

Monitoria 16 - RDD

Ramon Bertotto

2024-11-22

Uma literatura que se beneficiou muito do RDD é a de economia política, se tratando do design de eleições acirradas. Basicamente um candidato ganha a votação se ele tiver uma porcentagem mínima dos votos. Dessa forma é possível usar eleições para avaliar uma variedade de resultados. Vamos replicar alguns resultados de Lee, Moretti e Butler (2004): Do Politicians or Voters Pick Policies?

Existem duas principais visões sobre o papel das eleições numa democracia representativa: teoria da convergência e da divergência. O teorema da convergência diz que a ideologia do voto heterogêneo força cada candidato a moderar a sua posição. Já a teoria da divergência tem uma visão mais de senso comum dos políticos. Quando um partido político não consegue se comprometer para alguma política a convergência é prejudicada e o resultado final é a divergência, caracterizando-a: depois de ser eleito, o político/partido tenta implementar a sua política preferida. Nesse caso extremo quem vota não é capaz de forçar o candidato eleito em se comprometer com uma política, o que é expresso em dois candidados opostos escolhendo políticas muito diferentes no cenário de vitória de um deles.

Vamos usar dois data-sets pra rodar esse exercício aqui, o primeiro é uma medida de quão liberal é um político votado. Esse índice vem da Americans for Democratic Action (ADA) ligado a eleição do congresso para eleições de 46 a 95. Se usa esse índice como um medida de voto dos candidatos. Para cada congresso eleito, a ADA escolhe 25 votos de perfil roll-call e cria um índice que vai de 0 a 100, onde 100 é mais liberal possível.

A running variable é share de votos que foi pra um democrata, e se argumenta que as vitórias dos democratas são exógenas para checar se vale divergência ou convergência. Se vale convergência republicanos e democratas que ganharam em disputas acirradas devem votar identicamente, enquanto se vale divergência eles devem votar de forma diferente. A ideia é que na margem de uma eleição acirrada a distribuição de preferência dos eleitores é igual, e se essas preferências são iguais mas as políticas implementadas diferem no cutoff, isso é evidência sugetiva de que os eleitos escolhem as diretrizes políticas.

O cutoff aqui é um share de votos de 0,5 para que o partido democrata vença, e se argumenta que ao redor desse valor a eleição dos democratas é aleatória, e por isso que D_t é atribuído aleatoriamente. Se reproduz o trabalho de Lee, Moretti e Butler. Veja que estamos pegando um subset de eleições onde os democratas ficaram com share de votos entre 48 e 52, ou seja, eleições que foram bem acirradas.

Replicação

Primeiro, vamos instalar as bibliotecas que nós vamos usar

```
#install.packages("tidyverse")
#install.packages("haven")
#install.packages("estimatr")
#install.packages("modelsummary")
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'tibble' was built under R version 4.3.3
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
## Warning: package 'purrr' was built under R version 4.3.3
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
## Warning: package 'stringr' was built under R version 4.3.3
```

```
## Warning: package 'forcats' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2    3.5.1      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.1
## ✓ purrr      1.0.2
## — Conflicts ————— tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(haven)
```

```
## Warning: package 'haven' was built under R version 4.3.3
```

```
library(estimatr)
```

```
## Warning: package 'estimatr' was built under R version 4.3.3
```

```
library(modelsummary)
```

```
## Warning: package 'modelsummary' was built under R version 4.3.3
```

```
## `modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing
## backend. Learn more at: https://vincentarelbundock.github.io/tinytable/
##
## Revert to `kableExtra` for one session:
##
## options(modelsummary_factory_default = 'kableExtra')
## options(modelsummary_factory_latex = 'kableExtra')
## options(modelsummary_factory_html = 'kableExtra')
##
## Silence this message forever:
##
## config_modelsummary(startup_message = FALSE)
```

Você pode puxar a base de dados de duas diferentes formas: você pode rodar um código e puxar ela do github do Scott Cunningham, ou você pode baixar ela lá do site e abrir ela, abaixo segue o código

```
lmb_data <- read_dta("C:/Users/15534507/Downloads/lmb-data.dta")
```

Aqui se você baixar o arquivo e quiser importar ele manualmente pelo R. Se lembre que você precisa mudar o caminho da pasta.

Agora, a gente vai filtrar a base de dados em uma banda próxima ao cutoff, isto é, eleições nas quais os democratas ficaram com votos entre 48% e 52%:

```
lmb_subset <- lmb_data %>%
  filter(lagdemvoteshare>.48 & lagdemvoteshare<.52)
```

Por fim, vamos rodar as regressões, se roda como um OLS simples, não tem segredo nenhum. Veja que as variáveis democrat e lag-democrat são variáveis binárias, enquanto Score é uma variável “contínua” que vai de 0 a 100. A gente vai rodar com o `lm_robust` porque ele permite diretamente usar os erros padrões clusterizados. Depois se usa o `model-summary` pra poder comparar os resultados

```
lm_1 <- lm_robust(score ~ lagdemocrat, data = lmb_subset, clusters = id)
lm_2 <- lm_robust(score ~ democrat, data = lmb_subset, clusters = id)
lm_3 <- lm_robust(democrat ~ lagdemocrat, data = lmb_subset, clusters = id)
modelsummary(list(Score=lm_1,Score=lm_2,Democrat=lm_3))
```

	Score	Score	Democrat
(Intercept)	31.196	18.747	0.242
	(1.334)	(0.843)	(0.020)
lagdemocrat	21.284		0.484
	(1.951)		(0.029)
democrat		47.706	
		(1.356)	
Num.Obs.	915	915	915

	Score	Score	Democrat
R2	0.115	0.578	0.235
R2 Adj.	0.114	0.578	0.234
AIC	8795.5	8117.3	1088.6
BIC	8809.9	8131.8	1103.0
RMSE	29.49	20.36	0.44
Std.Errors	by: id	by: id	by: id

A primeira regressão basicamente nos diz que o efeito de uma vitória democrática aumenta os votos em políticas liberais em 21 pontos no próximo período, enquanto no período corrente esse efeito é de 47.7. A última coluna nos diz que a vitória democrática na eleição anterior aumenta a chance de reeleição em 48%. Isso é evidência de divergência e vantagem dos incumbentes. Quando se menciona regressão local é porque restringimos nossa amostra pra um ponto próximo ao cut-off. Se rodamos a regressão com todos os dados obtemos resultados bem diferentes:

Rodando se nós usamos todos os dados:

```
#using all data (note data used is lmb_data, not lmb_subset)
lm_4 <- lm_robust(score ~ lagdemocrat, data = lmb_data, clusters = id)
lm_5 <- lm_robust(score ~ democrat, data = lmb_data, clusters = id)
lm_6 <- lm_robust(democrat ~ lagdemocrat, data = lmb_data, clusters = id)
modelsummary(list(Score=lm_4,Score=lm_5,Democrat=lm_6))
```

	Score	Score	Democrat
(Intercept)	23.539	17.576	0.120
	(0.337)	(0.263)	(0.004)
lagdemocrat	31.506		0.818
	(0.484)		(0.005)
democrat		40.763	
		(0.418)	
Num.Obs.	13588	13588	13588
R2	0.227	0.376	0.676
R2 Adj.	0.227	0.376	0.676
AIC	129786.7	126881.1	3902.1
BIC	129809.3	126903.7	3924.7
RMSE	28.69	25.78	0.28

	Score	Score	Democrat
Std.Errors	by: id	by: id	by: id

Todos os coeficientes mudam: o efeito da eleição no período passado aumenta os votos em políticas liberais no período seguinte, enquanto o efeito no período atual diminuiu. O efeito da eleição na reeleição também aumentou consideravelmente.

Agora vamos adicionar alguns controles e recentralizar a variável de eleição

```
lmb_data <- lmb_data %>%
  mutate(demvoteshare_c = demvoteshare - 0.5)

lm_7 <- lm_robust(score ~ lagdemocrat + demvoteshare_c, data = lmb_data, clusters = id)
lm_8 <- lm_robust(score ~ democrat + demvoteshare_c, data = lmb_data, clusters = id)
lm_9 <- lm_robust(democrat ~ lagdemocrat + demvoteshare_c, data = lmb_data, clusters = id)
modelsummary(list(Score=lm_7,Score=lm_8,Democrat=lm_9))
```

	Score	Score	Democrat
(Intercept)	22.883	11.034	0.212
	(0.443)	(0.336)	(0.005)
lagdemocrat	33.451		0.552
	(0.848)		(0.010)
demvoteshare_c	-5.626	-48.938	0.773
	(1.898)	(1.642)	(0.019)
democrat		58.502	
		(0.656)	
Num.Obs.	13577	13577	13577
R2	0.227	0.424	0.735
R2 Adj.	0.227	0.424	0.735
AIC	129674.5	125683.0	1153.2
BIC	129704.5	125713.1	1183.3
RMSE	28.68	24.76	0.25
Std.Errors	by: id	by: id	by: id

Fazendo isso, o efeito da eleição passada no índice de votos liberais é de 33 pontos, o da eleição corrente é de 58 pontos, e o efeito da eleição passada na corrente é de 55%.

Outro procedimento que é comum é deixarmos ter uma linha de regressão para cada lado do cut-off (0,5). Pra fazer isso, basta interagir a running variable com a variável de tratamento:

```
lm_10 <- lm_robust(score ~ lagdemocrat*demvoteshare_c,
                  data = lmb_data, clusters = id)
lm_11 <- lm_robust(score ~ democrat*demvoteshare_c,
                  data = lmb_data, clusters = id)
lm_12 <- lm_robust(democrat ~ lagdemocrat*demvoteshare_c,
                  data = lmb_data, clusters = id)

modelsummary(list(Score=lm_10,Score=lm_11,Democrat=lm_12))
```

	Score	Score	Democrat
(Intercept)	31.435	16.816	0.287
	(0.541)	(0.419)	(0.008)
lagdemocrat	30.508		0.526
	(0.817)		(0.010)
demvoteshare_c	66.042	-5.683	1.403
	(3.161)	(2.611)	(0.044)
lagdemocrat × demvoteshare_c	-96.475		-0.849
	(3.853)		(0.049)
democrat		55.431	
		(0.637)	
democrat × demvoteshare_c		-55.152	
		(3.219)	
Num.Obs.	13577	13577	13577
R2	0.267	0.434	0.749
R2 Adj.	0.267	0.434	0.749
AIC	128963.7	125441.1	443.1
BIC	129001.3	125478.7	480.7
RMSE	27.94	24.54	0.25
Std.Errors	by: id	by: id	by: id

Os modelos reportados são regressões globais. Elas nos dizem que o efeito da vitória democrática em $t - 1$ no score de votos liberais é de 30,5, enquanto o efeito da vitória democrática em t é de 55,43. A vitória democrática em $t - 1$ aumenta em 53% a chance de reeleição democrática.

Agora, vamos inserir um termo quadrático pra capturar possíveis não linearidades no nosso modelo. Pode-ser que o efeito de tratamento esteja capturando essa não-linearidade:

```
lmb_data <- lmb_data %>%
  mutate(demvoteshare_sq = demvoteshare_c^2)

lm_13 <- lm_robust(score ~ lagdemocrat*demvoteshare_c + lagdemocrat*demvoteshare_sq,
  data = lmb_data, clusters = id)
lm_14 <- lm_robust(score ~ democrat*demvoteshare_c + democrat*demvoteshare_sq,
  data = lmb_data, clusters = id)
lm_15 <- lm_robust(democrat ~ lagdemocrat*demvoteshare_c + lagdemocrat*demvoteshare_sq,
  data = lmb_data, clusters = id)

modelsummary(list(Score=lm_13,Score=lm_14,Democrat=lm_15))
```

	Score	Score	Democrat
(Intercept)	33.547	15.606	0.330
	(0.714)	(0.575)	(0.013)
lagdemocrat	13.030		0.322
	(1.286)		(0.018)
demvoteshare_c	134.977	-23.850	2.798
	(9.786)	(6.713)	(0.196)
demvoteshare_sq	212.127	-41.729	4.294
	(22.763)	(14.686)	(0.456)
lagdemocrat × demvoteshare_c	57.055		0.091
	(15.412)		(0.241)
lagdemocrat × demvoteshare_sq	-641.851		-8.804
	(31.331)		(0.517)
democrat		44.402	
		(0.909)	
democrat × demvoteshare_c		111.896	
		(9.781)	
democrat × demvoteshare_sq		-229.954	

	Score	Score	Democrat
	(19.546)		
Num.Obs.	13577	13577	13577
R2	0.371	0.456	0.822
R2 Adj.	0.370	0.456	0.822
AIC	126894.8	124919.4	-4226.4
BIC	126947.4	124972.0	-4173.8
RMSE	25.89	24.07	0.21
Std.Errors	by: id	by: id	by: id

Agora, os efeitos caem consideravelmente. Vamos nos lembrar que estamos estimando regressões globais, e por isso que as estimativas podem ser maiores. Geralmente, isso é porque existem alguns outliers nos dados que fazem a distância até c_0 se espalhar mais amplamente. Vamos então fazer uma regressão local em até 5 p.p. do cutoff:

```
lmb_data <- lmb_data %>%
  filter(demvoteshare > .45 & demvoteshare < .55) %>%
  mutate(demvoteshare_sq = demvoteshare_c^2)

lm_16 <- lm_robust(score ~ lagdemocrat*demvoteshare_c + lagdemocrat*demvoteshare_sq,
  data = lmb_data, clusters = id)
lm_17 <- lm_robust(score ~ democrat*demvoteshare_c + democrat*demvoteshare_sq,
  data = lmb_data, clusters = id)
lm_18 <- lm_robust(democrat ~ lagdemocrat*demvoteshare_c + lagdemocrat*demvoteshare_sq,
  data = lmb_data, clusters = id)
modelsummary(list(Score=lm_16,Score=lm_17,Democrat=lm_18))
```

	Score	Score	Democrat
(Intercept)	37.121	21.437	0.418
	(0.969)	(1.819)	(0.013)
lagdemocrat	7.347		0.167
	(1.587)		(0.020)
demvoteshare_c	830.925	450.846	15.699
	(20.956)	(161.352)	(0.228)
demvoteshare_sq	5333.335	7878.904	91.607
	(838.325)	(2995.192)	(10.893)
lagdemocrat × demvoteshare_c	-156.876		0.125

	Score	Score	Democrat
	(35.740)		(0.357)
lagdemocrat × demvoteshare_sq	-10116.678		-188.329
	(1435.130)		(16.351)
democrat		45.191	
		(2.679)	
democrat × demvoteshare_c		-688.343	
		(247.711)	
democrat × demvoteshare_sq		-3887.820	
		(4802.371)	
Num.Obs.	2387	2387	2387
R2	0.445	0.563	0.774
R2 Adj.	0.444	0.562	0.774
AIC	21860.2	21290.4	-74.4
BIC	21900.6	21330.9	-34.0
RMSE	23.50	20.86	0.24
Std.Errors	by: id	by: id	by: id

Estamos usando mais observações do que quando rodamos a regressão local anterior. Pra recapitular o que fizemos: adicionamos a running variable, e incluímos ela de várias formas.