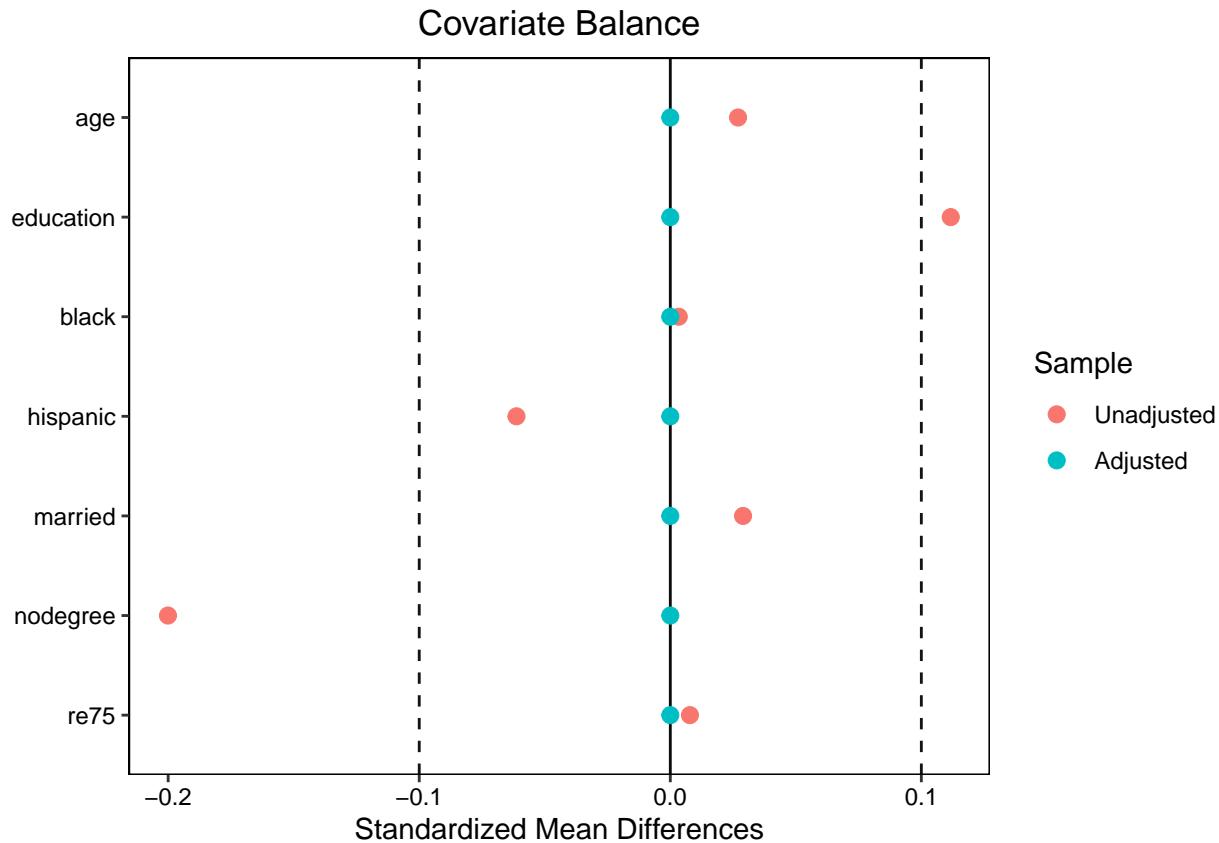
exact.data <- match.data(exact.match)

# Lets Check Covariate Balance
covs <- subset(MyData, select = -c(treat, re78))
m.out <- matchit(f.build("treat", covs), data = MyData, method = "exact")

love.plot(m.out, binary = "std", threshold = .1)

## Note: s.d.denom not specified; assuming pooled.

```



```
# Estimate Average Treatment Effect

exact.model <- lm(re78 ~ treat + age + education +
                    black + married + nodegree + re75 + hispanic, data = exact.data)

stargazer(ols.model, exact.model,
          title="Model with Different Types of Matches",
          type="text", column.labels=c("OLS Model", "Exact Matches"))
```

```
##
## Model with Different Types of Matches
## =====
##                               Dependent variable:
##                               -----
##                               re78
## OLS Model           Exact Matches
## (1)                  (2)
## -----
## treat                806.511*      1,111.872
##                      (467.887)    (1,001.031)
## 
## age                  17.392       39.186
##                      (36.188)    (106.032)
## 
## education            175.323      183.483
##                      (179.581)   (547.027)
```

```

##          -1,445.541*
##                (801.451)
##
##      98.422
##                (1,046.107)
##
##      71.865      -704.116
##                (652.266)      (2,476.021)
##
##      -470.397      -3,501.047*
##                (742.928)      (1,910.287)
##
##      0.170***      (0.047)
##
##      3,879.605      4,154.232
##                (2,604.433)      (6,646.500)
##
## -----
## Observations           722            137
## R2                   0.043          0.075
## Adjusted R2           0.032          0.040
## Residual Std. Error  6,152.525 (df = 713) 5,679.853 (df = 131)
## F Statistic          3.966*** (df = 8; 713) 2.134* (df = 5; 131)
## -----
## Note:                  *p<0.1; **p<0.05; ***p<0.01

```

## An Illustrative Example

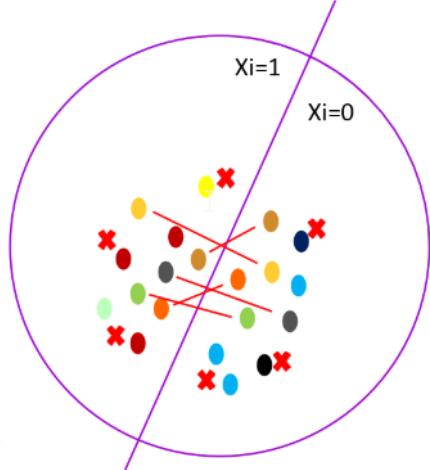
- Bryson(2002) examines the impact of union membership on wage premium.
- Matching based on demographic, job, and employer characteristics.
- Find a raw 17-25% union premium in gross hourly wages for the private sector in Britain.
- However, post-matching this difference falls to between 3 - 6%.

\* You don't have random assignment; you want to find two individuals with equal propensity to get treated, but only one gets and the other does not.

- Does union membership increase wages?

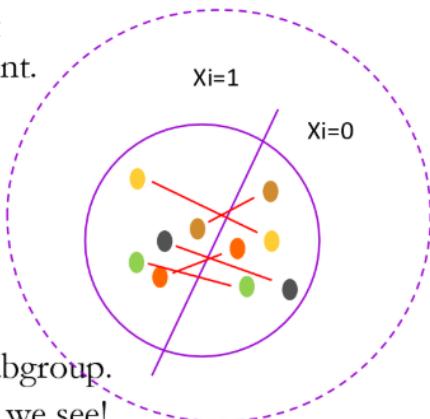
# Matching Techniques More Generally

- Primary Approach:
  - Propensity Score Matching.
- What's the Basic Idea?
  - You don't have randomized assignment; treatment is correlated with observed features that also impact the outcome directly.
  - We can't ignore it, we can't reliably control it away, and we don't have enough data for flexible non-parametric regression models.



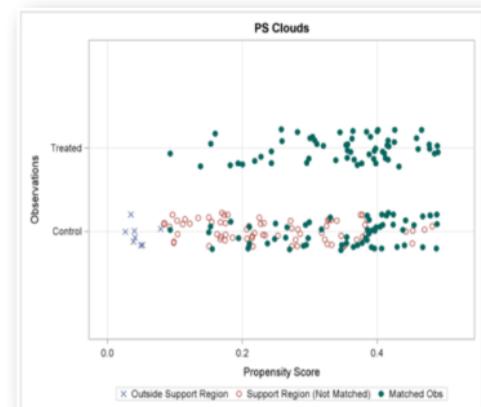
## Matching Prunes the Dataset

- Throw away data until treat and control look the same on all but treatment.
  - For every treated unit, try to find a control subject that looks exactly the same, except for the treatment.
- If I cannot find a match, throw away the subject.
  - Works, but requires a lot of data to get power.
- Downsides / Assumptions
  - We only recover the treatment effect for a local subgroup.
  - We assume treatment only depends on things that we see!



# Matching Step-by-Step

- Propensity Score Matching (PSM)
  - Check covariate balance
  - Model treatment as a function of the confounders, e.g., Logit or Probit.
  - Get predicted probability of treatment (propensity) from the model.
  - For each treated observation, find control observations with close propensity scores.
    - It is possible that no match can be found for a treated observation, in which case it is removed.



▷

```
# Propensity Score Matching
PS<- glm(treat ~ age + education +
           black + married + nodegree + re75 + hispanic,
           family = "binomial", data = MyData)

summary(PS)
```

```
##
## Call:
## glm(formula = treat ~ age + education + black + married + nodegree +
##       re75 + hispanic, family = "binomial", data = MyData)
##
## Deviance Residuals:
##    Min      1Q   Median      3Q     Max
## -1.2318 -0.9981 -0.9696  1.3521  1.4851
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 4.343e-01 8.599e-01  0.505  0.6135
## age         -1.624e-03 1.204e-02 -0.135  0.8927
## education   -2.378e-02 5.966e-02 -0.399  0.6901
## black        -9.743e-02 2.635e-01 -0.370  0.7116
## married      9.292e-02 2.158e-01  0.431  0.6668
## nodegree    -5.292e-01 2.443e-01 -2.167  0.0303 *
## re75        -3.121e-06 1.550e-05 -0.201  0.8404
## hispanic    -2.525e-01 3.481e-01 -0.725  0.4682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 978.09 on 721 degrees of freedom
## Residual deviance: 970.25 on 714 degrees of freedom
```

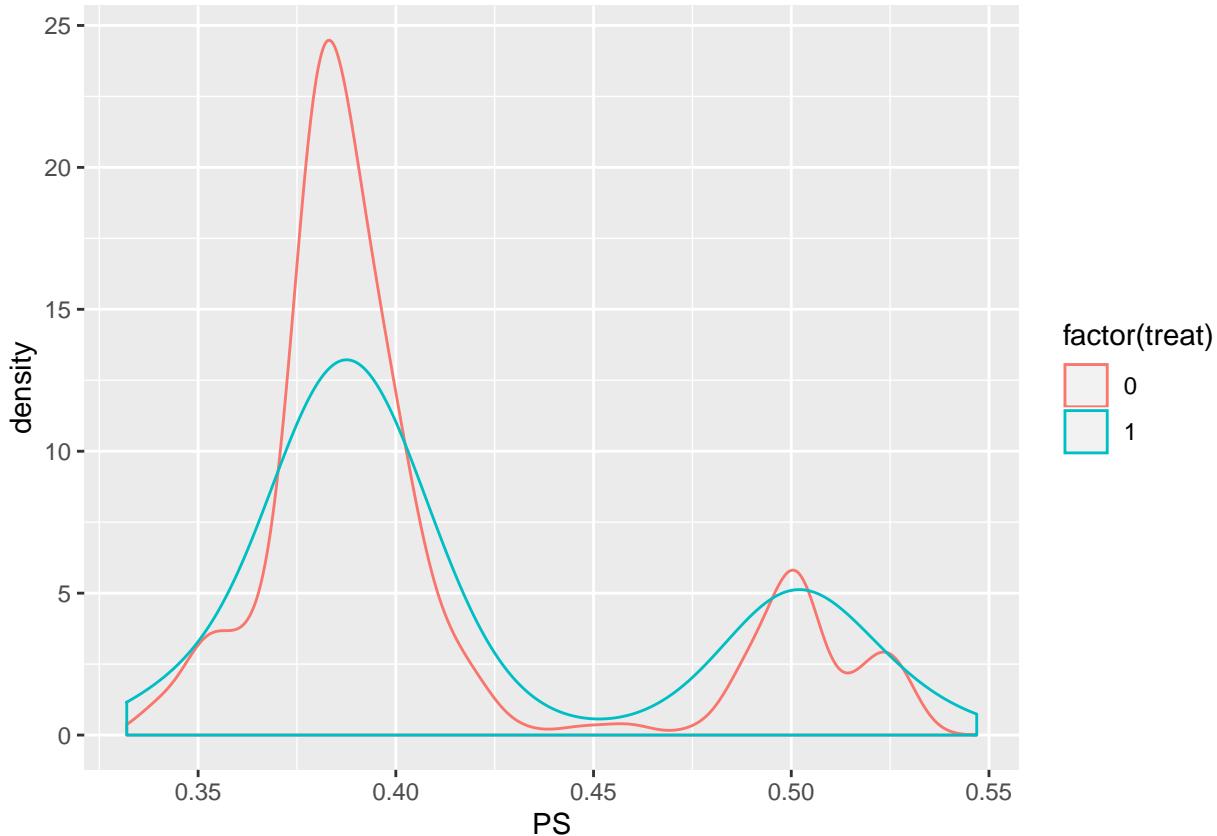
```

## AIC: 986.25
##
## Number of Fisher Scoring iterations: 4

# No Matching
MyData$PS<-glm(treat ~ age + education +
  black + married + nodegree + re75 +
  hispanic, data=MyData, family = "binomial")$fitted.values

ggplot(MyData, aes(x = PS, color = factor(treat))) +geom_density()

```



```

## The treated group and the control are pretty similar except for the large
# spike of the control group

# Propensity Score Matching - Nearest Match
nearest.match <- matchit(formula = treat ~ age + education +
  black + married + nodegree + re75 +
  hispanic, data = MyData, method = "nearest", distance = "logit")

matchit(formula = treat ~ age + education +
  black + married + nodegree + re75 +
  hispanic, data = MyData, method = "nearest", distance = "logit")

```

```

##
## Call:

```

```

## matchit(formula = treat ~ age + education + black + married +
##          nodegree + re75 + hispanic, data = MyData, method = "nearest",
##          distance = "logit")
##
## Sample sizes:
##           Control Treated
## All          425     297
## Matched      297     297
## Unmatched    128      0
## Discarded     0      0

```

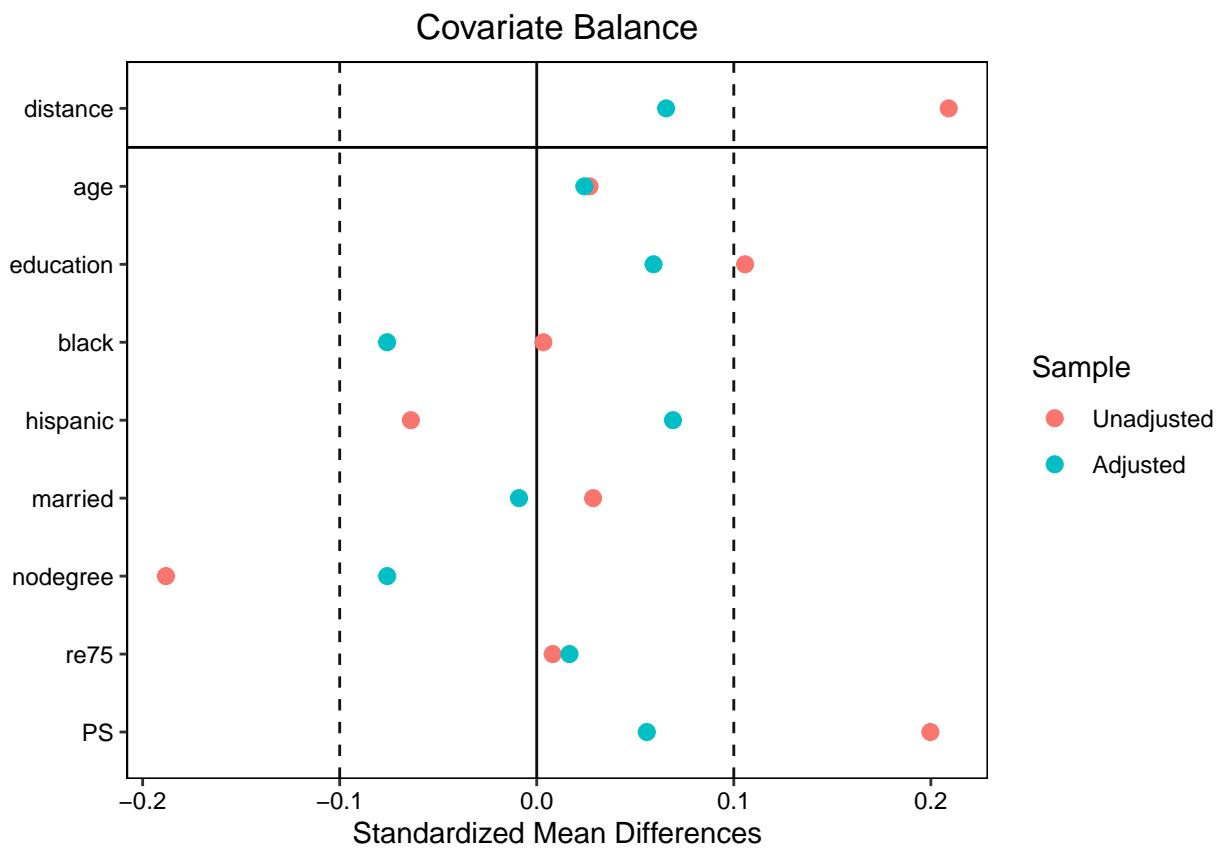
```

# Lets Check Covariate Balance
covs <- subset(MyData, select = -c(treat, re78))

m.out <- matchit(f.build("treat", covs),
                  data = MyData, method = "nearest", distance ="logit")

love.plot(m.out, binary = "std", threshold = .1)

```



```

# We are not matching on covariates, but instead matching on propensity
# score. On a covariate - covariate basis, they may not be an exact match.
# The nearest match will look better than the whole data set that was originally plotted.

```

## Benefits of Matching

- When # Confounders is Large Relative to Sample Size
  - Helps the matching process by collapsing many observed variables into a single feature – the propensity score.
- Provides a Way to Recover “As-Good-As” Randomization, Post Hoc
  - If my randomization breaks (or I’m using an observational dataset) I can construct an experimental setup after the fact.
- Forces Attention to Covariate Imbalance
  - Forces researcher to evaluate covariate balance before estimating effects.
  - Note: this can also be used as a way of performing “randomization checks.”

## Some Issues

What Problems Do You See Here?

1. Omitted Variable Bias – Still don’t know what we don’t know. PSM assumes treatment assignment is fully determined by observed variables.
2. Lack of Support – There may not be any good matches. Choice of caliper can be somewhat arbitrary.
3. Propensity Function – Logit or Probit is just as arbitrary as conditioning on the controls linearly! Note: you do still get benefits of dimensionality reduction.



# Some Solutions

- What Can We Do About Them?

1. Nothing – there is no way to evaluate OVB empirically (we can ‘eliminate’ some forms – more on this later).
2. Evaluate Quality of Matches – e.g., plot densities to see common support i.e., visually check matching quality.
3. Use Something More Flexible – Nothing prevents you from implementing the best flexible predictive model you can come up with (e.g., Random Forest). Alternatively, other matching algorithms e.g., CEM (more on this one shortly).



## Quality of Matches

- Where Are We Okay vs. Not?

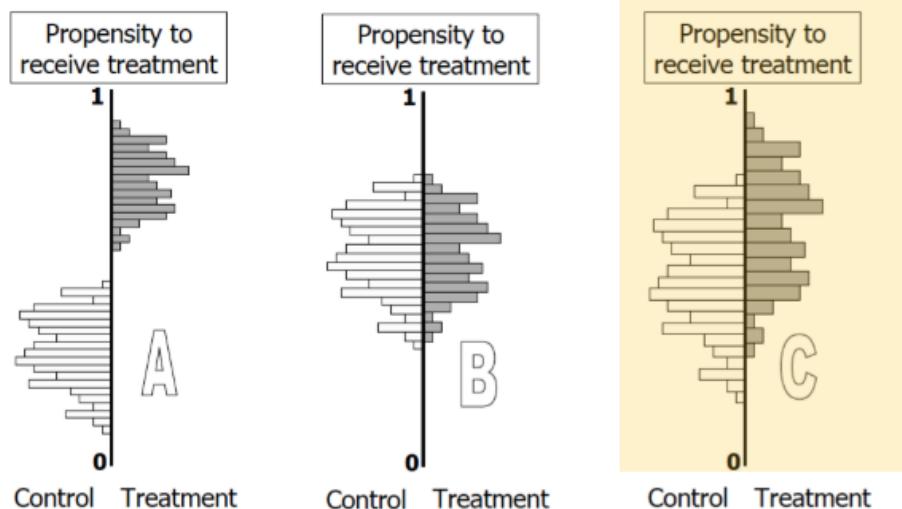


Figure 11: Improve by removing observations that do not match. B is the goal.

# Notes: Arbitrary Propensity Functions

## • Option A: Coarsened Exact Matching (CEM)

- Motivation: Collapsing all variables into a single propensity score seems too arbitrary, but directly matching on all variables is too hard (and sometimes impossible). Instead, let's directly match on discretized versions of all variables.
- We don't model treatment as a function of confounders. Instead, we directly enforce balance on each confounder, separately.
  - Discretize each variable into categories. Then, for each treated observation, find control observations that match exactly on all discretized variables.
- Binning algorithms (same ones that underpin histograms) are used to chop variables into bins. You are only a good match on a given confounder if you fall in the same bin.

- Theory of CEM shows that it can clearly improve covariate balance.

```
# Coarsend Exact Matching
# Automatic Coarsend Exact Matching
autocem.match <- matchit(formula = treat ~ age + education +
                         black + married + nodegree + re75 +
                         hispanic, data = MyData, method = "cem")

##
## Using 'treat'='1' as baseline group

matchit(formula = treat ~ age + education +
         black + married + nodegree + re75 +
         hispanic, data = MyData, method = "cem")

##
## Using 'treat'='1' as baseline group

##
## Call:
## matchit(formula = treat ~ age + education + black + married +
##           nodegree + re75 + hispanic, data = MyData, method = "cem")
## 

## Sample sizes:
##          Control Treated
## All        425     297
## Matched    299     212
## Unmatched  126      85
## Discarded   0       0

# Matched 299 212 - Coarsened exact match
```

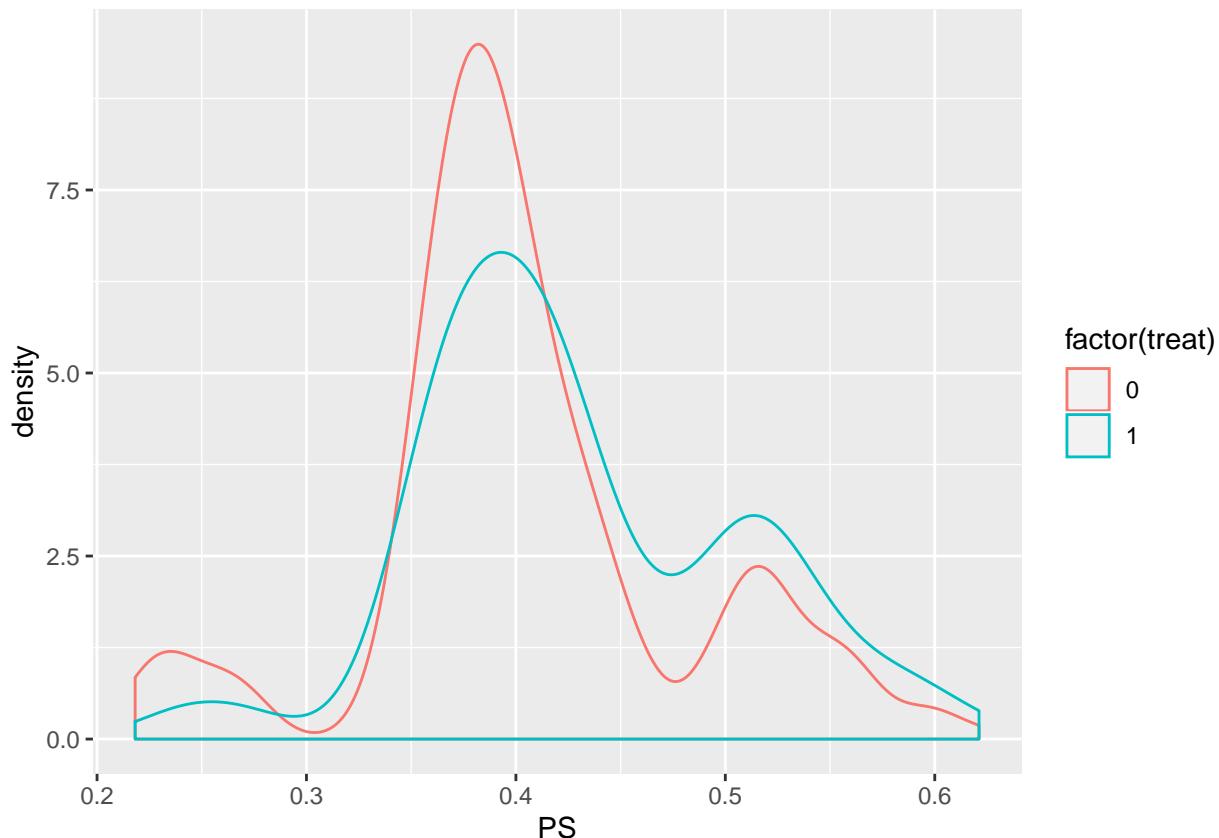
```

autocem.data <- match.data(autocem.match)

autocem.data$PS<-glm(treat ~ age + education +
                      black + married + nodegree + re75 +
                      hispanic, data=autocem.data, family = "binomial")$fitted.values

ggplot(autocem.data, aes(x = PS, color = factor(treat))) + geom_density()

```



*# Matching is not as effective because we're not using propensity score to match.*

*# Lets Check Covariate balance*

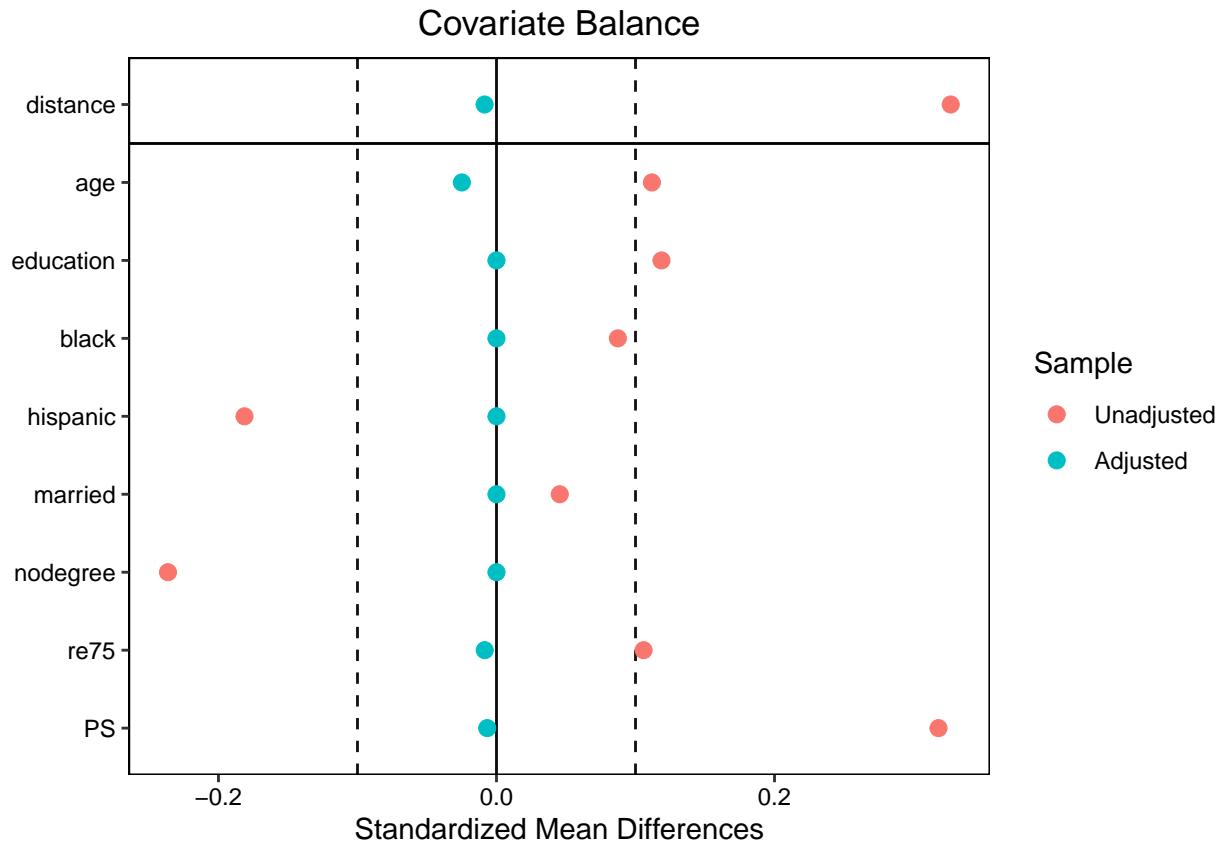
```
m.out <- matchit(f.build("treat", covs), data = autocem.data, method = "cem")
```

##

## Using 'treat'='1' as baseline group

```
love.plot(m.out, binary = "std", threshold = .1)
```

## Note: s.d.denom not specified; assuming pooled.



```
# Now we match on covariates, and that's why the plot looks better - it's a 1-1 match.
```

## Sensitivity Analysis

- More Generally...
  - It is a good idea to explore robustness of the result to alternative matching techniques, PSM vs CEM
  - Try different propensity functions (logit vs. probit vs. something else).
  - Matching with replacement vs. matching without replacement.
  - Nearest-neighbor versus a caliper.

Time Shift TV Ex.

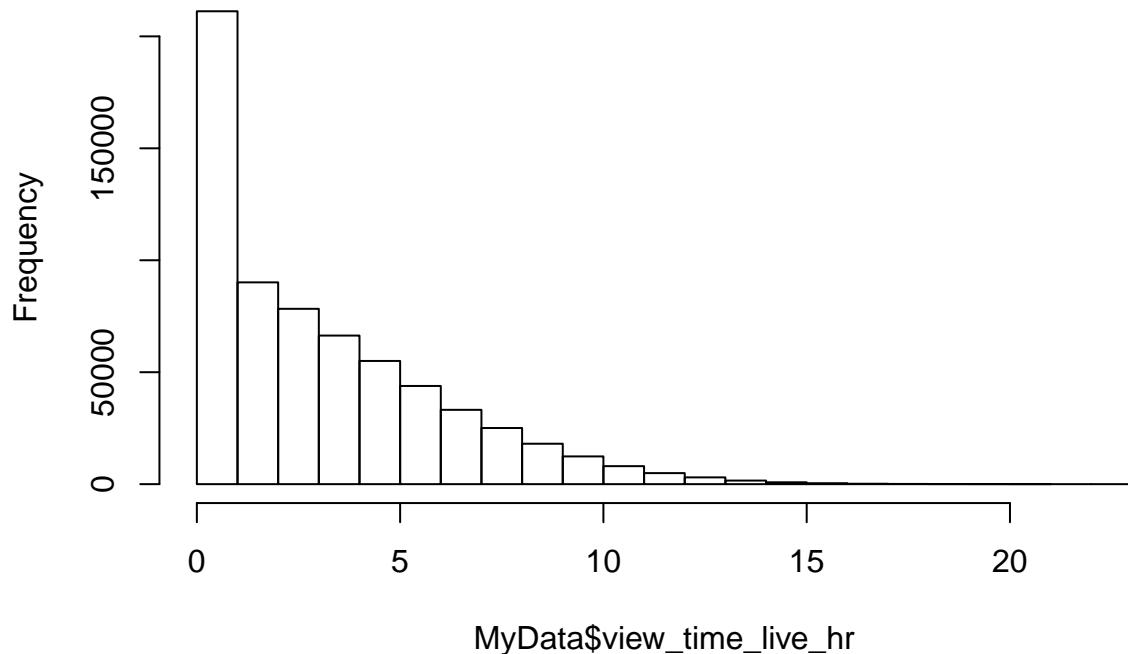
```

setwd("~/MSBA 2020 All Files/Spring 2020/MSBA 6440 - Causal Inference via Econometrics Exprmnt/Week 4 - Mat
***** Load the data ****#
MyData<-read.csv("TSTV-Obs-Dataset-2.csv")

hist(MyData$view_time_live_hr)

```

## Histogram of MyData\$view\_time\_live\_hr



```

***** Let's get a sense of the data ****#

```

```

#how long is the period of observation?

```

```

max(MyData$week)-min(MyData$week)

```

```

## [1] 13

```

```

#How many subjects got TSTV? (Treated)

```

```

length(unique(MyData$id[MyData$premium==TRUE]))

```

```

## [1] 8348

```

```

#How many subjects did not get TSTV? (Control)

```

```

length(unique(MyData$id[MyData$premium==FALSE]))

```

```

## [1] 41686

```

```

#In what 'week' does the "treatment" begin?
min(unique(MyData$week[MyData$after==TRUE])) 

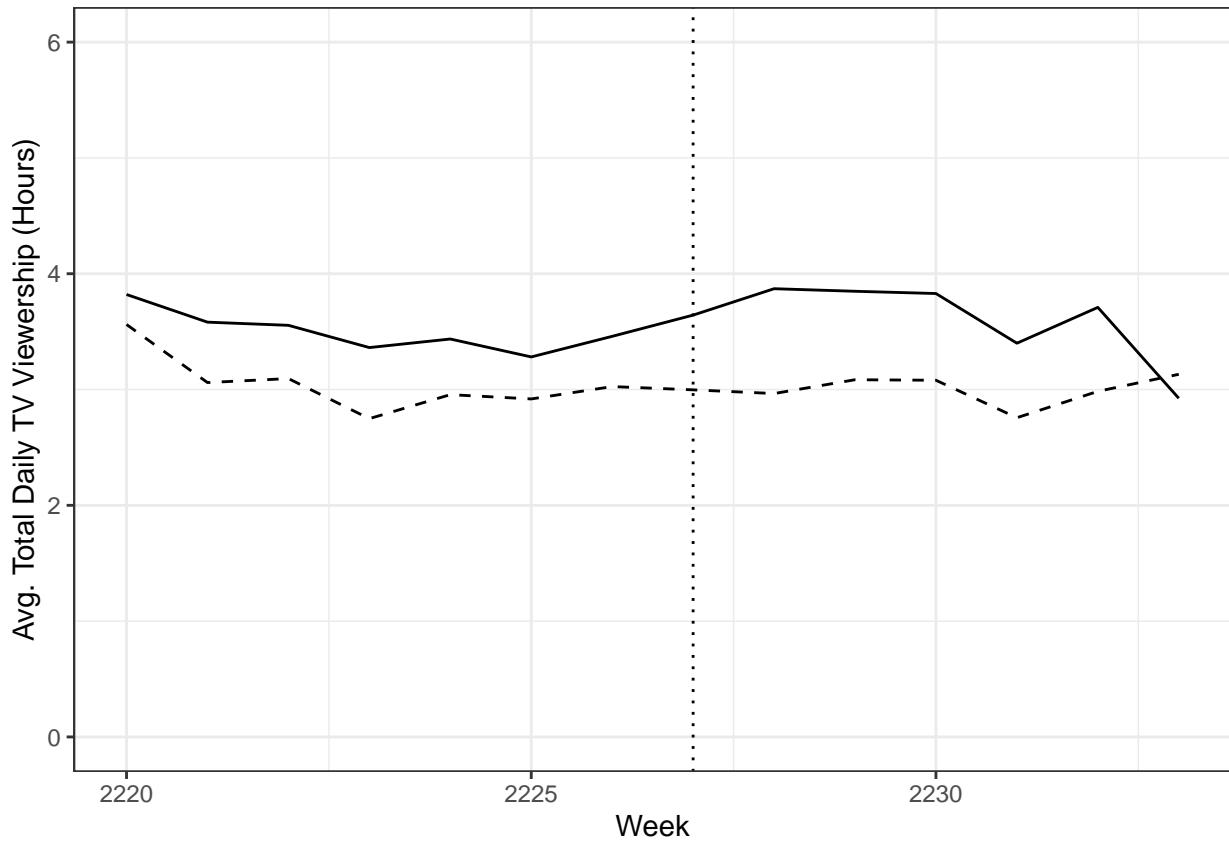
## [1] 2227

#Let's just look at what is going on with average viewership behavior
#for treated vs. untreated, in the weeks around the treatment date.
MyDataAggregated <- aggregate(MyData,
                                by=list(MyData$premium, MyData$week), FUN=mean)

# plot for total TV time
p <- ggplot(MyDataAggregated)
p <- p +
  geom_line(data=MyDataAggregated[MyDataAggregated$premium==FALSE,],
            aes(week, view_time_total_hr), linetype='dashed') # non premium
p <- p +
  geom_line(data=MyDataAggregated[MyDataAggregated$premium==TRUE,],
            aes(week, view_time_total_hr), linetype='solid') # premium

p <- p + geom_vline(xintercept=2227, linetype='dotted')
p <- p + xlab("Week") + ylab("Avg. Total Daily TV Viewership (Hours)")
p <- p + ylim(0, 6) + xlim(2220,2233) + theme_bw()
p

```

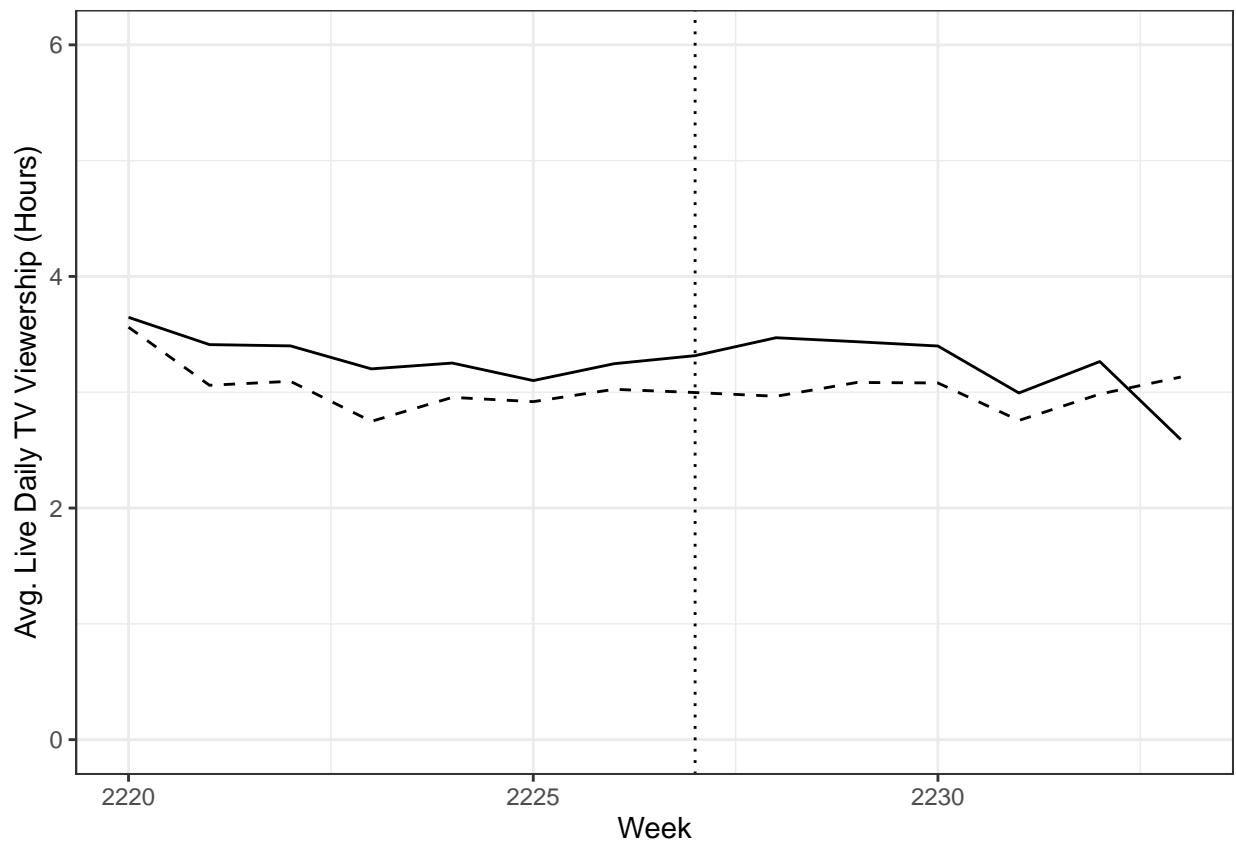


```

# plot for live TV time
p <- ggplot(MyDataAggregated)
p <- p +
  geom_line(data=MyDataAggregated[MyDataAggregated$premium==FALSE,],
            aes(week, view_time_live_hr), linetype='dashed')
p <- p +
  geom_line(data=MyDataAggregated[MyDataAggregated$premium==TRUE,],
            aes(week, view_time_live_hr), linetype='solid')

p <- p + geom_vline(xintercept=2227, linetype='dotted')
p <- p + xlab("Week") + ylab("Avg. Live Daily TV Viewership (Hours)")
p <- p + ylim(0, 6) + xlim(2220,2233) + theme_bw()
p

```



```

# The dip shows that viewers with TSTV started watching less live TV.

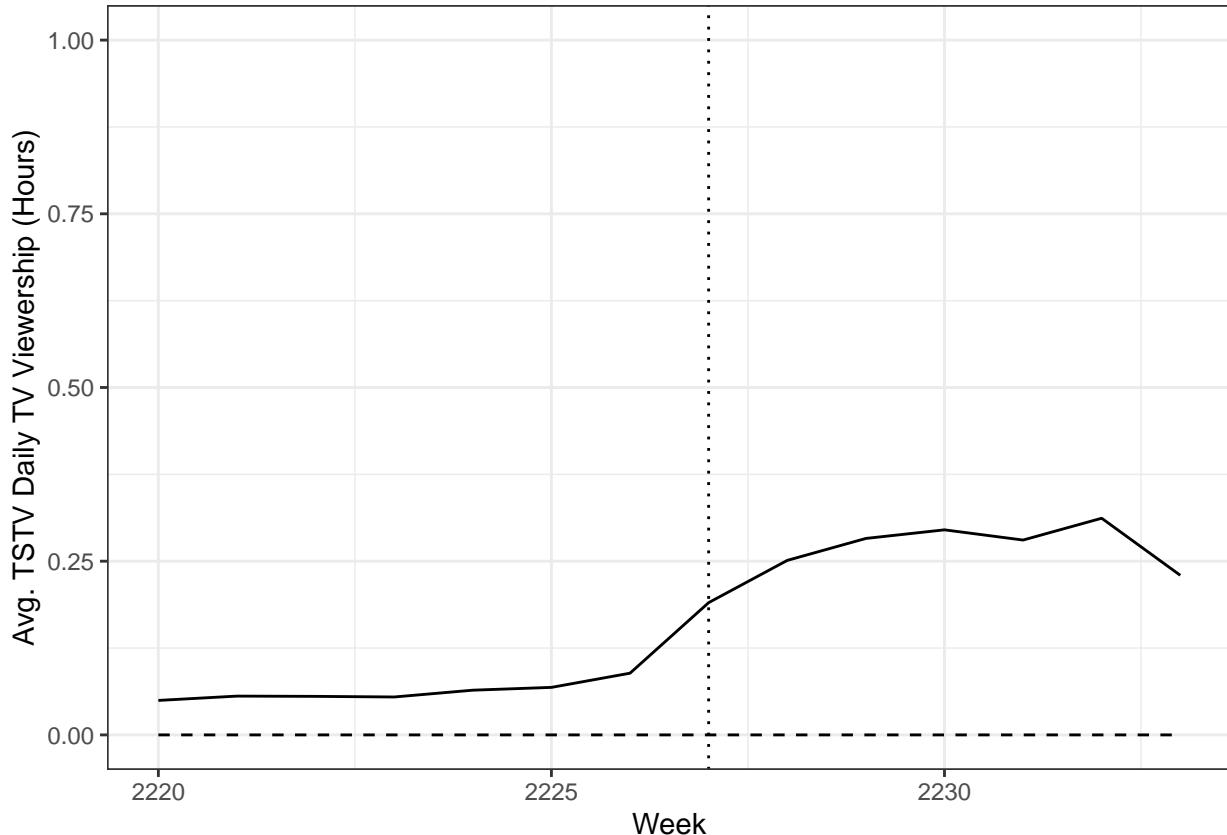
# plot for TSTV time
p <- ggplot(MyDataAggregated)
p <- p +
  geom_line(data=MyDataAggregated[MyDataAggregated$premium==FALSE,],
            aes(week, view_time_tstv_hr), linetype='dashed')
p <- p +
  geom_line(data=MyDataAggregated[MyDataAggregated$premium==TRUE,],
            aes(week, view_time_tstv_hr), linetype='solid')

```

```

p <- p + geom_vline(xintercept=2227, linetype='dotted')
p <- p + xlab("Week") + ylab("Avg. TSTV Daily TV Viewership (Hours)")
p <- p + ylim(0, 1) + xlim(2220,2233) + theme_bw()
p

```



```

##### Propensity Score Matching #####
#For this demonstration, we will use data from the pre-period for matching.
#We will then estimate the effect of TSTV gifting in the post period.

##### CREATE A SUMMARY DATASET BEFORE vs. AFTER TSTV IS AVAILABLE #####
MyDataSummary <- aggregate(MyData,by=list(MyData$id,MyData$after),FUN=mean)
MyDataSummary$view_time_total_sq <- MyDataSummary$view_time_total_hr^2

# Okay, let's check out our covariate balance; we have one confounder here, view_time_total_hr.
# This is a dependent variable, but we are going to match on it in the pre-period.
# That is, we only want subjects who had similar viewership activity before TSTV showed up.

MyPreData <- MyDataSummary[MyDataSummary$after == FALSE,]

tabUnmatched <- CreateTableOne(vars=c("view_time_total_hr",
                                         "view_time_total_sq"), strata="premium",
                                         test=TRUE,data=MyPreData)
print(tabUnmatched, smd=TRUE)

```

```

## Stratified by premium
##          0          1      p     test
## n       41686     8348
## view_time_total_hr (mean (SD)) 2.98 (2.42) 3.46 (2.69) <0.001
## view_time_total_sq (mean (SD)) 14.72 (22.20) 19.21 (28.21) <0.001
## Stratified by premium
## SMD
## n
## view_time_total_hr (mean (SD)) 0.191
## view_time_total_sq (mean (SD)) 0.177

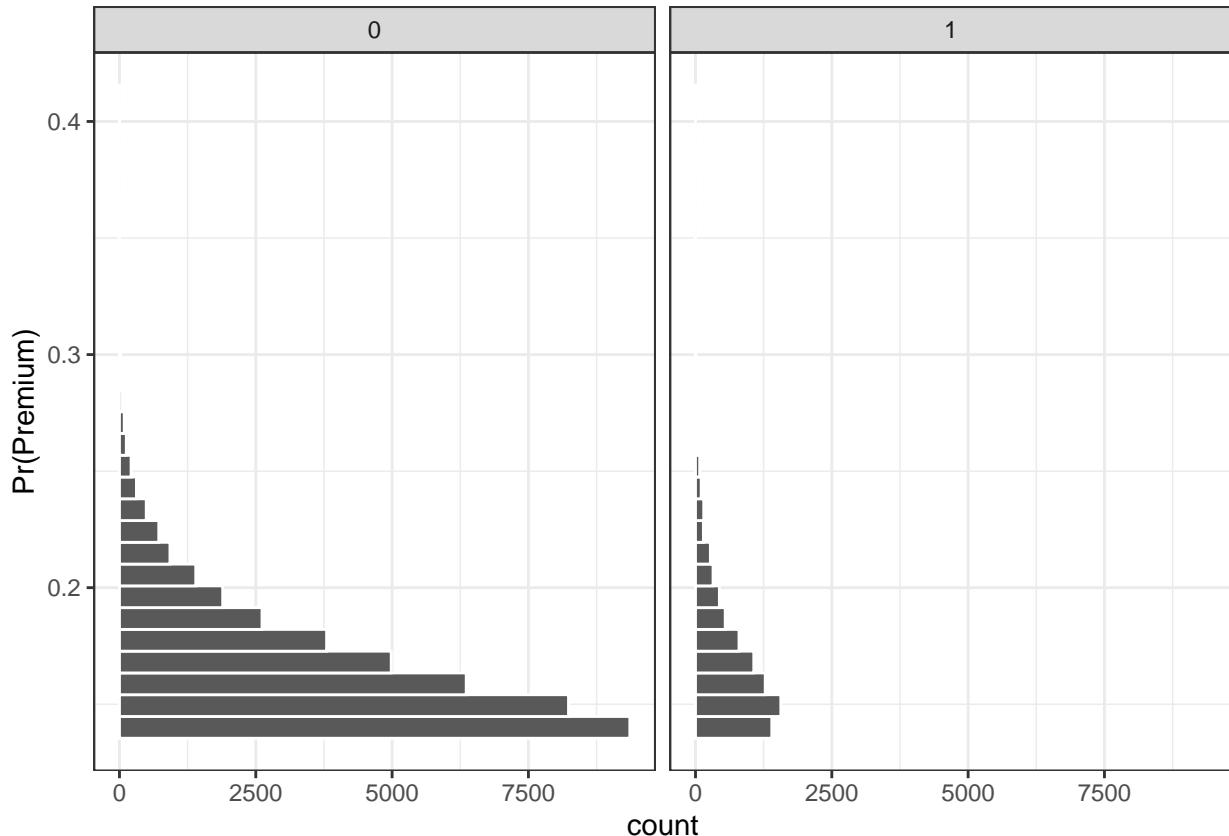
# On average, the premium group watches 30 mins more TV than the control group.

# Whoa, lots of imbalance here...
# Let's see what propensity scores look like...

MyPreData$PS<-glm(premium~view_time_total_hr+view_time_total_sq,
                     data=MyPreData, family = "binomial")$fitted.values

ggplot(MyPreData, aes(x = PS)) +
  geom_histogram(color = "white") +
  facet_wrap(~premium) + xlab("Pr(Premium)") + theme_bw() + coord_flip()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```

**** Match treated and control households on propensity to receive premium based
# on pre-treatment time watching TV ***#
# Note: the matchit command may take a long time to run with large datasets

Matched_Output <- matchit(premium ~ view_time_total_hr +
                           view_time_total_sq, data = MyPreData,
                           method = 'nearest', distance = "logit", caliper = 0.001, replace = FALSE)

summary(Matched_Output)

##
## Call:
## matchit(formula = premium ~ view_time_total_hr + view_time_total_sq,
##          data = MyPreData, method = "nearest", distance = "logit",
##          caliper = 0.001, replace = FALSE)
##
## Summary of balance for all data:
##           Means Treated Means Control SD Control Mean Diff
## distance      0.1714     0.1659    0.0267  0.0055
## view_time_total_hr   3.4632     2.9754    2.4220  0.4878
## view_time_total_sq   19.2077    14.7191   22.1998  4.4887
##           eQQ Med eQQ Mean eQQ Max
## distance      0.0042     0.0055    0.0288
## view_time_total_hr  0.4336     0.4873    1.6377
## view_time_total_sq   2.3247     4.4755   39.9870
##
##
## Summary of balance for matched data:
##           Means Treated Means Control SD Control Mean Diff
## distance      0.1686     0.1686    0.0262  0e+00
## view_time_total_hr   3.2493     3.2492    2.3885  1e-04
## view_time_total_sq   16.2621    16.2613   21.4349  8e-04
##           eQQ Med eQQ Mean eQQ Max
## distance      0.0000     0.0000    0.0000
## view_time_total_hr  0.0007     0.0008    0.0033
## view_time_total_sq   0.0032     0.0056    0.0369
##
## Percent Balance Improvement:
##           Mean Diff. eQQ Med eQQ Mean eQQ Max
## distance      99.9779    99.8337   99.8456 99.9042
## view_time_total_hr  99.9759    99.8366   99.8324 99.7999
## view_time_total_sq   99.9824    99.8620   99.8756 99.9076
##
## Sample sizes:
##           Control Treated
## All        41686    8348
## Matched     8112    8112
## Unmatched   33574    236
## Discarded      0      0

Matched.ids <- data.table(match.data(Matched_Output))$id

Matched_Data = match.data(Matched_Output)

```

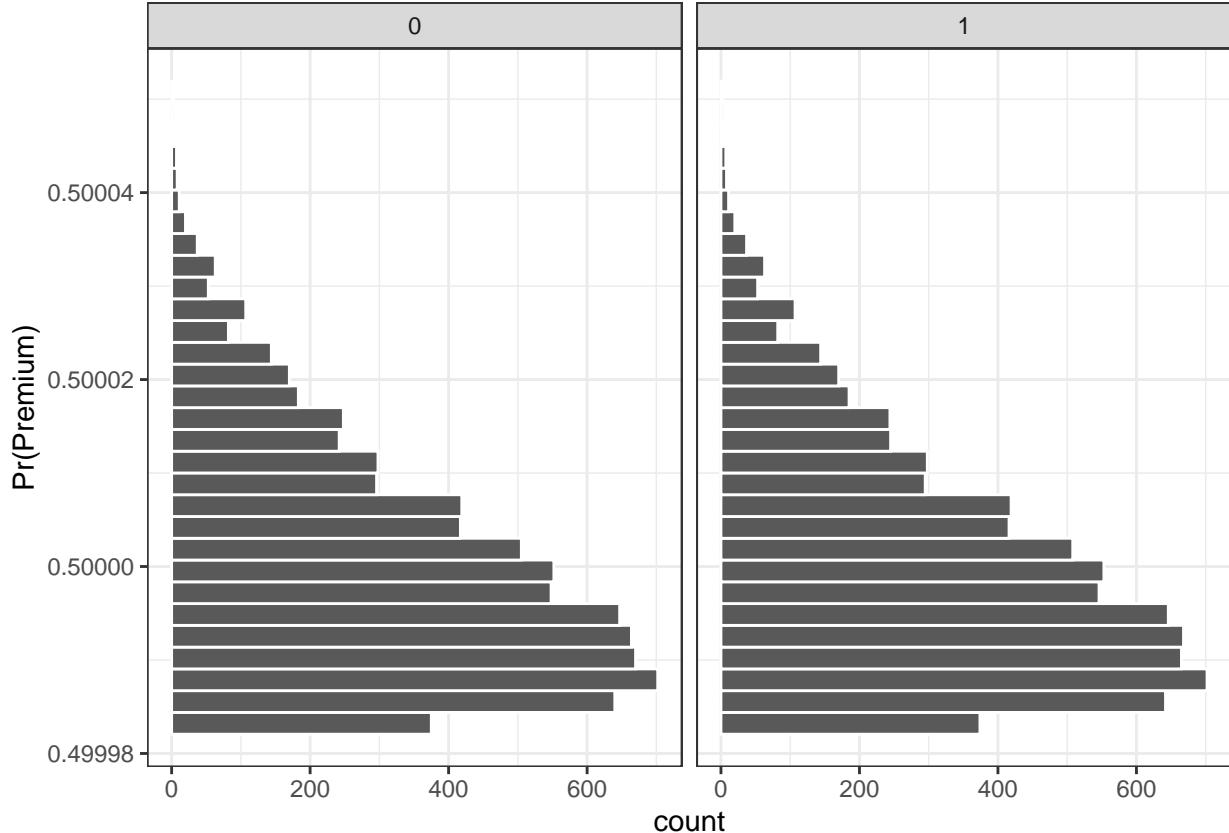
```

Matched_Data$PS = glm(premium ~ view_time_total_hr,
                      data = Matched_Data, family = "binomial")$fitted.values

ggplot(Matched_Data, aes(x = PS)) +
  geom_histogram(color = "white") +
  facet_wrap(~premium) + xlab("Pr(Premium)") + theme_bw() + coord_flip()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```



```

# There is some overlap between the groups.

tabMatched <- CreateTableOne(vars=c("view_time_total_hr",
                                      "view_time_total_sq"), strata="premium", test=TRUE,data=Matched_Data)

print(tabMatched, smd=TRUE)

```

Stratified by premium			
	0	1	p
## n	8112	8112	
## view_time_total_hr (mean (SD))	3.25 (2.39)	3.25 (2.39)	0.997
## view_time_total_sq (mean (SD))	16.26 (21.43)	16.26 (21.44)	0.998
##	Stratified by premium		
##	SMD		
## n			

```

##   view_time_total_hr (mean (SD)) <0.001
##   view_time_total_sq (mean (SD)) <0.001

#Now let's estimate the treatment effect with vs. without matching.
MyDataPost <- MyDataSummary[MyDataSummary$after==TRUE,]
unmatched_ate <- lm(data=MyDataPost,view_time_total_hr~premium)
matched_ate <- lm(data=MyDataPost[MyDataPost$id %in% Matched.ids,], view_time_total_hr ~ premium)

#Produce the output table.
stargazer(unmatched_ate,matched_ate,
           title="Matched vs. Unmatched Estimates",
           column.labels=c("Total Viewership", "Total Viewership"),type="text")


$$\begin{array}{l}
\text{##} \\
\text{## Matched vs. Unmatched Estimates} \\
\text{## =====} \\
\text{##} \quad \text{Dependent variable:} \\
\text{##} \quad \text{-----} \\
\text{##} \quad \text{view_time_total_hr} \\
\text{##} \quad \begin{array}{cc} \text{Total Viewership} & \text{Total Viewership} \\ (1) & (2) \end{array} \\
\text{##} \quad \text{-----} \\
\text{##} \quad \text{premium} & 0.614*** \\
\text{##} & (0.031) \\
\text{##} \\
\text{##} \quad \text{Constant} & 2.990*** \\
\text{##} & (0.013) \\
\text{##} \\
\text{##} \quad \text{Observations} & 48,483 \\
\text{##} \quad \text{R2} & 0.008 \\
\text{##} \quad \text{Adjusted R2} & 0.008 \\
\text{##} \quad \text{Residual Std. Error} & 2.586 (df = 48481) \quad 2.531 (df = 15918) \\
\text{##} \quad \text{F Statistic} & 388.341*** (df = 1; 48481) \quad 27.086*** (df = 1; 15918) \\
\text{##} \quad \text{-----} \\
\text{## Note:} & *p<0.1; **p<0.05; ***p<0.01
\end{array}$$


# If we include propensity matching, then we don't see as much of an increase
# because most people are already watching too much TV.

```

## Session 5: Fixed Effect Regression (plm)

### Confounders Are Likely (multiple observations by company)

- Imagine that the true data-generating process looks like this..
  - Rho refers to correlation here.

$$\begin{aligned}
Y_{it} &= \alpha + \beta_1 Treat_{it} + \gamma_1 Confound_i + \mu_i \\
\rho(Treat, Confound) &\neq 0
\end{aligned}$$

\* Not the subscripts... + The confound is not time-varying, whereas Y and Treat and time varying. + We can take advantage of this fact! + We can transform the data to make the confound disappear..

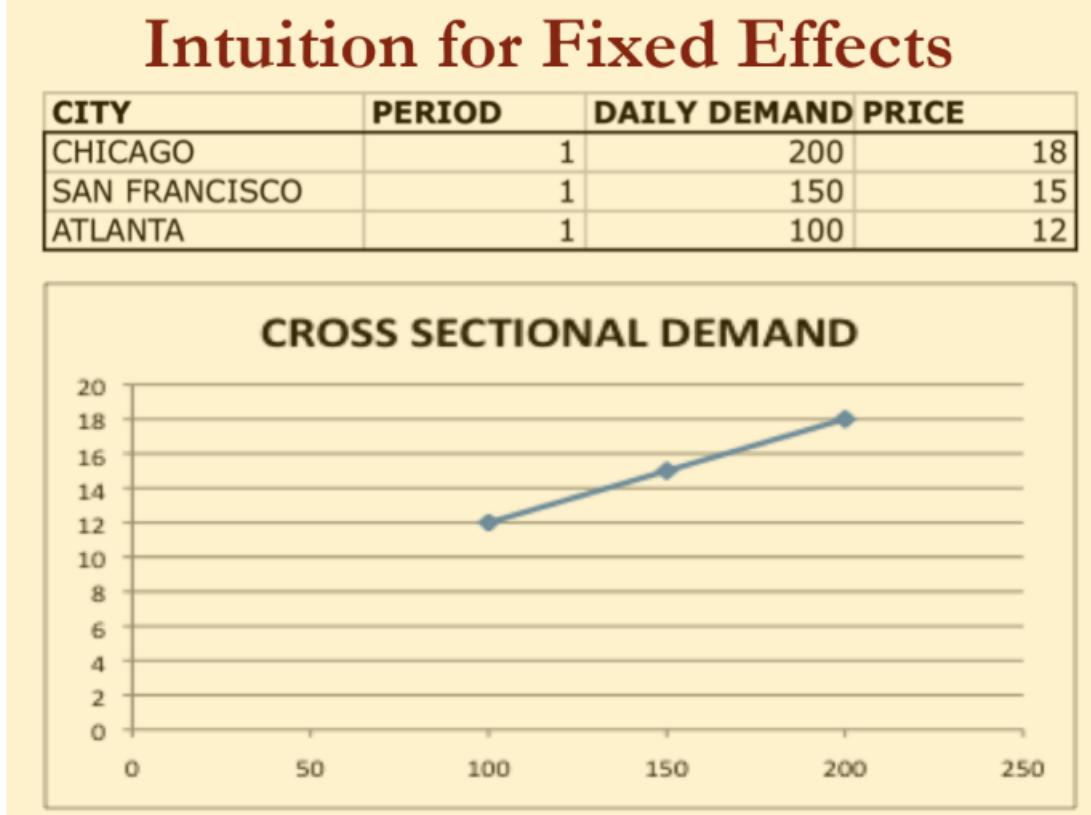
## Fixed Effect Regression

I

- De-meaning the time-series to get rid of the Confound.
  - Subtract the average of every variable from itself.
  - Also known as a ‘within transformation’.
  - This is officially what people mean when they talk about fixed effect regression.
- Time-invariant variables, demeaned, go to 0.

$$(Y_{it} - \bar{Y}_i) = \beta_1(Treat_{it} - \bar{Treat}_i) + \gamma_i(Confound_i - \bar{Confound}_i) + (u_{it} - \bar{u}_i)$$

$$(Y_{it} - \bar{Y}_i) = \beta_1(Treat_{it} - \bar{Treat}_i + (u_{it} - \bar{u}_i))$$



mand of pizza goes up with price; fixed effect of the city.

\* De-

## Intuition for Fixed Effects

CITY	PERIOD	DAILY DEMAND	PRICE
CHICAGO	1	200	18
	2	180	20
SAN FRANCISCO	1	150	15
	2	125	17
ATLANTA	1	100	12
	2	85	16



## Some Issues

- Some Questions...

1. What if your variable of interest is not time-varying?
2. What if you are worried about time varying confounds?

- Some Answers

1. You cannot study the effect of the variable under this framework. You may just have to live with controls or matching.
2. This approach does not resolve dynamic confounds.



## Other Approaches: LSDV

- Least-Squares Dummy Variable.

- Include a dummy for every subject ID.
- This dummy subsumes / absorbs anything about the subject that does not change across observations.
- Note: remember the dummy variable trap; you need to omit one dummy as the “reference group.”

- LSDV is mathematically equivalent to the Within Transform.

- This derives from the Frisch-Waugh-Lovell Theorem.

## Inter-Temporal Differences

- Work with numerical first derivatives (slopes).
- Take  $dY/dt$  of our regression equation also causes Confound to go to 0.

$$Y_{it} = \alpha + \beta_1 Treat_{it} + \gamma_1 Confound_i + u_{it}$$

- Numerically, this is just transforming the data into time series differences... e.g., first differences.

$$Y_{it} - Y_{it-1} = \beta(Treat_{it} - Treat_{it-1}) + (u_{it} - u_{it-1})$$

## When to Use Which Estimator?

- First Difference estimator has weaker assumptions
  - We allow correlation between our variables and error term.
  - We just require that their rates of change (slopes) not be correlated.

$$E[u_{it} - u_{it-1} | x_{it} - x_{it-1}] = 0$$

- The trade-off is that the estimator is less efficient
  - It requires more data to arrive at the right estimate of beta.
  - The question is whether the trade-off is worthwhile.
  - MHE talks a bit about when FD and Within are preferable; it also mentions that their estimates “bracket” the true estimate of Beta.

## Random Effects Model

- Treat  $Confound_i$  as a random variable.
- $Confound_i$  is assumed to be uncorrelated with  $X_i$ .
- If the time-invariant  $Confound_i$  is truly uncorrelated with  $X_i$ , i.e., the assumption behind random effect regression holds, then random effect regression is more efficient than fixed effect regression.
- Use the **Hausman** Test. If it is rejected, use fixed effect regression.

```

setwd("~/MSBA 2020 All Files/Spring 2020/MSBA 6440 - Causal Inference via Econometrics Experiments/Week 5 - Fixed Effects Models")

PData<-read.csv("KoopTobias.csv")

# Let's try a fixed effect regression.

within_reg <- plm(data=PData, LOGWAGE~EDUC,
                    index=c("PERSONID"), effect="individual", model="within")

pooling_reg <- plm(data=PData, LOGWAGE~EDUC,
                     index=c("PERSONID"), effect="individual", model="pooling")

ols_reg <- lm(data=PData, LOGWAGE~EDUC)

# Let's see if panel data model is needed

pFtest(within_reg, pooling_reg)

## 
## F test for individual effects
##
## data: LOGWAGE ~ EDUC
## F = 8.949, df1 = 2177, df2 = 15740, p-value < 2.2e-16
## alternative hypothesis: significant effects

stargazer(within_reg, pooling_reg, ols_reg,
           title="Within vs. Pooling Models vs. OLS",
           column.labels = c("Within", "Pooling", "OLS"), type="text")

```

Dependent variable: LOGWAGE			
	panel	linear	OLS
## EDUC	Within (1)	Pooling (2)	OLS (3)
## Constant		1.330*** (0.025)	1.330*** (0.025)
## Observations	17,919	17,919	17,919
## R2	0.064	0.077	0.077
## Adjusted R2	-0.065	0.077	0.077
## Residual Std. Error			0.507 (df = 17917)
## F Statistic	1,084.241*** (df = 1; 15740)	1,496.940*** (df = 1; 17917)	1,496.940*** (df = 1; 17917)

```

## =====
## Note: *p<0.1; **p<0.05; ***p<0.01

# Fixed Effect vs Random Effect

within_reg <- plm(data=PData,LOGWAGE~EDUC + ABILITY,
                   index=c("PERSONID"),effect="individual",model="within")

random_reg = plm(LOGWAGE ~ EDUC + ABILITY, data = PData,
                  index=c("PERSONID"), effect="individual", model="random")

stargazer(within_reg,random_reg,
           title="Fixed vs. Random Effect Model",
           column.labels = c("Within", "Random"), type="text")

## =====
## Fixed vs. Random Effect Model
## =====
## Dependent variable:
## -----
##          LOGWAGE
##          Within      Random
##          (1)        (2)
## -----
## EDUC          0.198***    0.114***  

##             (0.006)    (0.004)
## 
## ABILITY          0.004  

##                 (0.010)
## 
## Constant         0.808***  

##                 (0.049)
## 
## -----
## Observations     17,919      17,919
## R2              0.064       0.137
## Adjusted R2     -0.065      0.137
## F Statistic   1,084.241*** (df = 1; 15740) 2,638.582***  

## =====
## Note: *p<0.1; **p<0.05; ***p<0.01

# Hausman test
phptest(within_reg, random_reg)

## 
## Hausman Test
## 
## data: LOGWAGE ~ EDUC + ABILITY
## chisq = 333.13, df = 1, p-value < 2.2e-16
## alternative hypothesis: one model is inconsistent

```

```

# Serial Correlation (Breusch Godfrey Test)
pbgttest(within_reg)

## 
## Breusch-Godfrey/Wooldridge test for serial correlation in panel
## models
## 
## data: LOGWAGE ~ EDUC + ABILITY
## chisq = 1088.8, df = 1, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors

# Testing for Heteroskedasticity
bptest(LOGWAGE ~ EDUC + factor(PERSONID), data = PData)

## 
## studentized Breusch-Pagan test
## 
## data: LOGWAGE ~ EDUC + factor(PERSONID)
## BP = 4222.4, df = 2178, p-value < 2.2e-16

# Heteroskedasticity and Serial Correlation Consistent Estimator
coeftest(within_reg) # Original coefficients

## 
## t test of coefficients:
## 
##      Estimate Std. Error t value Pr(>|t|)    
## EDUC 0.1980043  0.0060133 32.928 < 2.2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(within_reg, vcovHC) # Heteroskedasticity consistent coefficients

## 
## t test of coefficients:
## 
##      Estimate Std. Error t value Pr(>|t|)    
## EDUC 0.198004   0.009866 20.069 < 2.2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```