

Tarea 3 - Franco Díaz - Nicolás Vallejos

December 5, 2022

0.1 Tarea 3 - Franco Díaz y Nicolás Vallejos

0.1.1 Parte 1 - Experimentos

1. Asumiendo la existencia de recursos disponibles e implementacion a nivel de estudiante, sugiera un tratamiento que pueda ser testeado a traves de un experimento aleatorizado controlado. Sea especifico en cuanto a los detalles del tratamiento (costos, materiales, duracion, etcetera).

Se sugiere el tratamiento de implementar unas encuestas de contenidos a los alumnos pertenecientes a una asignatura en específico, estos alumnos serán divididos entre grupo control y grupo experimental, donde el grupo experimental será escogido de manera aleatoria a partir del registro de alumnos en la asignatura. Además, esta encuesta pregunta por preferencias que ellos tengan con respecto a las temáticas que se ven durante el transcurso de la asignatura (considerando temas de preferencia y posibles ejemplos prácticos).

Como aspecto a destacar, esta encuesta se aplicará a todos los alumnos en la clase final de cada semana, al término de esta, y se implementarán los cambios más votados por los alumnos en la siguiente sesión de clases, pero solo serán consideradas las respuestas registradas de parte del grupo experimental, para que de esta forma se pueda disminuir en cierto grado la contaminación que pueda generar el tener conocimiento de la participación en este tratamiento.

En lo respectivo a los costos, estos serán relativamente bajos, tomando en consideración un promedio de 30 alumnos por asignatura y que una caja de hojas de 500 unidades tiene un costo aproximado de 5.800 pesos, por lo que para cubrir de manera semestral a los alumnos (que ocuparían un total de 720 hojas cada 6 meses) solo se requerirían 2 cajas, sumando un total de 11.600 pesos. En base a lo anteriormente expuesto, los recursos o materiales necesarios serán las hojas requeridas para cada encuesta, las cuales dependen directamente del número de alumnos de la asignatura.


En cuanto a los plazos, se planea que el tratamiento tenga una duración de 6 meses (o un año para comparar semestre 1 y semestre 2).

Un supuesto importante a tener en consideración para la aplicación de todo este tratamiento es que: los alumnos deciden por su propia voluntad y no por un sesgo de grupo, lo que mejoraría en gran proporción la calidad de los resultados que se obtengan de los análisis.

2. Defina los grupos de tratamiento y control para implementar su experimento. Describa en detalle el mecanismo de asignacion aleatorio que permite la comparacion entre grupos.

El mecanismo de asignación aleatoria que permite la comparación es el siguiente: los grupos serán asignados de manera aleatoria a partir de la lista de clase, escogiendo a alumnos para el grupo

tratamiento hasta que estos compongan la mitad del curso (o la mitad +1 en caso de que el número de estudiantes cursando la asignatura sea impar)

3. Que metodo considera el mas apropiado para la estimacion del efecto promedio? (pre-test, pre-post test, Salomon 4 group). Justifique su respuesta en base a las ventajas y desventajas de cada metodo. 


El método más apropiado para la estimación del efecto promedio es el método pre-post test, esto debido principalmente a que con este método se contará con data del outcome previo a la intervención y permitiría una mejor comparación del efecto del tratamiento al obtener una medida de cambio (diferencia de diferencias).

Además, otro punto importante por el que nos inclinamos por este método es por las mismas limitaciones que tiene el aplicar solo post test:

- Ya que es un método que es difícil de generalizar y aplicar en todos los casos.
- Que presenta limitaciones en la información que brinda al no conocerse el grupo a través de un pre test, lo que provoca que no se pueda conocer si la relación causal del tratamiento está relacionada a los resultados.
- Que solo entrega información respecto al efecto promedio (diferencia de medias entre grupos).
- Que puede inducir en gran pérdida de la muestra.

4. Ahora suponga que no es posible implementar un experimento a nivel de estudiante, sino a nivel de clase. Como ajustaria los elementos de su experimento para poder ser implementado a nivel de cluster? Sea específico respecto tanto del tratamiento como del metodo de asignacion aleatorio y potencial comparacion entre grupos de tratamiento y control.

Para poder aplicar este mismo tratamiento a nivel de cluster, habría que ampliar la escala de la muestra, esta vez no limitándose a un solo curso, sino que, considerando ahora a diferentes cursos o secciones para una misma asignatura, con el supuesto de que siempre habrá más de una sección para cada asignatura, que entre estas no habrá un traspaso de información y que tampoco presentarán un sesgo grupal en las respuestas marcadas por cada estudiante.

En cuanto al método de asignación, este será una elección aleatoria entre el pool de secciones que se cuente para cada asignatura, realizando este proceso hasta contar con la mitad de las secciones en el grupo tratamiento y la otra mitad en el grupo control, siendo el grupo tratamiento la mitad más 1 en el caso de que las secciones sean impares. 

Además, otra particularidad del tratamiento entre la aplicación a individuos y a cluster, es que ahora los cambios sugeridos solo se aplicarán a los grupos experimentales, no a todos por igual, para así poder tener una comparativa de los efectos mucho más eficaz entre las secciones que no aplican tratamientos con respecto a las que efectivamente los implementan, teniendo un potencial de comparación entre grupos mucho mayor que el tratamiento original.

5. Suponga que en vez de un experimento, se planifica que sea un programa implementado a nivel de toda la Universidad. Como ajustaria los elementos descritos anteriormente para poder comparar el efecto de la intervencion.

Para el caso de una adaptación a nivel de toda la Universidad, se deben aplicar cambios tanto en el alcance de la encuesta como en el enfoque de esta, ya que los grupos control y experimental pasarían a ser todas las facultades del campus, y el enfoque que tendrían estas encuestas no serían

relacionados a cambios en asignaturas, sino cambios en la misma Universidad (sugerencias de cambios en ciertos procesos, horarios, nuevos espacios/implementos, etc).

La forma de dividir aleatoriamente a los grupos es similar a los procesos descritos anteriormente, seleccionando al azar entre las facultades pertenecientes a la Universidad hasta que tanto grupo experimental como grupo control cuenten con la mitad de las facultades para cada uno y considerando que en el caso de que el número de facultades sea impar, el grupo experimental será la mitad más 1.

Ahora, la forma de aplicación sería equivalente, considerando los cambios más votados entre todas las facultades del grupo control, donde se entrega una encuesta a cada facultad y las respuestas registradas son marcadas por un consenso entre las directivas o centros de alumnos de cada una de estas. En lo relacionado a los plazos, los cambios a implementar serán registrados a inicio de cada mes e implementados en el siguiente, debido a que pueden haber opciones que requieran de un mayor espacio de tiempo para poder ser aplicadas, por lo que el rango de evaluación del tratamiento será de 1 año.

Los supuestos a considerar en este nivel es que: no hay intercambio de información entre las directivas y no se produce sesgo de grupo en las respuestas registradas por cada facultad.

0.1.2 Parte 2 - Experimento Controlado

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
import random
import sklearn
import scipy
from scipy.linalg import eigh, cholesky
from scipy.stats import norm
import linearmodels.panel as lmp
from pylab import plot, show, axis, subplot, xlabel, ylabel, grid

%matplotlib inline
```

Primero, simulamos algunos datos para analizar el impacto potencial de un experimento controlado aleatorio (ECA).

Creamos las siguientes variables:

- tres variables aleatorias no correlacionadas (X1, X2, X3).
- estado de tratamiento (T)
- variable de tiempo (p)
- variable de conglomerado o agrupación (cl)
- variable de resultado en función del tiempo y la interacción del estado del tratamiento, más heterogeneidad aleatoria (y)

```

[2]: np.random.seed(13) #set seed
      nsize = 4000 #sample size

      noise = norm.rvs(size=(3, nsize)) #vector de variables N(0,1)
      noise = noise.transpose() #se transpone Xc
      Xc = pd.DataFrame(noise, columns=['x1','x2','x3']) # variables correlacionadas
      ↪ en formato de dataset (media cero)
      #time periods and treatment asignment
      Xc['p'] = 1 # "variable de tratamiento"
      Xc.loc[0:1999,'p'] = 0
      tr = np.random.binomial(1, 0.5, size=2000) #treatment status
      Xc.loc[0:1999,'T'] = tr
      Xc.loc[2000:3999,'T'] = tr

      #variable de grupos.
      Xc['c1'] = 1 # "variable de grupo"
      x = 0
      y = 49

      for i in range(40):
          Xc.loc[x:y,'c1'] = i + 1
          x += 50
          y += 50

      x = 2000
      y = 2049
      for i in range(40):
          Xc.loc[x:y,'c1'] = i + 1
          x += 50
          y += 50

      print(Xc)

      #outcome variable
      Xc['y'] = (0.85*Xc['x1']) + (0.61*(Xc['T']*Xc['p']*Xc['x2'])) + (-0.
      ↪ 17*Xc['p']*Xc['x3'])

      #descripción de la data
      Xc.describe()

```

	x1	x2	x3	p	T	c1
0	-0.712391	-0.486713	0.946033	0	1.0	1
1	0.753766	-0.620399	1.350281	0	1.0	1
2	-0.044503	1.333673	2.337030	0	0.0	1
3	0.451812	1.399984	-0.406571	0	1.0	1
4	1.345102	0.331703	-0.800677	0	0.0	1

```

...
3995  0.290293 -1.778934  0.952999  1  0.0  40
3996  0.054900  0.180794  1.392776  1  1.0  40
3997 -1.794603  0.991854 -2.333434  1  1.0  40
3998  1.441689  0.637163 -0.479020  1  1.0  40
3999  2.167645  0.340809  0.313752  1  1.0  40

```

[4000 rows x 6 columns]

```

[2]:
count    x1      x2      x3      p      T \
mean     0.002455  0.041325  0.002443  0.500000  0.509500
std      0.997459  1.009580  0.983740  0.500063  0.499972
min      -3.563094 -4.706426 -3.801465  0.000000  0.000000
25%      -0.666401 -0.641522 -0.652250  0.000000  0.000000
50%      -0.005215  0.045937 -0.005107  0.500000  1.000000
75%       0.662230  0.705770  0.660045  1.000000  1.000000
max       3.538423  3.837147  3.570808  1.000000  1.000000

count    cl      y
mean     20.50000  0.010133
std      11.54484  0.911615
min       1.00000 -3.733534
25%      10.75000 -0.601327
50%      20.50000 -0.005952
75%      30.25000  0.622285
max      40.00000  4.484399

```

```
[3]: Xc["y*"]=norm.cdf(Xc["y"])
```

```
[4]: Xc.describe()
```

```

[4]:
count    x1      x2      x3      p      T \
mean     0.002455  0.041325  0.002443  0.500000  0.509500
std      0.997459  1.009580  0.983740  0.500063  0.499972
min      -3.563094 -4.706426 -3.801465  0.000000  0.000000
25%      -0.666401 -0.641522 -0.652250  0.000000  0.000000
50%      -0.005215  0.045937 -0.005107  0.500000  1.000000
75%       0.662230  0.705770  0.660045  1.000000  1.000000
max       3.538423  3.837147  3.570808  1.000000  1.000000

count    cl      y      y*
mean     20.50000  0.010133  0.501313
std      11.54484  0.911615  0.272286

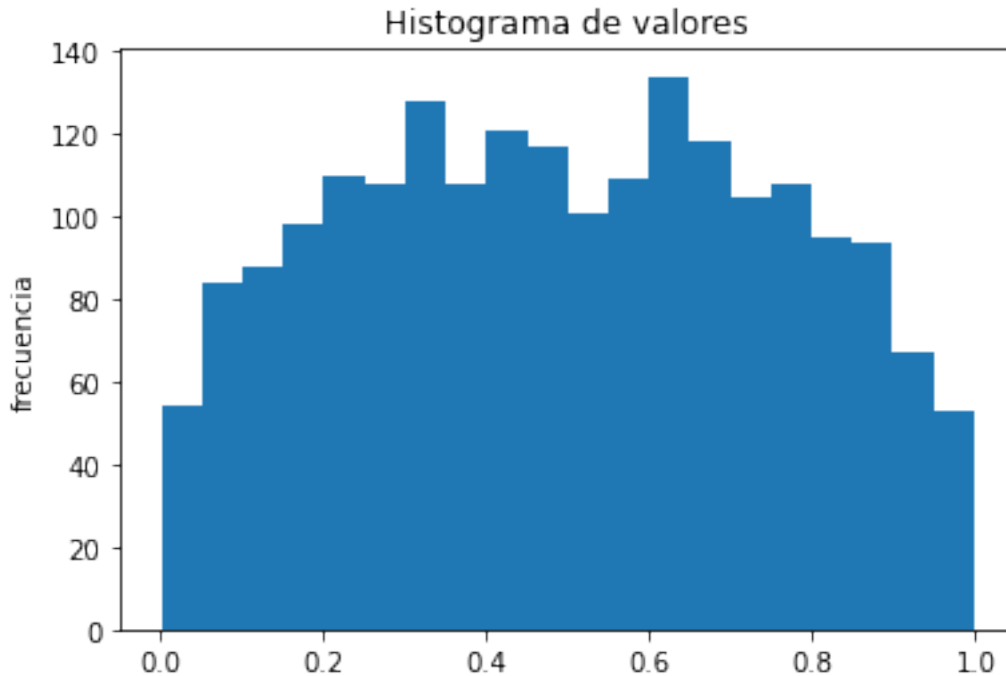
```

min	1.00000	-3.733534	0.000094
25%	10.75000	-0.601327	0.273811
50%	20.50000	-0.005952	0.497625
75%	30.25000	0.622285	0.733123
max	40.00000	4.484399	0.999996

```
[5]: plt.hist(Xc.loc[(Xc["p"]==0),"y"], 10)
plt.ylabel("frecuencia")
plt.title("Histograma de valores")
plt.show()
```



```
[6]: plt.hist(Xc.loc[(Xc["p"]==0),"y*"], 20)
plt.ylabel("frecuencia")
plt.title("Histograma de valores")
plt.show()
```



```
[7]: #División de grupos
#Se utiliza la metodología de corte arbitrario de fuerza bruta para definir los
    ↪ cortes de
#los grupos
#Grupo total pre test (asistencia promedio 80%)
Xc.loc[(Xc["y*"]<=0.76) & (Xc["p"]==0) , "y**"]=1
Xc.loc[(Xc["y*"]>0.76) & (Xc["p"]==0) , "y**"]=0

#Grupo total post test (asistencia promedio 75%)
Xc.loc[(Xc["y*"]<=0.61) & (Xc["p"]==1), "y**"]=1
Xc.loc[(Xc["y*"]>0.61) & (Xc["p"]==1), "y**"]=0

#Grupo Tratamiento post test (90% de asistencia)
Xc.loc[(Xc["y*"]<=0.91) & (Xc["p"]==1) & (Xc["T"]==1), "y**"]=1
Xc.loc[(Xc["y*"]>0.91) & (Xc["p"]==1) & (Xc["T"]==1), "y**"]=0
```

```
[8]: Xc.groupby(by=["p","T"]).mean()
```

```
[8]:
```

		x1	x2	x3	c1	y	y*	y**
p	T							
0	0.0	-0.012085	0.029170	0.003354	20.503568	-0.010272	0.495030	0.800204
	1.0	-0.011634	0.055821	0.023457	20.496565	-0.009889	0.497673	0.814524
1	0.0	-0.002813	0.032514	-0.003534	20.503568	-0.001791	0.497539	0.637105
	1.0	0.035612	0.047015	-0.013695	20.496565	0.061277	0.514637	0.887144

```
[9]: Xc.groupby(by=["p"]).mean()
```

```
[9]:
```

	x1	x2	x3	T	c1	y	y*	y**
p								
0	-0.011855	0.042749	0.013596	0.5095	20.5	-0.010077	0.496376	0.8075
1	0.016764	0.039902	-0.008711	0.5095	20.5	0.030343	0.506251	0.7645

0.1.3 Comentarios sobre resultados

Como se evidencia en los resultados de las tablas superiores, antes del tratamiento había en promedio un 80% de asistencia por parte de los alumnos. Bajando a un 75% de asistencia después de haber aplicado el experimento (como se muestra en la tabla 2). Además, para los que son parte del experimento su asistencia llega aproximadamente al 90%, tal y como se observa en la tabla 1 de esta sección.

```
[10]: #post-test
y = Xc.loc[2000:,"y**"]
X = Xc.loc[2000:,"T"]
X = sm.add_constant(X)
model = sm.Logit(y, X)
results_logit = model.fit()

#print(results_logit.summary())
#mfx = results_logit.get_margeff()
#print(mfx.summary())
```

Optimization terminated successfully.

Current function value: 0.500880

Iterations 6

Logit Regression Results

```
=====
Dep. Variable:          y**    No. Observations:          2000
Model:                Logit    Df Residuals:            1998
Method:                MLE     Df Model:                1
Date:                  Mon, 28 Nov 2022    Pseudo R-squ.:          0.08236
Time:                  20:36:37    Log-Likelihood:         -1001.8
converged:              True    LL-Null:                -1091.7
Covariance Type:        nonrobust    LLR p-value:            5.286e-41
=====
```

	coef	std err	z	P> z	[0.025	0.975]
const	0.5628	0.066	8.476	0.000	0.433	0.693
T	1.4991	0.119	12.575	0.000	1.265	1.733

Logit Marginal Effects

Dep. Variable:	y**					
Method:	dydx					
At:	overall					
	dy/dx	std err	z	P> z	[0.025	0.975]
T	0.2465	0.017	14.117	0.000	0.212	0.281

0.1.4 Interpretación post test

En la aplicación del método post test, se debe obtener un efecto promedio calculado a partir de la media de ambos grupos, por lo que, observando la figura superior, tenemos que el efecto de la aplicación del tratamiento en términos del aumento de la asistencia es de un 24,65%, lo que compensa el valor obtenido en la constante, que sería la asistencia del grupo control, y nos permite obtener finalmente una asistencia promedio entre ambos grupos del 80% aproximadamente.

```
[11]: #pre-post test
y=Xc['y**']
Xc["hh"] = Xc['p']*Xc['T'] #hh=p*T (variable interacción de tiempo con
    ↪tratamiento). Estimador de diferencias en diferencias
X=Xc[['p','T','hh']] #variables
X = sm.add_constant(X) #constante
model = sm.Logit(y, X) #modelo Logit
results_logit = model.fit()
print(results_logit.summary())
mfx = results_logit.get_margeff()
print(mfx.summary())
```

Optimization terminated successfully.

Current function value: 0.495272

Iterations 6

Logit Regression Results

=====						
Dep. Variable:	y**	No. Observations:	4000			
Model:	Logit	Df Residuals:	3996			
Method:	MLE	Df Model:	3			
Date:	Mon, 28 Nov 2022	Pseudo R-squ.:	0.04610			
Time:	20:36:37	Log-Likelihood:	-1981.1			
converged:	True	LL-Null:	-2076.8			
Covariance Type:	nonrobust	LLR p-value:	2.903e-41			
=====						
	coef	std err	z	P> z	[0.025	0.975]

const	1.3876	0.080	17.377	0.000	1.231	1.544
p	-0.8247	0.104	-7.942	0.000	-1.028	-0.621
T	0.0921	0.113	0.812	0.417	-0.130	0.314
hh	1.4070	0.165	8.549	0.000	1.084	1.730

Logit Marginal Effects						
Dep. Variable:	y**					
Method:	dydx					
At:	overall					
	dy/dx	std err	z	P> z	[0.025	0.975]
p	-0.1319	0.016	-8.139	0.000	-0.164	-0.100
T	0.0147	0.018	0.812	0.417	-0.021	0.050
hh	0.2250	0.026	8.733	0.000	0.174	0.275

0.1.5 Interpretación pre-post test

Ahora, para el caso del método pre-post test, el cual evalúa y compara los resultados obtenidos anterior y posterior a la aplicación del tratamiento, se tiene que:

- Una variación de una unidad en el periodo genera una disminución de la asistencia de un 13,19%.
- Para el caso del tratamiento, notamos que el hecho de aplicar el tratamiento tiene un impacto de un 1,47% en el aumento de la asistencia de los alumnos.
- Finalmente, la interacción del tiempo con el tratamiento genera un aumento de un 22% en la asistencia promedio de los alumnos a clase una vez finalizado el periodo de análisis.

Además, es importante plantear que los altos valores obtenidos en la sección de coeficientes, puede deberse a que el modelo no logra ajustar correctamente el efecto de las variables independientes.

```
[12]: #clustered standard errors
results3 = model.fit(cov_type="cluster", cov_kws={'groups': Xc['cl']})
print(results3.summary())
```

Optimization terminated successfully.

Current function value: 0.495272

Iterations 6

Logit Regression Results						
Dep. Variable:	y**	No. Observations:	4000			
Model:	Logit	Df Residuals:	3996			
Method:	MLE	Df Model:	3			
Date:	Mon, 28 Nov 2022	Pseudo R-squ.:	0.04610			
Time:	20:36:37	Log-Likelihood:	-1981.1			
converged:	True	LL-Null:	-2076.8			
Covariance Type:	cluster	LLR p-value:	2.903e-41			
	coef	std err	z	P> z	[0.025	0.975]
const	1.3876	0.094	14.734	0.000	1.203	1.572

p	-0.8247	0.117	-7.048	0.000	-1.054	-0.595
T	0.0921	0.126	0.733	0.464	-0.154	0.339
hh	1.4070	0.206	6.818	0.000	1.003	1.811

=====

0.2 Parte 3 - Experimentos Naturales

Simule un experimento natural (e.g. intervencion de politica publica) tal que se reduce la proporcion de individuos con 3 hijos o mas que declaran beber alcohol en el tercer periodo a la mitad. Para ello, genere una variable de tratamiento (todos los individuos con mas de 2 hijos son parte de la intervencion), y una nueva variable llamada sdrinky, talque es identica a drinkly en los periodos 1 y 2 , pero sustituya los valores aleatoriamente en el periodo 3 para generar el efecto esperado.

```
[13]: charls = pd.read_csv('../charls.csv')
charls.dropna(inplace=True)
charls.reset_index(drop=True, inplace=True)

#Convertimos las observaciones de la variable drinkly a enteras
for i in range(len(charls["drinkly"])):
    if charls["drinkly"][i] != ".m":
        charls["drinkly"][i] = int(charls["drinkly"][i])
charls.describe()
```

```
<ipython-input-13-fcdf3c02873e>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
charls["drinkly"][i] = int(charls["drinkly"][i])
```

```
[13]:
```

	age	bnrps	cesd	child	dnrps \
count	21045.000000	21045.000000	21045.000000	21045.000000	21045.000000
mean	59.386553	59.610683	8.656878	2.825232	0.740889
std	9.016106	51.905928	6.307677	1.372179	0.438157
min	20.000000	0.000000	0.000000	0.000000	0.000000
25%	52.000000	0.000000	4.000000	2.000000	0.000000
50%	59.000000	60.000000	7.000000	3.000000	1.000000
75%	65.000000	74.875404	12.000000	4.000000	1.000000
max	95.000000	300.000000	30.000000	10.000000	1.000000

	female	hrsusu	hsize	intmonth	married \
count	21045.000000	21045.000000	21045.000000	21045.000000	21045.000000
mean	0.521026	2.548166	3.585222	7.511143	0.907674
std	0.499570	1.757182	1.720136	0.865851	0.289492
min	0.000000	0.000000	1.000000	1.000000	0.000000
25%	0.000000	0.000000	2.000000	7.000000	1.000000
50%	1.000000	3.401197	3.000000	7.000000	1.000000
75%	1.000000	4.025352	5.000000	8.000000	1.000000

max	1.000000	5.123964	16.000000	12.000000	1.000000
-----	----------	----------	-----------	-----------	----------

	nrps	retage	retired	schadj	urban \
count	21045.000000	21045.000000	21045.000000	21045.000000	21045.000000
mean	0.519078	1.280969	0.204942	4.162414	0.206652
std	0.499648	3.830963	0.403669	3.540039	0.404914
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	0.000000	0.000000	4.000000	0.000000
75%	1.000000	0.000000	0.000000	8.000000	0.000000
max	1.000000	51.000000	1.000000	16.000000	1.000000

	wave	wealth	inid
count	21045.000000	2.104500e+04	21045.000000
mean	1.909385	6.783959e+03	12747.082870
std	0.817975	5.453065e+04	7769.025809
min	1.000000	-1.648450e+06	1.000000
25%	1.000000	1.000000e+02	5176.000000
50%	2.000000	1.000000e+03	13314.000000
75%	3.000000	6.800000e+03	19650.000000
max	3.000000	1.040000e+06	25403.000000

```
[14]: #Calculo proporción de bebedores actuales con más de hijos 2 en el periodo 3.
trat= 0
per3=0
for i in range(len(charls["wave"])):
    if charls["wave"][i]==3 and charls["drinkly"][i]!=".m":
        per3+=1
    if charls["drinkly"][i]==1 and charls["wave"][i]==3 and
↳charls["child"][i]>=3:
        trat+=1

print("La proporción de bebedores actuales antes del tratamiento es: "+str(trat/
↳per3))
print("La proporción de bebedores luego de aplicada la política pública debiese
↳ser: "+str(trat/(per3*2)))
```

La proporción de bebedores actuales antes del tratamiento es:

0.17209528439474964

La proporción de bebedores luego de aplicada la política pública debiese ser:

0.08604764219737482

```
[15]: #Creación variable de tratamiento
charls["tratamiento"] = 0
charls["tratamiento"][charls["child"] > 2] = 1

#Variable "sdrinkly"
```

```

charls["sdrinkly"] = charls["drinkly"]
for i in range(len(charls["sdrinkly"])):
    if charls["wave"][i]==3 and charls["child"][i]>=3 and (trat/per3)>0.
↳08604764219737482:
        x=random.uniform(0,1) #Se asigna de forma aleatoria si el individuo
↳deja de beber alcohol luego del tratamiento.
        if x>0.5:
            charls["sdrinkly"]=0
            trat-=1
print("El tratamiento generado por el gobierno hace que la proporción de
↳bebedores con 3 hijos o mas en el periodo 3 disminuya un 50%, llegando a
↳una propoción de: "+str(float(trat/per3)))

```

El tratamiento generado por el gobierno hace que la proporción de bebedores con 3 hijos o mas en el periodo 3 disminuya un 50%, llegando a una propoción de: 0.08604764219737482

<ipython-input-15-67813bdb8d8d>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
charls["tratamiento"][charls["child"] > 2] = 1
```