

# Regresión Discontinua

Maestría en Economía Uninorte

Carlos Andrés Yanes

2024-08-30

## Preambulo

### Introducción a la Regresión Discontinua

La regresión discontinua es un método cuasi-experimental utilizado para identificar efectos causales de un tratamiento. Se basa en cortes (*cutoff*) que surgen por ley o por diseño y que implican una discontinuidad en la implementación del tratamiento, definido a lo largo de alguna variable llamada la “variable de corte”.

El método funciona al comparar observaciones que se extienden estrechamente a ambos lados del umbral o punto de corte, permitiendo estimar el efecto promedio del tratamiento en entornos donde la aleatorización era inviable. La intuición detrás de la regresión discontinua se ilustra bien con la evaluación de becas basadas en mérito, donde el principal problema es la endogeneidad de la asignación.

Existen dos tipos principales de regresión discontinua:

- **Sharp**: La probabilidad de recibir el tratamiento pasa de 0 a 1 en la discontinuidad.
- **Fuzzy**: La probabilidad de recibir el tratamiento cambia abruptamente en la discontinuidad, pero no pasa de 0 a 1 debido a la posibilidad de “always-takers” y “never-takers”.

La regresión discontinua se ha vuelto cada vez más popular en los últimos años para evaluar efectos causales de intervenciones en diversas disciplinas como estadística, econometría, ciencia política y epidemiología. Es un método útil para la evaluación de políticas públicas, permitiendo analizar sus efectos a corto, mediano y largo plazo.

 **Clave!**

La regresión discontinua tiene mayor desarrollo. Este material es meramente académico e introductorio ha sido brindado por el Profesor Colin Cameron de la Universidad de Duke. Todos los créditos son para él.

## Datos

- Datos originales `rdsenate.dta`
- Son propiedad de Sebastian Calonico, Matias Cattaneo, Max Farrell, and Rocco Titiunik (2017),
- El paper se denomina “Rdrobust: Software for Regression-discontinuity Designs,” *The Stata Journal*, 17(2), pages 372-404.

## Limpiando el Environment de R

Siempre es bueno limpiar el entorno de R. Esto se hace con el objeto de no cometer fallos o errores por uso de una data adicional que no se requiera por el momento

```
rm(list = ls())
```

## Preparación del entorno para ejecución

Vamos a cargar un par de paquetes de una vez con la opción de `pacman()` y añadiremos `huxtable` que sirve para darle estilo a las salidas de las regresiones inclusive a html

```
library(pacman)
p_load(lmtest, foreign, haven, tidyverse, stargazer, dplyr, estimatr, ggplot2, sandwich, huxtable)
```

## Cargar la base de datos

La aplicación es para las elecciones al Senado de EE.UU. desde 1914 hasta 2010. La variable de asignación es el margen de victoria del Partido Demócrata en un escaño del Senado en el año  $t$ , y la variable de resultado es la proporción de votos del Partido Demócrata en la elección subsecuente para el mismo escaño, una elección que generalmente ocurre en el año  $t + 2$ . La hipótesis en consideración es que existe una ventaja del incumbente, por lo que una victoria (derrota) ajustada en una elección probablemente conduzca a una victoria (derrota) en la elección subsecuente.

```
base.sen <- read_dta("incumbency.dta")
head(base.sen)
```

| year     | vote | margin | class | termshouse | termssenate | population | win |
|----------|------|--------|-------|------------|-------------|------------|-----|
| 1.91e+03 | 36.1 | -7.69  | 3     | 3          | 6           | 1.23e+06   | 0   |
| 1.92e+03 | 45.5 | -3.92  | 1     | 0          | 4           | 1.29e+06   | 0   |
| 1.92e+03 | 45.6 | -6.87  | 1     | 0          | 7           | 1.43e+06   | 0   |
| 1.93e+03 | 48.5 | -27.7  | 3     | 0          | 3           | 1.53e+06   | 0   |
| 1.93e+03 | 51.7 | -8.26  | 1     | 0          | 1           | 1.58e+06   | 0   |
| 1.93e+03 | 39.8 | 0.732  | 3     | 4          | 1           | 1.64e+06   | 1   |

## Etiquetas

La base de datos contiene los siguientes elementos:

| Variable       | Almacenamiento | Visualización | Valor | Etiqueta de Variable  |
|----------------|----------------|---------------|-------|---|
| state          | float          | %9.0g         |       | ID del Estado   |
| year           | float          | %10.0g        |       | Año de Elección   |
| vote           | float          | %9.0g         |       | Proporción de votos demócratas en la próxima elección   |
| margin         | float          | %9.0g         |       | Margen de victoria demócrata  |
| class          | float          | %9.0g         |       | Clase del Senado  |
| termshouseint  |                | %54.0g        |       | Número acumulado de mandatos servidos en la Cámara de Representantes de EE.UU. por el congreso en funciones |
| termssenatint  |                | %53.0g        |       | Número acumulado de mandatos servidos en el Senado de EE.UU. por el congreso en funciones                   |
| populationlong |                | %10.0g        |       | Población del Estado  |
| win            | float          | %9.0g         |       | = 1 si el margen > 0  |

## Tabla estadística

Siempre se hace indispensable el análisis estadístico de la base de datos que se tiene.

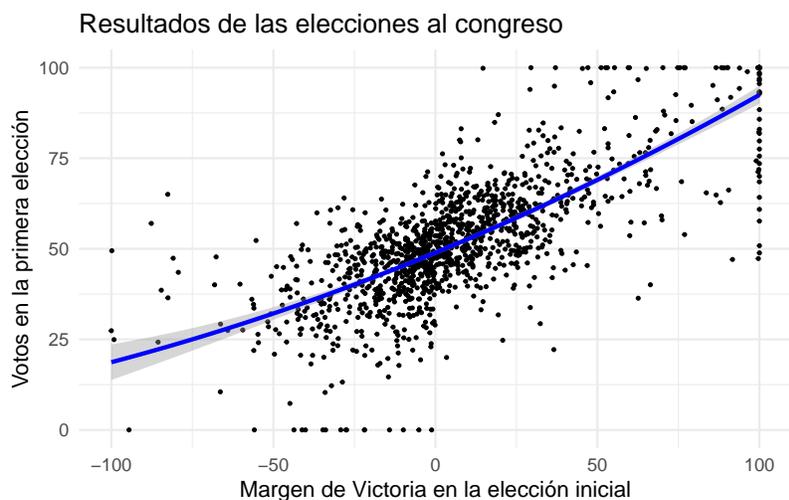
```
base.sen %>%
  dplyr::select(margin, vote, win)%>%
  summary()
```

| margin           | vote           | win             |
|------------------|----------------|-----------------|
| Min. : -100.000  | Min. : 0.00    | Min. : 0.0000   |
| 1st Qu.: -12.206 | 1st Qu.: 42.67 | 1st Qu.: 0.0000 |
| Median : 2.166   | Median : 50.55 | Median : 1.0000 |
| Mean : 7.171     | Mean : 52.67   | Mean : 0.5396   |
| 3rd Qu.: 22.766  | 3rd Qu.: 61.35 | 3rd Qu.: 1.0000 |
| Max. : 100.000   | Max. : 100.00  | Max. : 1.0000   |
|                  | NA's : 93      |                 |

## Grafico de entrada

Si tenemos continuidad, es posible graficar el conjunto de datos y con ello mirar su comportamiento.

```
# Gráfico
ggplot(base.sen, aes(x = margin, y = vote)) +
  geom_point(size = 0.5) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), color = "blue") +
  labs(title = "Resultados de las elecciones al congreso",
       y = "Votos en la primera elección",
       x = "Margen de Victoria en la elección inicial") +
  theme_minimal()
```

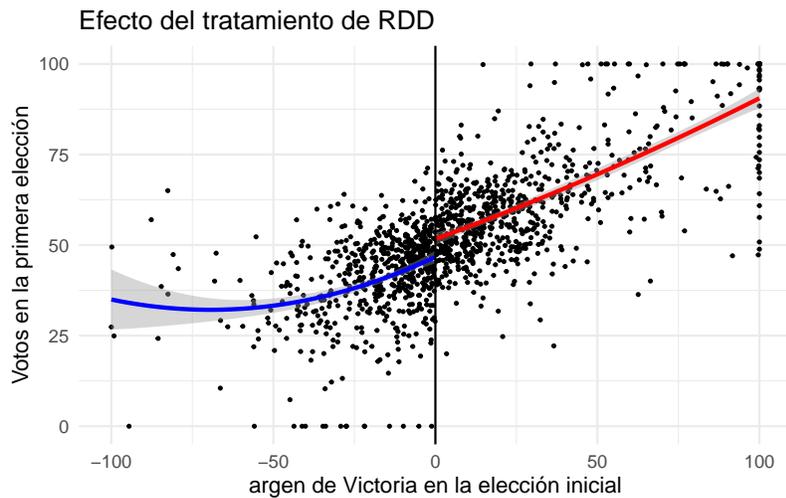


## Gráfico Sharp en RDD

Cuando se ha definido un punto de corte, empieza la parte para definir que grupo será tomado como **Tratado** y quien como **Control**

```
ggplot(base.sen, aes(x = margin, y = vote)) +
  geom_point(size = 0.5) +
  geom_smooth(data = filter(base.sen, margin < 0), method = "lm", formula = y ~ poly(x, 2), color = "red") +
  geom_smooth(data = filter(base.sen, margin > 0), method = "lm", formula = y ~ poly(x, 2), color = "blue") +
  geom_vline(xintercept = 0)
```

```
labs(title = "Efecto del tratamiento de RDD",
      y = "Votos en la primera elección",
      x = "argen de Victoria en la elección inicial") +
theme_minimal()
```



En un diseño sharp, el tratamiento se asigna de manera clara y sin ambigüedad en función de la variable de corte. No hay excepciones: todos los que están por encima del umbral reciben el tratamiento, y todos los que están por debajo no lo reciben.

## Regresión de polinomio

Crearemos un par de variables adicionales (al cuadrado) para mirar los efectos de forma funcional y estimar el modelo y analizar

```
base.sen <- base.sen %>%
  mutate(winmarg = win * margin,
         marginsq = margin^2,
         winmargsq = win * marginsq)
```

```
model_quad <- lm(vote ~ win + margin + marginsq + winmarg + winmargsq, data = base.sen)
coeftest(model_quad, vcov = vcovHC(model_quad, type = "HC1"))
```

t test of coefficients:

|              | Estimate   | Std. Error | t value | Pr(> t )  |     |
|--------------|------------|------------|---------|-----------|-----|
| (Intercept)  | 46.7395524 | 0.8421520  | 55.5001 | < 2.2e-16 | *** |
| win          | 4.9348166  | 1.1286846  | 4.3722  | 1.329e-05 | *** |
| margin       | 0.4206294  | 0.0789100  | 5.3305  | 1.155e-07 | *** |
| margin_sq    | 0.0030298  | 0.0012332  | 2.4568  | 0.01415   | *   |
| winmargin    | -0.0947090 | 0.0973555  | -0.9728 | 0.33083   |     |
| winmargin_sq | -0.0024027 | 0.0013705  | -1.7531 | 0.07982   | .   |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Note que al parecer si hay un **efecto** de los individuos que han sido tratados sobre los controles. Aquellos que definitivamente ganan la primera elección tienden a tener un margen muy considerable de ser elegidos en la próxima elección.

#### **i** Pendiente

La estipulación del punto de corte hace muy similares a individuos que definitivamente ganaron por poco en comparación con aquellos que perdieron la elección también por muy poco.

## Estimación de modelo simple

Si miramos un modelo sin las co-variables, podríamos tener un acercamiento como:

```
model_notreat <- lm(vote ~ margin, data = base.sen)
coeftest(model_notreat, vcov = vcovHC(model_notreat, type = "HC1"))
```

t test of coefficients:

|             | Estimate  | Std. Error | t value | Pr(> t )  |     |
|-------------|-----------|------------|---------|-----------|-----|
| (Intercept) | 49.536828 | 0.341114   | 145.221 | < 2.2e-16 | *** |
| margin      | 0.396699  | 0.012697   | 31.244  | < 2.2e-16 | *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Sin ninguno de los controles pero siendo un modelo robusto (por el tratamiento de los errores), Tenemos tambien un efecto pero algo menor.

## Modelo lineal

Pondremos ahora a prueba asumiendo que la relación es lineal, un modelo de tipo:

```
model_linear <- lm(vote ~ win + margin, data = base.sen)
coeftest(model_linear, vcov = vcovHC(model_linear, type = "HC1"))
```

t test of coefficients:

|             | Estimate  | Std. Error | t value | Pr(> t )      |
|-------------|-----------|------------|---------|---------------|
| (Intercept) | 47.330843 | 0.526218   | 89.9454 | < 2.2e-16 *** |
| win         | 4.784610  | 0.859808   | 5.5647  | 3.188e-08 *** |
| margin      | 0.348062  | 0.017043   | 20.4223 | < 2.2e-16 *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Seguimos teniendo efecto y positivo.

## Otros tipo de regresión

Podemos aprovechar y de acuerdo a las distintas formas funcionales o  $f(x)$  de la distribución de selección podemos mirar lo siguiente:

```
# Incluyendo una interacción
model_sepquad <- lm(vote ~ win + margin + marginsq + winmarg + winmargsq, data = base.sen)
coeftest(model_sepquad, vcov = vcovHC(model_sepquad, type = "HC1"))
```

t test of coefficients:

|             | Estimate  | Std. Error | t value | Pr(> t )      |
|-------------|-----------|------------|---------|---------------|
| (Intercept) | 46.739524 | 0.8421520  | 55.5001 | < 2.2e-16 *** |
| win         | 4.9348166 | 1.1286846  | 4.3722  | 1.329e-05 *** |

```

margin      0.4206294  0.0789100  5.3305 1.155e-07 ***
marginsq    0.0030298  0.0012332  2.4568  0.01415 *
winmarg     -0.0947090  0.0973555 -0.9728  0.33083
winmargsq   -0.0024027  0.0013705 -1.7531  0.07982 .
---

```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Un modelo mas

Cuando se miran las observaciones<sup>1</sup> tenemos que considerar que no hay efecto alguno, esto porque se esta moviendo el umbral

<sup>1</sup> Se esta filtrando por cutoff menor a 25 el margen va ser negativo y por ello hay que corregir por valor absoluto

```

model_local <- lm(vote ~ win + margin + marginsq + winmarg + winmargsq, data = filter(base.sen
coefstest(model_local, vcov = vcovHC(model_local, type = "HC1"))

```

t test of coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.2993286  1.3344346 33.9465 < 2.2e-16 ***
win          6.7502090  1.7118423  3.9432 8.71e-05 ***
margin       0.1691664  0.2764715  0.6119  0.5408
marginsq    -0.0028464  0.0114293 -0.2490  0.8034
winmarg      0.2826828  0.3548650  0.7966  0.4259
winmargsq   -0.0070250  0.0145293 -0.4835  0.6289
---

```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Modelo con errores robustos tipo cluster

Los errores estándar robustos clusterizados ajustan tanto para la heterocedasticidad como para la correlación dentro de los clusters. Permiten que los errores estén correlacionados dentro de los clusters (por ejemplo, empresas, escuelas) pero asumen *independencia* entre los clusters. Esto proporciona estimaciones más precisas de los *errores estándar*, reflejando adecuadamente la estructura de dependencia en los datos.

```

model_localclu <- lm(vote ~ win + margin + marginsq + winmarg + winmargsq, data = filter(base.
coefstest(model_localclu, vcov = vcovCL(model_localclu, cluster = ~state))

```

t test of coefficients:

|             | Estimate   | Std. Error | t value | Pr(> t )      |
|-------------|------------|------------|---------|---------------|
| (Intercept) | 45.2993286 | 1.4357662  | 31.5506 | < 2.2e-16 *** |
| win         | 6.7502090  | 1.7794762  | 3.7934  | 0.0001593 *** |
| margin      | 0.1691664  | 0.2721161  | 0.6217  | 0.5343279     |
| marginsq    | -0.0028464 | 0.0115833  | -0.2457 | 0.8059514     |
| winmarg     | 0.2826828  | 0.3568624  | 0.7921  | 0.4285066     |
| winmargsq   | -0.0070250 | 0.0151406  | -0.4640 | 0.6427817     |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Acá ambos grupos al parecer son similares y no existe diferencia en efecto por ellos.

Usamos el paquete de **Stargazer** para darle propiedad y hablar de alguna comparación si existe en alguno de los modelos.

```

# Extraer los errores estandar de cada modelo
se_notreat <- coefstest(model_notreat, vcov = vcovHC(model_notreat, type = "HC1"))[, "Std. Error"]
se_linear <- coefstest(model_linear, vcov = vcovHC(model_linear, type = "HC1"))[, "Std. Error"]
se_sepquad <- coefstest(model_sepquad, vcov = vcovHC(model_sepquad, type = "HC1"))[, "Std. Error"]
se_local <- coefstest(model_local, vcov = vcovHC(model_local, type = "HC1"))[, "Std. Error"]
se_localclu <- coefstest(model_localclu, vcov = vcovCL(model_localclu, cluster = ~state))[, "Std. Error"]

## Etiquetas
models <- list(
  "No tratamiento" = model_notreat,
  "Modelo lineal" = model_linear,
  "Tratamiento cuadrático" = model_sepquad,
  "LATE " = model_local,
  "LATE con cluster" = model_localclu
)

# Prepare the list of standard errors for huxreg
se_list <- list(
  se_notreat,

```

```

se_linear,
se_sepquad,
se_local,
se_localclu
)

# Combine models and standard errors into huxreg
huxreg_output <- huxreg(
  models,
  coefs = c("win" = "win", "margin" = "margin", "marginsq" = "marginsq",
            "winmarg" = "winmarg", "winmargsq" = "winmargsq", "(Intercept)" = "(Intercept)"),
  statistics = c(N = "nobs", R2 = "r.squared"),
  error_pos = "below",
  robust_se = se_list
)

print(huxreg_output)

```

|                 | No<br>tratamient<br>o | Modelo<br>lineal      | Tratamient<br>o<br>cuadrático | LATE                  | LATE con<br>cluster   |
|-----------------|-----------------------|-----------------------|-------------------------------|-----------------------|-----------------------|
| win             |                       | 4.785 ***<br>(0.923)  | 4.935 ***<br>(1.256)          | 6.750 ***<br>(1.783)  | 6.750 ***<br>(1.783)  |
| margin          | 0.397 ***<br>(0.010)  | 0.348 ***<br>(0.013)  | 0.421 ***<br>(0.069)          | 0.169<br>(0.257)      | 0.169<br>(0.257)      |
| marginsq        |                       |                       | 0.003 **<br>(0.001)           | -0.003<br>(0.011)     | -0.003<br>(0.011)     |
| winmarg         |                       |                       | -0.095<br>(0.087)             | 0.283<br>(0.361)      | 0.283<br>(0.361)      |
| winmargsq       |                       |                       | -0.002 *<br>(0.001)           | -0.007<br>(0.015)     | -0.007<br>(0.015)     |
| (Intercept<br>) | 49.537 ***<br>(0.339) | 47.331 ***<br>(0.542) | 46.740 ***<br>(0.896)         | 45.299 ***<br>(1.250) | 45.299 ***<br>(1.250) |
| N               | 1297                  | 1297                  | 1297                          | 845                   | 845                   |
| R2              | 0.569                 | 0.578                 | 0.591                         | 0.290                 | 0.290                 |

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

Column names: names, No tratamiento, Modelo lineal, Tratamiento cuadrático, LATE , LATE con cluster

```
## Opcion de html externo  
#quick_html(huxreg_output, file = "regression_table.html")
```

## Aclarando los efectos

ATE: Responde a la pregunta “¿Cuál es el efecto promedio del tratamiento si pudiéramos asignarlo aleatoriamente a toda la población?”

LATE: Responde a la pregunta “¿Cuál es el efecto promedio del tratamiento para aquellos individuos cuya decisión de tratamiento fue afectada por la variable instrumental?”

Encontramos que los efectos son significativos hasta el 95% de significancia para los modelos donde estamos suponiendo que exista aleatorización. Cuando miramos el LATE vemos que no es así. Lo que podría finalmente significar que en los periodos electorales podrían haber casos excepcionales incluso o candidatos que perdieron antes pero en una futura logran ser los ganadores o alcanzar un nivel más alto en utilidad/bienestar si se tratara de un programa social.

### Cuidado

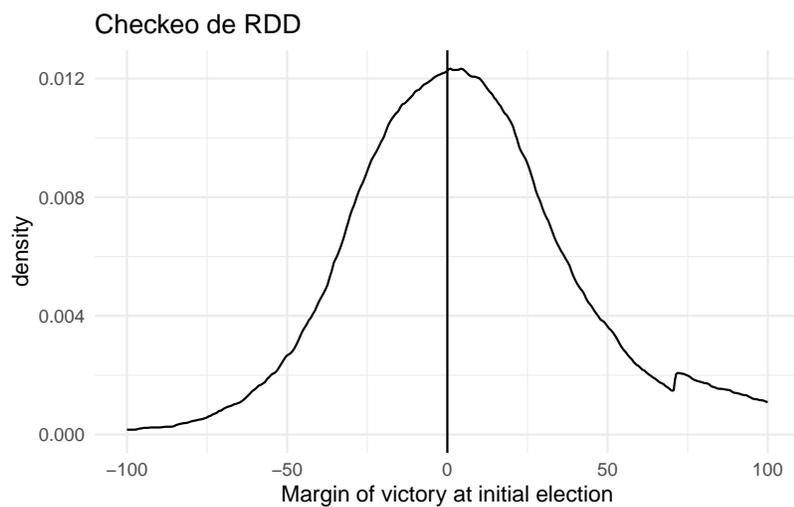
**LATE:** Relaja el supuesto de que todos pueden ser tratados o no tratados y se basa en una variable instrumental que afecta la probabilidad de recibir el tratamiento pero no directamente los resultados, excepto a través del tratamiento.

## Test de McCreary

El test de McCrary es esencial para validar la integridad de los resultados en la **regresión discontinua**, asegurando que los

resultados no están sesgados por una selección inadecuada o manipulación en la variable de asignación cerca del umbral.

```
# Kernel density plot
ggplot(base.sen, aes(x = margin)) +
  geom_density(kernel = "rectangular", adjust = 3) +
  geom_vline(xintercept = 0) +
  labs(title = "Checkeo de RDD",
       x = "Margin of victory at initial election") +
  theme_minimal()
```



Por lo pronto se cumple el supuesto!! El test de McCrary se convierte en una prueba diagnóstica utilizada en el contexto de los diseños de discontinuidad de regresión o RDD para evaluar si hay manipulación en la variable de corte (o running variable). En un RDD, se asume que los individuos no pueden manipular precisamente su posición respecto al umbral de tratamiento.

## Agradecimientos

Mucho de este trabajo se debe a los artículos, recursos (free commons) y material del Profesor Colin Cameron, autor del libro: Microeconometrics Using Stata.

**Carlos Yanes Guerra | Departamento de Economía | Universidad del Norte**