



```

/*****
Semana 10: Power calculations

Universidad de San Andrés
Economía Aplicada

2022

>
*****/
*      Bronstein      García Vassallo      López      Riottini
/*****/
Este archivo sigue la siguiente estructura:

0) Set up environment

1) Replicación de gráficos

*****/
* 0) Set up environment
*=====

global main "C:\Users\Franco\Documents\GitHub\Problem-Sets\PS 9"
global output "$main/output"
global input "$main/input"

cd "$main"

* Generamos una muestra simulada con datos de pagos de impuestos de empresas
*=====
clear all
set seed 123 // seteamos semilla para poder replicar los resultados
set obs 15000
gen ganancias_estimadas = rnormal(10000,2000)
drop if ganancias_estimadas<0

gen impuestos_pagados = 0.2*ganancias_estimadas + rnormal(0,500) // este es el termino
> de error que hay que modificar en el 2.
drop if impuestos_pagados<0
*=====

* Simulamos un efecto del 2.5% con 2000 obs
*=====
preserve

sample 2000, count // aleatoriamente me quedo con 2000 obs

* Aleatoriamente asigno el tratamiento al 50% de las obs:
gen temp = runiform()
gen T=0
replace T = 1 if temp<0.5

* A los tratados les aumento el outcome un 2.5%
replace impuestos_pagados = impuestos_pagados * (1+0.025) if T==1

* Regreso el outcome en el tratamiento
reg impuestos_pagados T, robust

restore
*=====

* Repetimos 500 veces y nos fijamos el % de veces que rechazamos H0
*=====
mat R = J(500,2,.)
forvalues x=1(1)500 {
preserve

sample 2000, count

gen temp = runiform()
gen T=0
replace T = 1 if temp<0.5

```

```

replace impuestos_pagados = impuestos_pagados * (1+0.025) if T==1
reg impuestos_pagados T, robust
mat R[`x',1]=_b[T]/_se[T]

restore
}

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
restore
*=====

* Repetimos la simulación pero para distintos tamaños de muestra y para distintos efec
> tos
*=====
local i=1
mat resultados = J(50,4,.)

foreach efecto in 0.01 0.025 0.05 0.075 0.1{
forvalues size = 1000(1000)10000 {
mat R = J(500,2,.)

forvalues x=1(1)500 {

preserve

sample `size' , count

gen temp = runiform()
gen T=0
replace T = 1 if temp<0.5

replace impuestos_pagados = impuestos_pagados * (1+`efecto') if T==1

reg impuestos_pagados T, robust
mat R[`x',1]=_b[T]/_se[T]

restore
}

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
scalar media = r(mean)

mat resultados[`i',3] = `efecto'
mat resultados[`i',2] = media
mat resultados[`i',1] = `size'
restore

local i=`i'+1
}
}

clear
svmat resultados

```

```

rename resultados1 sample_size
rename resultados2 st_power
rename resultados3 efecto

replace st_power=round(st_power,.01)
separate st_power, by(efecto)
*=====

* Gráfico
*=====
set scheme slcolor
twoway (connected st_power1 sample_size) (connected st_power2 sample_size) ///
      (connected st_power3 sample_size) (connected st_power4 sample_size) ///
      (connected st_power5 sample_size), ytitle("Power") ///
      xtitle("Number of observations") ///
      legend(label(1 "1%") label(2 "2.5%") label(3 "5%") label(4 "7.5%") label(5 "10%")) ///
      legend(rows(1) title("Effect")) xscale(titlegap(3)) yscale(titlegap(3))
graph export "$main/output/Graph 1.png", replace
*=====

* Generamos una muestra simulada con datos de pagos de impuestos de empresas
* Ahora seteamos la var. del error en 5000
*=====
clear all
set seed 1234 // seteamos semilla para poder replicar los resultados
set obs 15000
gen ganancias_estimadas = rnormal(10000,2000)
drop if ganancias_estimadas<0

gen impuestos_pagados = 0.2*ganancias_estimadas + rnormal(0,5000) // este es el termin
> o de error que hay que modificar en el 2.
drop if impuestos_pagados<0
*=====

* Repetimos la simulación pero para distintos tamaños de muestra y para distintos efec
> tos
*=====
local i=1
mat resultados = J(50,4,.)

foreach efecto in 0.01 0.025 0.05 0.075 0.1{
  forvalues size = 1000(1000)10000 {
    mat R = J(500,2,.)

    forvalues x=1(1)500 {

      preserve

      sample `size' , count

      gen temp = runiform()
      gen T=0
      replace T = 1 if temp<0.5

      replace impuestos_pagados = impuestos_pagados * (1+`efecto') if T==1

      reg impuestos_pagados T, robust

      mat R[`x',1]=_b[T]/_se[T]

      restore
    }
  }
}

```

```

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
scalar media = r(mean)

mat resultados['i',3] = `efecto'
mat resultados['i',2] = media
mat resultados['i',1] = `size'
restore

local i=`i'+1
}
}

clear
svmat resultados

rename resultados1 sample_size
rename resultados2 st_power
rename resultados3 efecto

replace st_power=round(st_power,.01)
separate st_power, by(efecto)
*=====

* Gráfico
*=====
set scheme slcolor
twoway (connected st_power1 sample_size) (connected st_power2 sample_size) ///
(connectd st_power3 sample_size) (connected st_power4 sample_size) ///
(connectd st_power5 sample_size), ytitle("Power") ///
xtitle("Number of observations") ///
legend(label(1 "1%") label(2 "2.5%") label(3 "5%") label(4 "7.5%") label(5 "10%")) ///
legend(rows(1) title("Effect")) xscale(titlegap(3)) yscale(titlegap(3))
graph export "$main/output/Graph 2.png", replace
*=====

* Generamos una muestra simulada con datos de pagos de impuestos de empresas
* Ahora seteamos la var. del error en 5000
*=====
clear all
set seed 1234 // seteamos semilla para poder replicar los resultados
set obs 15000
gen ganancias_estimadas = rnormal(10000,2000)
drop if ganancias_estimadas<0

gen impuestos_pagados = 0.2*ganancias_estimadas + rnormal(0,5000) // este es el termin
> o de error que hay que modificar en el 2.
drop if impuestos_pagados<0
*=====

* Repetimos la simulación pero para distintos tamaños de muestra y para distintos efec
> tos
* Asignando tratamiento al 20% de las obs.
*=====
local i=1
mat resultados = J(50,4,.)

foreach efecto in 0.01 0.025 0.05 0.075 0.1{
forvalues size = 1000(1000)10000 {
mat R = J(500,2,.)

forvalues x=1(1)500 {

preserve

```

```

sample `size' , count

gen temp = runiform()
gen T=0
replace T = 1 if temp<0.2

replace impuestos_pagados = impuestos_pagados * (1+`efecto') if T==1

reg impuestos_pagados T, robust

mat R[`x',1]=_b[T]/_se[T]

restore
}

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
scalar media = r(mean)

mat resultados[`i',3] = `efecto'
mat resultados[`i',2] = media
mat resultados[`i',1] = `size'
restore

local i=`i'+1
}
}

clear
svmat resultados

rename resultados1 sample_size
rename resultados2 st_power
rename resultados3 efecto

replace st_power=round(st_power,.01)
separate st_power, by(efecto)
*=====

* Gráfico
*=====
set scheme slcolor
twoway (connected st_power1 sample_size) (connected st_power2 sample_size) ///
(connect st_power3 sample_size) (connected st_power4 sample_size) ///
(connect st_power5 sample_size), ytitle("Power") ///
xtitle("Number of observations") ///
legend(label(1 "1%") label(2 "2.5%") label(3 "5%") label(4 "7.5%") label(5 "10%")) ///
legend(rows(1) title("Effect")) xscale(titlegap(3)) yscale(titlegap(3))
graph export "$main/output/Graph 3.png", replace
*=====

* Generamos una muestra simulada con datos de pagos de impuestos de empresas
* Ahora seteamos la var. del error en 5000
*=====
clear all
set seed 1234 // seteamos semilla para poder replicar los resultados
set obs 15000
gen ganancias_estimadas = rnormal(10000,2000)
drop if ganancias_estimadas<0

gen impuestos_pagados = 0.2*ganancias_estimadas + rnormal(0,5000) // este es el termin
> o de error que hay que modificar en el 2.
drop if impuestos_pagados<0
*=====

```

```

* Repetimos la simulación pero para distintos tamaños de muestra y para distintos efec
> tos
* Asignando tratamiento al 80% de las obs.
*=====*
local i=1
mat resultados = J(50,4,.)

foreach efecto in 0.01 0.025 0.05 0.075 0.1{
forvalues size = 1000(1000)10000 {
mat R = J(500,2,.)

forvalues x=1(1)500 {

preserve

sample `size' , count

gen temp = runiform()
gen T=0
replace T = 1 if temp<0.8

replace impuestos_pagados = impuestos_pagados * (1+`efecto') if T==1

reg impuestos_pagados T, robust

mat R[`x',1]=_b[T]/_se[T]

restore
}

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
scalar media = r(mean)

mat resultados[`i',3] = `efecto'
mat resultados[`i',2] = media
mat resultados[`i',1] = `size'
restore

local i=`i'+1
}
}

clear
svmat resultados

rename resultados1 sample_size
rename resultados2 st_power
rename resultados3 efecto

replace st_power=round(st_power,.01)
separate st_power, by(efecto)
*=====*

```

```

* Gráfico
*=====
set scheme slcolor
twoway (connected st_power1 sample_size) (connected st_power2 sample_size) ///
      (connected st_power3 sample_size) (connected st_power4 sample_size) ///
      (connected st_power5 sample_size), ytitle("Power") ///
xtitle("Number of observations") ///
legend(label(1 "1%") label(2 "2.5%") label(3 "5%") label(4 "7.5%") label(5 "10%")) ///
legend(rows(1) title("Effect")) xscale(titlegap(3)) yscale(titlegap(3))
graph export "$main/output/Graph 4.png", replace
*=====

*Generamos una muestra simulada con datos de pagos de impuestos de empresas
*Agregamos datos de pagos de impuestos de empresas en 2019 a nuestra muestra simulada
*=====
clear all
set seed 1235 // seteamos semilla para poder replicar los resultados
set obs 15000
gen ganancias_estimadas = rnormal(10000,2000)
drop if ganancias_estimadas<0
gen ganancias_estimadas_2019 = rnormal(10000,2000)
drop if ganancias_estimadas<0

gen impuestos_pagados = 0.2*ganancias_estimadas + rnormal(0,500)
drop if impuestos_pagados<0
*=====

* Repetimos simulación de un efecto del 2.5% con 2000 obs
*=====
preserve

sample 2000, count // aleatoriamente me quedo con 2000 obs

* Aleatoriamente asigno el tratamiento al 50% de las obs:
gen temp = runiform()
gen T=0
replace T = 1 if temp<0.5

* A los tratados les aumento el outcome un 2.5%
replace impuestos_pagados = impuestos_pagados * (1+0.025) if T==1

* Regreso el outcome en el tratamiento
reg impuestos_pagados T ganancias_estimadas_2019, robust

restore
*=====

* Repetimos 500 veces y nos fijamos el % de veces que rechazamos H0
*=====
mat R = J(500,2,.)
forvalues x=1(1)500 {
preserve

sample 2000, count

gen temp = runiform()
gen T=0
replace T = 1 if temp<0.5

replace impuestos_pagados = impuestos_pagados * (1+0.025) if T==1

reg impuestos_pagados T ganancias_estimadas_2019, robust

mat R[`x',1]=_b[T]/_se[T]

restore
}

```

```

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
restore
*=====*

* Repetimos la simulación pero para distintos tamaños de muestra y para distintos efec
> tos
*=====*
local i=1
mat resultados = J(50,4,.)

foreach efecto in 0.01 0.025 0.05 0.075 0.1{
forvalues size = 1000(1000)10000 {
mat R = J(500,2,.)

forvalues x=1(1)500 {

preserve

sample `size' , count

gen temp = runiform()
gen T=0
replace T = 1 if temp<0.5

replace impuestos_pagados = impuestos_pagados * (1+`efecto') if T==1

reg impuestos_pagados T ganancias_estimadas_2019, robust

mat R[`x',1]=_b[T]/_se[T]

restore
}

preserve
clear
svmat R
gen reject = 0
replace reject = 1 if (R1>1.65)
drop if reject==.
sum reject
scalar media = r(mean)

mat resultados[`i',3] = `efecto'
mat resultados[`i',2] = media
mat resultados[`i',1] = `size'
restore

local i=`i'+1
}
}

clear
svmat resultados

rename resultados1 sample_size
rename resultados2 st_power
rename resultados3 efecto

replace st_power=round(st_power,.01)
separate st_power, by(efecto)
*=====*

```



```

* Gráfico
*=====
set scheme slcolor
twoway (connected st_power1 sample_size) (connected st_power2 sample_size) ///
      (connected st_power3 sample_size) (connected st_power4 sample_size) ///
      (connected st_power5 sample_size), ytitle("Power") ///
xtitle("Number of observations") ///
legend(label(1 "1%") label(2 "2.5%") label(3 "5%") label(4 "7.5%") label(5 "10%")) ///
legend(rows(1) title("Effect")) xscale(titlegap(3)) yscale(titlegap(3))
graph export "$main/output/Graph 5.png", replace
*=====

*Exportamos a PDF
*=====
translate "$main/programs/PS 9.do" "$output/PS 9 do-file.pdf", translator(txt2pdf) rep
> lace

```