Exercise: Analyzing Field Experiments

Following on from our simulation of omitted variable bias last week, we will now simulate an experimental analysis (you can adapt your script from last week or start again; the first 4 steps are the same):

1. Generate data on a population of 1,000 people. Specifically, create a variable (a vector) that randomly assigns these people to be male or female (50:50). *Hint: In R, try rbinom and in Stata, try rbinomial.*

```
N <- 1000
x <- rbinom(N,1,0.5)
```

2. Now we are going to simulate the potential outcomes - a measure of attitudes - *under control* (y_0) for our population. Create another variable of random normally-distributed values with mean of 5 and standard deviation of 1.

```
y0 <- rnorm(N,5,1)
```

3. One problem with observational data is that potential outcomes are often correlated with other variables such as gender. Adjust your value of y_0 to add 1 (one) for all units who are male.

```
y0 <- y0 + x
```

4. Now simulate potential outcomes *under treatment* (y_1) for all units. Define a *constant* treatment effect of $c = 2$ and create another vector $y_1 = y_0 + c$.

```
c <-2
y1 <- y0 + c
```

5. Next, let's assume a specific Treatment Assignment Mechanism – random assignment. Create a treatment variable D that gives each unit a 50% chance of binary treatment (like a coin flip).

```
data <- tibble(x,y0,y1) %>%
  mutate(D=rbinom(N,1,0.5))
```

6. Are gender and treatment correlated (they were strongly last week)? Calculate the correlation between x and D.

```
cor(data$x,data$D)
```

```
## [1] 0.005642042
```

7. In practice, we only observe one value: y_{obs}. Create a new variable y_{obs} which equals y_1 if $D = 1$ but which equals y_0 if $D = 0$.

```
data <- data %>% mutate(y_obs=case_when(D==1~y1,
                                        D==0~y0))
```

8. Compare the balance of observed pre-treatment covariates (gender) in the treatment and control groups using a difference-in-means test.

```
data %>% t.test(x ~ D, data=.)
```

```
##
##  Welch Two Sample t-test
##
## data:  x by D
## t = -0.17824, df = 996.38, p-value = 0.8586
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.06776096  0.05647645
## sample estimates:
## mean in group 0 mean in group 1
##       0.5061224       0.5117647
```

9. Compare the balance of observed potential outcomes (y_0, y_1) in the treatment and control groups using a difference-in-means test.

```
data %>% t.test(y0 ~ D, data=.)
```

```
##
##  Welch Two Sample t-test
##
## data:  y0 by D
## t = 0.4958, df = 997.63, p-value = 0.6201
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1038299  0.1740336
## sample estimates:
## mean in group 0 mean in group 1
##        5.551642        5.516540
```

```
data %>% t.test(y1 ~ D, data=.)
```

```
##
##  Welch Two Sample t-test
##
## data:  y1 by D
## t = 0.4958, df = 997.63, p-value = 0.6201
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1038299  0.1740336
## sample estimates:
## mean in group 0 mean in group 1
##        7.551642        7.516540
```

10. Based on the observable data, analyze the results of the experiment using a difference-in-means test of outcomes (y_{obs}) by treatment status (D). Interpret the result. Is this an accurate estimate of the treatment effect that we created at the start?

```
data %>% t.test(y_obs ~ D, data=.)
```

```
##
##  Welch Two Sample t-test
##
## data:  y_obs by D
## t = -27.753, df = 997.63, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.103830 -1.825966
## sample estimates:
## mean in group 0 mean in group 1
##        5.551642        7.516540
```

11. As an alternative, run the basic regression of observable outcomes (y_{obs}) on treatment (D). How does this compare to the difference-in-means test above?

```r
data %>% lm(y_obs ~ D, data=.) %>%
  stargazer(header=F, keep.stat=c("n"))
```

Table 1:

	Dependent variable:
	y_obs
D	1.965***
	(0.071)
Constant	5.552***
	(0.051)
Observations	1,000
Note:	*p<0.1; **p<0.05; ***p<0.01

12. Now add the control variable x to the regression. How do the results change?

```r
data %>% lm(y_obs ~ D + x, data=.) %>%
  stargazer(header=F, keep.stat=c("n"))
```

Table 2:

	Dependent variable:
	y_obs
D	1.959***
	(0.064)
x	0.991***
	(0.064)
Constant	5.050***
	(0.056)
Observations	1,000
Note:	*p<0.1; **p<0.05; ***p<0.01

13. Now let’s try to understand how things change when treatment is clustered:

    a. In one of your datasets, add an additional column which transforms (‘bins’) the y_0 variable into a categorical variable with 20 categories, where category ‘1’ represents the lowest values of y_0 and category ’20’ contains the highest values of y_0. This reflects the fact that people in the same cluster have more similar characteristics.

```r
data <- data %>% mutate(cluster=as.factor(ntile(y0,20)))
```

    b. Add a new binary treatment variable, *D_cluster*, that randomly assigns each cluster to treatment or control (so treatment is at the level of the cluster).

```r
cluster_treat <- tibble(cluster=factor(1:20),
                        D_cluster=rbinom(20,1,0.5))
```

	Model 1
(Intercept)	5.64^{***}
	(0.05)
D_cluster	1.78^{***}
	(0.07)
R^2	0.39
Adj. R^2	0.39
Num. obs.	1000
RMSE	1.12

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 3: Statistical models

	Model 1
(Intercept)	5.64^{***}
	(0.43)
D_cluster	1.78^{**}
	(0.52)
R^2	0.39
Adj. R^2	0.39
Num. obs.	1000
RMSE	1.12

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 4: Statistical models

```
data <- data %>% left_join(cluster_treat,by="cluster")
```

c. Recalculate y_{obs} based on the new treatment variable, $D\_cluster$.

```
data <- data %>% mutate(y_obs_cluster=case_when(D_cluster==1~y1,
                                      D_cluster==0~y0))
```

d. Now run the regression of observable outcomes (y_{obs}) on clustered treatment ($D\_cluster$). What is the confidence interval around the size of the effect $D\_cluster$? Does this makes sense?

```
data %>% lm(y_obs_cluster ~ D_cluster, data=.) %>%
  texreg(include.ci=F)
```

e. Now run the regression of observable outcomes (yobs) on clustered treatment (D_cluster), and cluster standard errors at the cluster level. How does the confidence interval on $D\_cluster$ differ from the previous analysis without clustered standard errors?

```
data %>% lm_robust(y_obs_cluster ~ D_cluster, data=., clusters=cluster) %>%
  texreg(include.ci=F)
```

14. Repeat the experiment and regression from Question 11 one hundred times with the same x, y_0 and y_1 every time, but re-randomize treatment assignment D each time. For each experiment, perform the basic regression and calculate the 95% confidence interval on the treatment variable. Finally, calculate how many of the 100 confidence intervals cover the real treatment effect ($c = 2$).

```
out <- list()
for (i in 1:100) {
  out[[i]] <- data %>% mutate(D=rbinom(N,1,0.5),
                         y_obs=case_when(D==1~y1,
                                      D==0~y0)) %>%
    lm(y_obs ~ D + x, data=.) %>%
    tidy() %>%
    filter(term=="D") %>%
    mutate(conf.lo=estimate-1.96*std.error,
           conf.hi=estimate+1.96*std.error) %>%
    select(conf.lo, conf.hi)
}

out %>% bind_rows() %>%
  mutate(covers_2=ifelse(conf.lo>2|conf.hi<2,0,1)) %>%
  tally(covers_2)
```

```
## # A tibble: 1 x 1
##        n
##    <dbl>
## 1    94
```

Stata Code for Qs 1-13

```
set obs 1000
gen x=rbinomial(1,0.5)
gen y0=rnormal(5,1)
replace y0=y0+x
gen y1=y0+2
gen D=rbinomial(1,0.5)
correlate x D
gen y_obs=y0
replace y_obs=y1 if D==1
ttest x, by(D)
ttest y0, by(D)
ttest y1, by(D)
ttest y_obs, by(D)
regress y_obs D
regress y_obs D x
egen cluster = cut(y0), group(20)
```

```
tempfile original_data
save       original_data'   keep cluster   duplicates drop   gen D_cluster=rbinomial(1,0.5)
merge 1:m cluster usingoriginal_data’, assert(match) nogenerate
regress y_obs D_cluster x
regress y_obs D_cluster x, vce(cluster cluster)
```