Randomized Experiments

Eduardo Fe

01/07/2019

Introduction.

For this exercise, we are going to use real data from Benjamin Olken's paper "Monitoring Corruption: Evidence from a Field Experiment in Indonesia", appeared in 2007 in the Journal of Political Economy. The paper and the data can be downloaded from Olken's website at MIT,

https://economics.mit.edu/faculty/bolken/published.

This paper presents a randomized field experiment on reducing corruption in over 600 Indonesian village road projects. The paper intends to gauge the effectiveness of direct top-down monitoring and increased grassroots participation of commmunities in monitoring. We will focus on estimating the effect of top-down monitoring only. In the experiment, the units of analysis are each of the 608 participating villages. In the active treatment (which is randomly allocated) villages are told, after funds for the road project have been awarded but before construction began, that their project would be audited by a central government agency. Indeed, audits commenced approximately seven months after construction was completed and data on the level of corruption in the construction project was gathered.

To measure corruption, Olken assembled a team of engineers and surveyors who, after the projects were completed, dug core samples in each road to estimate the quantity of materials used, surveyed local suppliers to estimate prices, and interviewered villagers to determine the wages paid on the project. From these data, Olken computed an estimate of the actual cost of the project. These estimate was compared with what the village leadership claimed the road cost to built. The magnitude of the discrepancy is the key measure of missing expenditures (corruption).

Before starting, let's load the required libraries,

```
library(haven)
library(tidyverse)
```

The main data file is `jperoaddata`. Prior to load this, it is good practice to see if the randomization worked. To do this, we can check if there is any correlation between the treatment indicator and some pre-treatment variables. Specifically, the file `jperandomizationdata.dta` contains data about a village's population, number of mosques, village budget in million of Rupees, percent of households in poverty, and others.

Let's load the data

```
datos <- read_dta("D:/teaching/methods@manchester/olken/jpepublic/jperandomizationdata.dta")
```

and see how many villages were allocated to the main treatment (audit),

```
datos %>% group_by(audit) %>% summarise(n=n())
```

```
## # A tibble: 2 x 2
##   audit     n
##   <dbl> <int>
## 1     0   324
## 2     1   282
```

To see if the randomization worked, we could run a regression of the treatment indicator (audit) and the available covariates. We will estimate clustered standard errors at the district level (for this we need the libraries `sandwich`, `lmtest` and `multiwayvcov`)

```
lmCheck <- lm(audit~zpop +totalmesjid+ totalallocation +
                  z4RABnumsubproj +zpercentpoorpra+ zdistancekec +
                  zkadesedyears+ zkadesage +zkadesbengkoktotal+
                  podeszhill, data =datos)

library(sandwich)
library(lmtest)
library(multiwayvcov)

coeftest(lmCheck, cluster.vcov(lmCheck, datos$kecnum))
```

```
##
## t test of coefficients:
##
##                       Estimate  Std. Error t value Pr(>|t|)
## (Intercept)         0.14108974  0.20475249  0.6891  0.49106
## zpop               -0.00673427  0.01186986 -0.5673  0.57071
## totalmesjid        -0.01701101  0.03720356 -0.4572  0.64767
## totalallocation    -0.00057238  0.00041206 -1.3891  0.16536
## z4RABnumsubproj    -0.01737681  0.02367359 -0.7340  0.46324
## zpercentpoorpra     0.23911867  0.12270117  1.9488  0.05181 .
## zdistancekec       -0.00134361  0.00531041 -0.2530  0.80035
## zkadesedyears       0.01132065  0.00867283  1.3053  0.19232
## zkadesage           0.00387916  0.00250324  1.5497  0.12178
## zkadesbengkoktotal  0.01100320  0.00623443  1.7649  0.07812 .
## podeszhill          0.13047371  0.07271134  1.7944  0.07328 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Most variables are not significant, as expected, but a few (village poverty rate, village head salary and the indicator of mountanious area) are significant at 10% level. Both poverty rate and village head salary could be predictors of corruption, which might be problematic for identification. Olken runs a test of the joint significance of all the variables to further evaluate if these latter results are problematic; he finds that jointly, the regressors do not predict the allocation in treatment.

The actual data for the evaluation in the experiment is in the file `jperoaddata.dta`, which we can load in a second data frame,

```
datos2 <- read_dta("D:/teaching/methods@manchester/olken/jpepublic/jperoaddata.dta")
```

This further shows one of the advantages of R (the ability to work with two or more datasets simultaneously). The key variables for the evaluation are `lndiffe...` which contain the difference between what villages claim they spent on the project and an independent estimate of what villages actually spent. Olken's paper gives details of the what these independent estimates were obtained. Let's start by summarising the data, which should confirm the result sin Table 3 in Olken's paper.

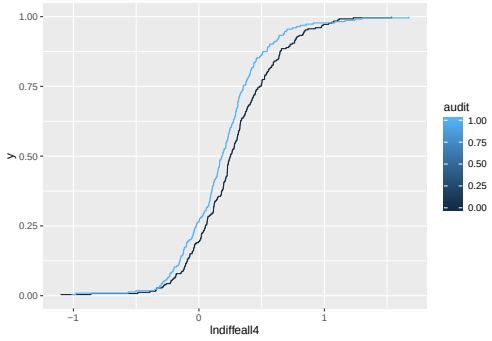
```
datos2 %>% summary()
```

Here you can see the average project size, the share of total reported expenses (the road project itself, ancillary projects or other projects), the share of project expenses due to different building materials, as well as the percentage of missing money in each category of project. For example, looking at the percentage missing in major items in road project (`lndiffeall4`), we learn the average percetange missing (reported expenditure exceeded the independent estimate) was 24 percent!

As a next step, we might want to compare the difference in expenditure by treatment/control groups using a graph. We can start by looking at the empirical distribution function of the difference in expenditure (between reported and estimated) by control/treatment group. To do this, we need `ggplot2`. To plot an

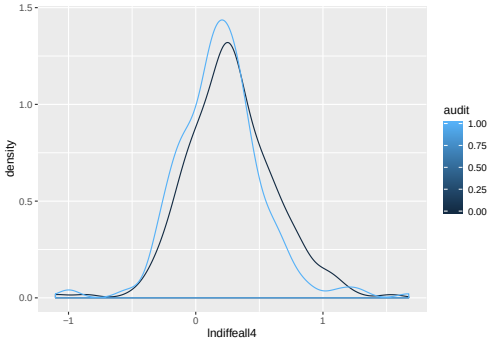
empirical cumulative distribution function, we need the *geometry* `stat_ecdf()`, and to produce a graph by group, we need to add `aes(group= varname)`, where `varname` specifies which variable defines each group (in our case, the treatment and control groups). We focus on the difference in expenditure in the major items in the road project,

```
library(ggplot2)
ggplot(datos2, aes(lndiffeall4) ) + stat_ecdf(pad= FALSE, aes(group= audit, color = audit))
```



In the graph, the blue line is the empirical distribution function of the treatment group. We first note that some villages reported to have spent less than the independent estimates suggested (about 25 per cent of the villages). Most villages reported expenditure that exceeded the independently estimated cost. Second, we note that the distribution of the control group is dominated (below) the distribution of the treatment group. This means that the reduction in missing expenditure induced by the treatment happened at all levels of the distribution (so the intervention seems to have affected all the villages). An alternative way to looking at this question is by looking at the smoothed histogram (density estimator) of the same variable by treatment and control group.

```
library(ggplot2)
ggplot(datos2, aes(lndiffeall4) ) + geom_density(pad= FALSE, aes(group= audit, color = audit))
```



We see that the percentage missing expenditure concentrated more around 0 in the treatment group and that the proportion of villages reporting very high levels of missing expenditure decreased considerably.

We can progress towards a more formal comparison of average percentage missing expenditure, by using a regression model. Let's start with a simple regression involving missing expenditure in major items in the road project and the treatment indicator. Given the randomization of the treatment, the ensuing estimate will have a causal interpretation,

```
lmModel1 <- lm(lndiffeall4~audit, data =datos2)


coeftest(lmModel1, cluster.vcov(lmModel1, datos2$kecnum))
```

```
##
```

```
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.276755   0.033348  8.2991 1.096e-15 ***
## audit       -0.085041   0.044150 -1.9262   0.05468 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, it seems like the intervention reduced missing expenditure in major project items, though the estimate is only significant at 10 per cent. Moving on, we can try to refine the estimates in a couple of ways. First, the engineers that assessed each project might differn in their assessments due to issues such as their education, their thoroughness, etc. It would thus be important to take this variation into account in the model. To do this, we can introduce a collection of indicators (fixed effects) identifying the engineer. We can ask `R` to automatically create these fixed effects for the regression using the option `factor(varname)`, where `varname` is the variable that identifies each engineer in the data. Therefore, the resulting regression is

```
lmModel2 <- lm(lndiffeall4~audit+factor(z7enumcode), data =datos2)
coeftest(lmModel2, cluster.vcov(lmModel1, datos2$kecnum))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.296032   0.033348  8.8771  < 2e-16 ***
## audit       -0.076345   0.044150 -1.7292  0.08444 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the fixed effects are not shown (but they were used in estimation). The effect of the treatment is still significant, though only at 10 per cent level. The value of the treatment effect seems, however, to be reduced by 1 percentage point. Finally, as a last check, we re-estimate the model using stratum fixed effects

```
lmModel2 <- lm(lndiffeall4~audit+factor(auditstratnum), data =datos2)
coeftest(lmModel2, cluster.vcov(lmModel1, datos2$kecnum))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.156303   0.033348  4.6871 3.735e-06 ***
## audit       -0.048462   0.044150 -1.0977     0.273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You can now continue to explore the data a bit further, and try to replicate some of the additional results in Olken's paper...