

Diferencias en Diferencias

Maestría en Economía Uninorte

Carlos Andrés Yanes

2024-08-24

Introducción

Seguimos trabajando el *ejemplo* de Nessa, Ali, and Abdul-Hakim (2012) para el análisis del impacto Programa de Microcréditos realizado en Bangladesh para 1998.

El método de **Diferencias en Diferencias (DID)** es una técnica econométrica ampliamente utilizada para evaluar el **impacto causal** de políticas públicas, programas o intervenciones que se implementan en ciertas unidades (como individuos, empresas, regiones, etc.) pero no en otras. Este método se basa en comparar la evolución de un resultado de interés entre un grupo tratado (que recibe la intervención) y un grupo de control (que no la recibe) *antes* y *después* del tratamiento.

El enfoque DID asume que, en ausencia del tratamiento, la diferencia en las tendencias del resultado entre los grupos tratado y de control habría permanecido constante. De esta manera, cualquier desviación de esta tendencia paralela se atribuye al efecto causal del tratamiento.

En economía, el método de Diferencias en Diferencias es particularmente útil cuando los experimentos aleatorios no son factibles o éticos, y permite controlar por factores no observados que podrían influir en los resultados, siempre que estos factores sean constantes en el tiempo. Esta técnica ha sido utilizada en estudios que analizan desde el impacto de cambios en las políticas fiscales y laborales hasta los efectos de programas

educativos y de salud, convirtiéndose en una herramienta clave para el análisis de políticas públicas.

Preparación

Antes de implementar el código de estimación, es crucial preparar la base de datos asegurando que las variables relevantes estén correctamente definidas y limpiadas. Esto implica verificar que las variables de tratamiento y resultado estén codificadas adecuadamente, que las covariables no presenten valores faltantes, y que los pesos muestrales, si son aplicables, estén correctamente asignados. Además, es esencial que los datos estén en el formato adecuado para ser utilizados en los modelos estadísticos, lo que incluye transformar variables según sea necesario y asegurarse de que todas las observaciones relevantes sean incluidas en el análisis.

Estipulación de la base

Vamos a cargar los paquetes a utilizar en esta ocasión

```
library(pacman)
p_load(survey, foreign, haven, tidyverse, huxtable, dplyr, Matching, ggplot2, plm, car)
```

El paso a seguir es cargar la base de datos (formato stata) a R.

```
datacon9198 <- read.dta("hh_9198.dta")
```

Transformación y adecuación de la data

Vamos a simular para tratar de acondicionar los datos de los microcreditos

```
# Creamos los ajustes
datacon9198 <- mutate(datacon9198, exptot0=ifelse(year == 0, exptot, 0))
datacon9198 <- group_by(datacon9198,nh) %>%
  mutate(exptot91 = max(exptot0))
datacon9198 <- subset(datacon9198, year == 1)
datacon9198 <- mutate(datacon9198, lexptot91=ifelse(year == 1, log(1+exptot91), 0))
datacon9198 <- mutate(datacon9198, lexptot98=ifelse(year == 1, log(1+exptot), 0))
datacon9198 <- mutate(datacon9198, lexptot9891 = lexptot98-lexptot91)
datacon9198 <- ungroup(datacon9198)
```

Para grupo Masculino y femenino

Vamos a montar una data de tal manera que jugamos con las variables para que por grupos tengamos una dummy para $T = 1$ periodo posterior y $T = 0$ como el anterior

```
datacon9198 <- mutate(datacon9198, lexptot = log(1 + exptot))
datacon9198 <- mutate(datacon9198, lnland = log(1 + hhland / 100))
datacon9198 <- mutate(datacon9198, dmmfd1=ifelse(dmmfd == 1 & year == 1, 1, 0))
datacon9198 <- group_by(datacon9198,nh) %>%
  mutate(dmmfd98 = max(dmmfd1))
datacon9198 <- mutate(datacon9198, dfmfd1=ifelse(dfmfd == 1 & year == 1, 1, 0))
datacon9198 <- group_by(datacon9198,nh) %>%
  mutate(dfmfd98 = max(dfmfd1))
datacon9198 <- mutate(datacon9198, dmmfdyr = dmmfd98*year)
datacon9198 <- mutate(datacon9198, dfmfdyr = dfmfd98*year)
datacon9198 <- ungroup(datacon9198)
head(datacon9198)
```

Ya con la estructura de datos preparada vamos a los modelos a estimar.

```
basico <- lm(lexptot ~ year + dfmfd98 + dfmfdyr, data = datacon9198)
huxreg(basico)
```

df	weight	vaccess	pcirr	rice	wheat	milk	potato	egg	oil	exptot0	exptot91	lexptot
1	1.04	1	0.45	12.6	8.12	11.5	8.55	2.2	40.6	0	3.96e+03	
1	1.01	1	0.45	12.6	8.12	11.5	8.55	2.2	40.6	0	3.45e+03	
1	1.01	1	0.45	12.6	8.12	11.5	8.55	2.2	40.6	0	3.69e+03	
1	1.01	1	0.45	12.6	8.12	11.5	8.55	2.2	40.6	0	2.37e+03	
1	0.763	1	0.2	10.2	6.09	10.8	6.87	2.03	43.3	0	4.19e+03	
0	2.25	1	0.2	10.2	6.09	10.8	6.87	2.03	43.3	0	5.24e+03	

Basic Model with FE on nh

```
lm <- lm(lexptot ~ year + dfmfdyr + dfmfd98 + factor(nh),
data = datacon9198) summary(lm)
```

```
#Check for multicollinearity
```

```
sqrt(vif(lm)) #Error in vif.default(lm) : there are aliased coef-
ficients in the model
```

```
# Contains multicollinearity check <- alias(lm) # Notice that
dfmfd98 = -1, therefore highly correlated with dfmfdyr
```

```
# Remove dfmfd98 lm <- lm(lexptot ~ year + dfmfdyr + fac-
tor(nh), data = datacon9198) sqrt(vif(lm))
```

```
# Output is fine now, so can proceed
```

```
# GVIF Df GVIF^(1/(2*Df)) # year 1.453455 1.00000
1.205593 # dfmfdyr 1.764237 1.00000 1.328246 # factor(nh)
1.414214 28.72281 1.000210
```

```
# Second method for testing for multicollinearity kappa() test
<- model.matrix(~ year + dfmfdyr + dfmfd98 + factor(nh),
data = datacon9198) kappa(test) # Output : 2.017073e+16
```

```
##### Because of an extra large kappa, there is collinearity in
our model and should be dealt with ##### conditional number
must be less than 30
```

```
summary(lm)
```

	(1)
(Intercept)	8.458 *** (0.027)
year	
dfmfd98	-0.003 (0.037)
dfmfdyr	
N	826
R2	0.000
logLik	-639.678
AIC	1285.355

*** p < 0.001; ** p < 0.01; * p < 0.05.

Using plm for fixed-effect

```
lm <- plm(lexptot ~ year + dfmfdyr + dfmfd98 + nh, data =
datacon9198, model = "within", index = "nh") summary(lm)
```

PSM with DD

Data setup

```
datacon9198 <- read.csv("Data/hh_9198.csv") datacon9198
<- mutate(datacon9198, lnland = log(1 + hhland / 100)) data-
con9198 <- mutate(datacon9198, dfmfd1=ifelse(dfmfd == 1 &
year == 1, 1, 0)) datacon9198 <- group_by(datacon9198,nh)
%>% mutate(dfmfd98 = max(dfmfd1)) datacon9198
```

```
<- filter(datacon9198, year == 0) datacon9198$X <-
1:nrow(datacon9198)
```

First Regression (Unbalanced)

```
des1 <- svydesign(id = ~X, weights = ~weight, data = data-
con9198) prog.lm <- svyglm(dfmfd98 ~ sexhead + agehead +
educhead + lnland + vaccess + pcirr + rice + wheat + milk +
oil, design=des1, family = quasibinomial(link = "probit"))
```

```
X <- prog.lm$fittedTr <- -datacon9198dfmfd
```

```
m.out <- Match(Tr = Tr, X = X, caliper = 0.01) sum-
mary(m.out)
```

```
MatchBalance(dfmfd98 ~ sexhead + agehead + educhead +
lnland + vaccess + pcirr + rice + wheat + milk + oil, data =
datacon9198, nboots = 1000)
```

```
#Graph density of propensity scores fit <- prog.lmdatafitfvalues
<- prog.lm$fitted.values
```

```
fit.control <- filter(fit, dfmfd == 0) fit.treated <- filter(fit,
dfmfd == 1)
```

```
ggplot() + geom_density(aes(x=fit.controlfvalues, linetype = '
2')) + geom_density(aes(x = fit.treatedfvalues, linetype =
'3')) + xlim(-.3,1) + xlab("") + scale_linetype_discrete(name
="", labels = c("Control", "Treated")) + ggtitle("Control and
Treated Densities")
```

Build data frame with ps and nh, then drop ps not matched

```
ps_dropped <- m.outindex.droppedps_datacon9198 <-
-data.frame(psm = prog.lmfitted.values) ps_datacon9198nh <-
-prog.lmdata$nh ps_datacon9198 <- ps_datacon9198[-
ps_dropped,] rownames(ps_datacon9198) <- NULL
```

```
#Merge to original data frame by nh datacon9198
<- read.csv("Data/hh_9198.csv") psm_datacon9198 <-
right_join(datacon9198, ps_datacon9198, by = "nh")
```

Re-estimate baseline model with matched data set

```
psm_datacon9198 <- mutate(psm_datacon9198, lex-
ptot = log(1 + exptot)) psm_datacon9198 <- mu-
tate(psm_datacon9198, lnland = log(1 + hhland / 100))
psm_datacon9198 <- mutate(psm_datacon9198, dfmfd1=ifelse(dfmfd
== 1 & year == 1, 1, 0)) psm_datacon9198 <- group_by(psm_datacon9198,nh)
%>% mutate(dfmfd98 = max(dfmfd1)) psm_datacon9198
<- mutate(psm_datacon9198, dfmfdyr = dfmfd98*year)
psm_datacon9198 <- ungroup(psm_datacon9198)
```

Re-estimate Basic Model

```
lm <- lm(lexptot ~ year + dfmfd98 + dfmfdyr, data =
psm_datacon9198) summary(lm)
```

Create Analytical Weights

```
psm_datacon9198a_weight <- -1psm_datacon9198a_weight <-
ifelse(psm_datacon9198dfmfd == 0,psm_datacon9198psm/(1-
psm_datacon9198$psm), 1)
```

Re-estimate with analytical weights

```
lm <- lm(lexptot ~ year + dfmfd98 + dfmfdyr, data =
psm_datacon9198, weights = a_weight) summary(lm)
```

End of Script