# Tarea3_Arancibia_Rios

December 10, 2022

Tarea 3 - César Arancibia, Francisco Ríos

[2]:
```
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 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
```

## 1 Parte 1

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).

En la búsqueda de medios y tratamientos capaces de ser aplicados a estudiantes específicos, se hace dificil la tarea de evitar la contaminación de los datos. Puesto que no son muchas las variables a controlar que no sean comentadas entre estudiantes y que descubran si son parte del grupo controlado. La principal idea del grupo de trabajo se basa en la aplicación de estímulos positivos para la participación y asistencia del alumno, no de aplicar castigos o medios obligatorios. Por ejemplo, una prueba o test por clase obligaría al estudiante a ir, pero no se puede aplicar a unos estudiantes si y otros no aleatoriamente. El mismo concepto eliminó la posible idea de aplicar buses de acercamiento para los estudiantes, desayunos en caso de que fuesen clases en la mañana, etc. En este sentido, nace la idea de un experimento sencillo para aumentar el porcentaje de asistencia individual con tal de aumentar la proporción de estudiantes que asisten a sus clases. El experimento se basa en la aplicación de estímulos verbales positivos en las interacciones con el profesor. Este tratamiento tendría un costo relativo a cero, y la probabilidad de que los alumnos "descubrieran" que son parte de un tratamiento es bastante baja. ¿A qué se refiere este tratamiento? A nivel de clase, los alumnos serían seleccionados aleatoriamente con tal de evaluar su asistencia, y clase a clase el profesor le felicitaría o mencionaría que su desempeño es positivo, destacando su presencia.

Esto no debería significar ninguna diferencia o preferencia en notas, décimas, actividades o aspectos académicos. Sólo un estímulo positivo emocional. La clase a evaluar debe delimitar correctamente el horario, y no variar ni en este aspecto ni en la cantidad de créditos o importancia relativa en la carrera, de evaluarse distintas asignaturas. La variable dependiente sería, como específica la instrucción, la asistencia promedio a la asignatura del estudiante. En este sentido, las variables a evaluar por alumno serían: Si reciben reforzamiento positivo por participación/asistencia a clase Cuantas veces reciben tratamiento en el periodo de estudio Su porcentaje de asistencia a clase en general Cantidad de horas de clase que tiene en el día de la aplicación del tratamiento Cantidad de días a la semana en que tiene clases Vive con su familia o es arrendatario. Distancia de su casa a la Universidad Primera nota en la asignatura Promedio notas en la asignatura Año cursado Cantidad de ramos reprobados hasta el momento Más un error fijo por individuo, debido a factores no observables tales como: Cantidad de amigos en la Universidad Cantidad de amigos en la asignatura Cercanía con su grupo social Apreciación personal de su grupo social Importancia que se le da a la asignatura El experimento debe ser aplicado durante una cantidad fija de tiempo, como por ejemplo un semestre, con tal de evaluar los resultados en un lapso de tiempo significativo. Ya que una semana, por ejemplo, no demostraría ningún resultado importante. La proporción promedio de estudiantes que asisten a clase debería reflejar de cierta manera la importancia de los estímulos positivos en la vida de los alumnos.

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.

Esta sería una asignación y selección aleatoria, en la que se espera que las variables sociales jueguen un rol importante. Una selección equitativa de estudiantes en el grupo control y en el grupo tratamiento, con variables similares, podría dejar ver la causalidad del reforzamiento positivo. Además, se puede extrapolar el resultado para sacar un contrafactual de los alumnos similares pero que no fueron afectados por el tratamiento.

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.

En este caso sería favorable el método de Solomon. Así no se perdería información en el tiempo, ya que la aplicación del tratamiento no es un momento específico sino una constante aplicación. Entonces, los cuatro grupos podrían ser comparados en función de la consistencia de los resultados. Por ejemplo, se estudiaría a los alumnos antes del tratamiento con tal de ver su participación y calidad de respuesta en la asignatura. Se aplicaría el tratamiento y luego se evaluaría un resultado posterior. El segundo grupo solo se vería afectado por los análisis previos y posteriores al tratamiento, pero sin pasar por este. Luego un tercer grupo no se evaluaría antes de, y un cuarto grupo solo vería el post test. Este método es apreciado por el grupo de trabajo en cuanto a la aplicación de este experimento al reducir los posibles efectos secundarios de la prueba previa al tratamiento. Además, permite controlar que la satisfacción, participación y calidad de los estudiantes en la asignatura sea observada por sí sola en cuatro escenarios completos. Como todos los alumnos se podrían ver como de igual interés, quizá la aplicación de un método simple tradicional haría perder ciertos resultados del experimento, haciéndolo ineficaz.

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 especifico respecto tanto del tratamiento como del metodo de asignacion aleatorio y potencial comparacion entre grupos de tratamiento y control.

Además, las variables discretas se transforman en binarias y las binarias no sé estandarizan cómo las demás para asegurar la cercanía entre individuos del grupo. La parte importante será asegurar la correlación entre los miembros entre el grupo pero la diferencia estadística entre los grupos. El experimento verá cambiado su enfoque más que nada en la forma de entregar el reforzamiento positivo a nivel grupal y se podía añadir preguntar por ciertas personas si faltan. Cosa de demostrar que se nota. Las variables individuales como las notas cambiarán a promedio de notas de todos los estudiantes. Y algunas como de si vive o no solo se debería cambiar a si la mayoría lo hace o no.

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

Sería lo mismo que la aplicación a Clúster. El problema sería que la aplicación del pre y post tratamiento, y el tratamiento tendría mucho más riesgo de ser contaminado por la comunicación. Además, pre y post deberá ser un programa completo a nivel universidad para asegurar la participación para no perder datos a lo largo del tratamiento. Se encontrarían también más problemas a nivel de aplicación, puesto que la heterogeneidad por temas de profesor, facultad y forma de aplicación de los tratamientos podría ser distinto.

## 2 Parte 2

6. A partir de sus respuestas en Parte 1, genere data para 40 grupos (considere cada grupo como una clase) con 50 estudiantes cada uno (asuma que los estudiantes son asignados aleatoriamente a cada clase). Cada estudiante debe tener data de asistencia en un periodo, generando una variable binaria aleatoria talque la asistencia promedio a traves de todos los grupos es de 80%.

[28]:
```
np.random.seed(0);

class clase:
    def __init__(self,_id):
        self._id = _id;
    def generar_est(self,cantidad,proporcion):
        self.c = cantidad;
        self.p = proporcion;
        irresponsables = np.sort(np.random.choice([j for j in range(self.c)],␣
 ↪round(self.c-(self.c*self.p)), replace=False));
        self.g = [];
        for j in range(self.c):
            if j in irresponsables:
                self.aux = estudiante(str(i)+"_"+str(j),0);
            else:
                self.aux = estudiante(str(i)+"_"+str(j),1);
            self.g.append(self.aux);
        return self.g;

class estudiante:
    def __init__(self,_id,asistencia):
```

```
        self._id = _id;
        self.a = asistencia;
    def return_asist(self):
        return self.a;

grupos = [];
grupos_val = [];
for i in range(40):
    cl = clase(i);
    aux = cl.generar_est(50,0.8);
    grupos.append(aux);
    aux2 = [];
    for k in range(len(aux)):
        aux2.append(aux[k].return_asist());
    grupos_val.append(aux2);

#grupos_val = np.array(grupos_val);

np.set_printoptions(threshold=np.inf);
grupos_val = [item for sub_item in grupos_val for item in sub_item];
#print(grupos_val)

grupos_val.extend(grupos_val.copy());
df = pd.DataFrame(grupos_val,columns=["A"]);

ruido = np.around(np.random.normal(loc=0,scale=1,size=4000),decimals=3);
df["X"] = ruido;
df["Cl"] = 0;
df["P"] = 0;
df.loc[2000:3999,"P"] = 1;
#print(df);
for i in range(len(df)//2):
    df.loc[i,"Cl"] = i // 50 + 1;
    df.loc[2000+i,"Cl"] = i // 50 + 1;
print(df);
print(df.describe());
```

```
      A      X  Cl  P
0     1  0.436   1  0
1     1 -0.599   1  0
2     0  0.033   1  0
3     1 -0.854   1  0
4     0 -0.720   1  0
…   ..    …   .. ..
3995  1  0.724  40  1
3996  1  0.606  40  1
3997  1 -1.290  40  1
```

```
3998  1  0.789  40  1
3999  0  1.961  40  1

[4000 rows x 4 columns]
                A            X          Cl            P
count  4000.00000  4000.000000  4000.00000  4000.000000
mean      0.80000    -0.009463    20.50000     0.500000
std       0.40005     0.987632    11.54484     0.500063
min       0.00000    -3.740000     1.00000     0.000000
25%       1.00000    -0.693750    10.75000     0.000000
50%       1.00000    -0.014500    20.50000     0.500000
75%       1.00000     0.656000    30.25000     1.000000
max       1.00000     3.802000    40.00000     1.000000
```

7. Genere un mecanismo de asignacion aleatorio a nivel de estudiante y muestre que en la data generada permite que ambos grupos (tratamiento y control) tienen una asistencia promedio comparable.

[29]:
```
np.random.seed(0);
opciones = np.sort(np.random.choice([i for i in␣
 ↪range(2000)],1000,replace=False));
df["T"] = 0;
for i in opciones:
    df.loc[i,"T"] = 1;
    df.loc[2000+i,"T"] = 1;

print(df);
print(df.describe());
print("Asistencia promedio grupo control:")
print(np.mean(df["A"][df["T"] == 0]));
print("Asistencia promedio grupo tratamiento:")
print(np.mean(df["A"][df["T"] == 1]));
```

```
      A      X  Cl  P  T
0     1  0.436   1  0  0
1     1 -0.599   1  0  1
2     0  0.033   1  0  1
3     1 -0.854   1  0  0
4     0 -0.720   1  0  1
…   ..    …  ..  .. ..
3995  1  0.724  40  1  1
3996  1  0.606  40  1  0
3997  1 -1.290  40  1  1
3998  1  0.789  40  1  0
3999  0  1.961  40  1  1

[4000 rows x 5 columns]
                A            X          Cl            P            T
```

```
count  4000.00000  4000.000000  4000.00000  4000.000000  4000.000000
mean      0.80000    -0.009463    20.50000     0.500000     0.500000
std       0.40005     0.987632    11.54484     0.500063     0.500063
min       0.00000    -3.740000     1.00000     0.000000     0.000000
25%       1.00000    -0.693750    10.75000     0.000000     0.000000
50%       1.00000    -0.014500    20.50000     0.500000     0.500000
75%       1.00000     0.656000    30.25000     1.000000     1.000000
max       1.00000     3.802000    40.00000     1.000000     1.000000
Asistencia promedio grupo control:
0.802
Asistencia promedio grupo tratamiento:
0.798
```

8. Genere un tratamiento que imcrementa la participacion en el grupo de tratamiento en 10 puntos porcentuales. Ademas en la data posterior al experimento, asuma que la participacion promedio cayo a 75%. Estime el efecto promedio del tratamiento usando solo post-test.

[30]:
```
#Y = a + B
df["Y"] = 0.85 * df["X"] + 0.61 * (df["T"] * df["P"]) + (-0.17 * df["P"]);
print(df.describe());
```

```
                A            X           Cl            P            T  \
count  4000.00000  4000.000000  4000.00000  4000.000000  4000.000000
mean      0.80000    -0.009463    20.50000     0.500000     0.500000
std       0.40005     0.987632    11.54484     0.500063     0.500063
min       0.00000    -3.740000     1.00000     0.000000     0.000000
25%       1.00000    -0.693750    10.75000     0.000000     0.000000
50%       1.00000    -0.014500    20.50000     0.500000     0.500000
75%       1.00000     0.656000    30.25000     1.000000     1.000000
max       1.00000     3.802000    40.00000     1.000000     1.000000

                Y
count  4000.000000
mean      0.059456
std       0.874681
min      -3.053200
25%      -0.538975
50%       0.049150
75%       0.641038
max       3.353800
```

[81]:
```
#df["Y"] = df["X"] + df["A"] * (1 - df["P"]) + df["T"] + df["A2"] * df["P"]# -␣
 ↪0.05 * df["P"];

#Normalizar Y
df["Y*"] = df["Y"];
df["Y"] = (df["Y"] - min(df["Y"]))/(max(df["Y"]) - min(df["Y"]));
```

```
a = 0.58; #Se jugo con estos valores para encontrar las proporciones correctas
b = 0.478;
c = 0.718;

#print(df.describe());
df.loc[((df["Y"] < a) & (df["P"]==0) , "Y*")] = 1;
df.loc[((df["Y"] >= a) & (df["P"]==0) , "Y*")] = 0;
df.loc[((df["Y"] < b) & (df["T"] == 0) & (df["P"] == 1) , "Y*")] = 1;
df.loc[((df["Y"] >= b) & (df["T"] == 0) & (df["P"] == 1) , "Y*")] = 0;
df.loc[((df["Y"] < c) & (df["T"] == 1) & (df["P"] == 1) , "Y*")] = 1;
df.loc[((df["Y"] >= c) & (df["T"] == 1) & (df["P"] == 1) , "Y*")] = 0;

#print(df);
#print(df.describe());

print(df.groupby(by=["P","T"]).mean());
print(df.groupby(by=["P"]).mean());

#post-test
y = df.loc[2000:,"Y*"];
X = df.loc[2000:,"T"];
X = sm.add_constant(X);
model = sm.Logit(y, X);
results = model.fit();
print(results.summary());
print(results.get_margeff().summary());
```

```
         A         X      Cl         Y     Y*   dd
P T
0 0  0.802 -0.015810  20.505  0.474444  0.809  0.0
  1  0.798 -0.029470  20.495  0.472632  0.790  0.0
1 0  0.802 -0.026115  20.505  0.446543  0.599  0.0
  1  0.798  0.033542  20.495  0.549666  0.900  1.0
     A         X    Cl    T         Y      Y*   dd
P
0  0.8 -0.022640  20.5  0.5  0.473538  0.7995  0.0
1  0.8  0.003714  20.5  0.5  0.498105  0.7495  0.5
Optimization terminated successfully.
         Current function value: 0.499249
         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.:                  0.1131
Time:                        17:39:18   Log-Likelihood:                -998.50
```

```
converged:                       True   LL-Null:                       -1125.8
Covariance Type:            nonrobust   LLR p-value:                 2.660e-57
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.4013      0.065      6.219      0.000       0.275       0.528
T              1.7959      0.124     14.531      0.000       1.554       2.038
==============================================================================
```

9. Estime el efecto promedio del tratamiento usando pre-post test con la data generada. Muestre que el efecto es equivalente usando ambos metodos.

[85]:
```
#pre-post test
df['dd'] = df['P']*df['T'];
#df["Y"] = (df["X"] + df["A"] * (1 - df["P"]) + df["A2"] * df["P"] + df["dd"])#␣
 ↪- 0.05 * df["P"];

#Normalizar Y
#df["Y"] = (df["Y"] - min(df["Y"]))/(max(df["Y"]) - min(df["Y"]));

y = df['Y*'];
X=df[["P","T","dd"]];
X = sm.add_constant(X);
model = sm.Logit(y, X);
results2 = model.fit();
print(results2.summary());
print(results2.get_margeff().summary());
```

```
Optimization terminated successfully.
         Current function value: 0.500031
         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.:                 0.06323
Time:                        17:43:27   Log-Likelihood:                -2000.1
converged:                       True   LL-Null:                       -2135.1
Covariance Type:            nonrobust   LLR p-value:                 3.082e-58
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.4435      0.080     17.944      0.000       1.286       1.601
P             -1.0422      0.103    -10.106      0.000      -1.244      -0.840
T             -0.1186      0.112     -1.061      0.289      -0.338       0.101
dd             1.9145      0.167     11.488      0.000       1.588       2.241
==============================================================================
```

```
        Logit Marginal Effects
=====================================
Dep. Variable:                      Y*
Method:                            dydx
At:                             overall
==============================================================================
                dy/dx    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
P             -0.1695      0.016    -10.552      0.000      -0.201      -0.138
T             -0.0193      0.018     -1.061      0.289      -0.055       0.016
dd             0.3114      0.026     11.970      0.000       0.260       0.362
==============================================================================
```

10. Estime el efecto ajustando los errores estandar por cluster (la variable grupo representa cada clase). Cual es la diferencia entre ambas estimaciones? Explique porque es esperable (o no) encontrar diferencias entre ambos metodos.

[80]:
```
#clustered standard errors
result3 = model.fit(cov_type="cluster", cov_kwds={'groups': df["Cl"]});
print(result3.summary());
```

```
Optimization terminated successfully.
         Current function value: 0.500031
         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.:                 0.06323
Time:                        17:29:19   Log-Likelihood:                -2000.1
converged:                       True   LL-Null:                       -2135.1
Covariance Type:              cluster   LLR p-value:                3.082e-58
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.4435      0.081     17.889      0.000       1.285       1.602
P             -1.0422      0.104    -10.037      0.000      -1.246      -0.839
T             -0.1186      0.111     -1.071      0.284      -0.336       0.099
dd             1.9145      0.147     13.032      0.000       1.627       2.202
==============================================================================
```

Se supone que es distinto el análisis x cluster que el individual por qué el cluster agrupa las observaciones y saca una medida grupal. Esta medida tiene que etar correlacionada intragrupo, pero no entre grupos. Si no se ajustase el modelo de error, podría obtenerse un error estandar menor, pero equivocado, ya que se trataria atodos los individuos como iguales pero separados y no como grupos dispares de individuos similares. Se puede notar que el tratamiento no es significativo en la aplicacion a grupos de control, por lo que puede existir correlacion no observada o pérdida de información por la creacion de grupos.

## 3 Parte 3

11. 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 sdrinlky, 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.

[114]:
```
charls = pd.read_csv("charls.csv");
charls.dropna(inplace=True);
charls.reset_index(drop=True, inplace=True);

drinklys_malos = charls[charls["drinkly"] == ".m"];
drinklys_malos.reset_index(inplace=True);
for i in range(len(drinklys_malos)):
    charls = charls[charls["inid"] != drinklys_malos['inid'][i]];
charls["drinkly"] = charls["drinkly"].astype(int);

print(charls.head(10));

charls["T"] = 0;
charls["T"][charls["child"] > 2] = 1;

charls["sdrinkly"] = charls["drinkly"];

#Calcular proporcion de drinkly
proporcion = sum(charls["drinkly"][charls["wave"] == 3][charls["T"] == 1])/
 ↪len(charls["drinkly"][charls["wave"] == 3][charls["T"] == 1]);
print("Proporcion: ", proporcion);
nueva_proporcion = proporcion/2;

aux = nueva_proporcion * len(charls["drinkly"][charls["wave"] == 3][charls["T"]␣
 ↪== 1]);
aux2 = np.random.choice([i for i in charls["sdrinkly"][charls["sdrinkly"] ==␣
 ↪1][charls["wave"] == 3][charls["T"] == 1].index],int(aux),replace=False);
for i in aux2:
    charls.loc[i,"sdrinkly"] = 0;

proporcion = sum(charls["sdrinkly"][charls["wave"] == 3][charls["T"] == 1])/
 ↪len(charls["sdrinkly"][charls["wave"] == 3][charls["T"] == 1]);
print("Nueva proporcion: ", proporcion);
print(charls);
```

```
   age      bnrps  cesd  child  dnrps  drinkly  female    hrsusu  hsize  \
0   46   0.000000   6.0      2      0        0       1  0.000000      4
1   48  58.964134   7.0      2      1        0       1  3.891820      4
2   50  60.000130   5.0      2      1        0       1  4.025352      7
```

```
3   48   0.000000   0.0      2      0        1       0  4.143135      4
4   50  58.964134   5.0      2      1        1       0  3.891820      4
5   52  60.000130   6.0      2      1        1       0  4.025352      7
6   56   0.000000   6.0      1      0        0       1  0.000000      6
7   60  60.000130   6.0      2      1        0       1  3.178054      4
8   59   0.000000   6.0      1      0        1       0  0.000000      6
9   63  60.000130   6.0      2      1        1       0  3.688879      4

   intmonth  married  nrps  retage  retired  schadj  urban  wave   wealth  \
0         7        1     0      24        0       0      0     1  -5800.0
1         7        1     1      17        0       0      0     2    100.0
2         8        1     1      10        0       0      0     3 -59970.0
3         7        1     0      22        0       4      0     1  -5800.0
4         7        1     1       0        0       4      0     2    100.0
5         8        1     1       0        0       4      0     3 -59970.0
6         8        1     0       0        0       0      0     1    350.0
7         8        1     1       0        0       0      0     3   1400.0
8         8        1     0       0        0       0      0     1    350.0
9         8        1     1       0        0       0      0     3   1400.0

   inid
0     1
1     1
2     1
3     2
4     2
5     2
6     3
7     3
8     4
9     4
C:\Users\PC\Anaconda\lib\site-packages\ipykernel_launcher.py:14:
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


Proporcion:  0.32786885245901637
Nueva proporcion:  0.16393442622950818
       age      bnrps  cesd  child  dnrps  drinkly  female    hrsusu  hsize  \
0       46   0.000000   6.0      2      0        0       1  0.000000      4
1       48  58.964134   7.0      2      1        0       1  3.891820      4
2       50  60.000130   5.0      2      1        0       1  4.025352      7
3       48   0.000000   0.0      2      0        1       0  4.143135      4
4       50  58.964134   5.0      2      1        1       0  3.891820      4
…     …         …    …     …     …       …      …        …     …
```

```
21040   55  87.628258   4.0      4      1        0       1  0.000000      4
21041   57  70.879349   2.0      4      1        1       1  0.000000      3
21042   71  87.628258   3.0      5      1        1       0  0.000000      1
21043   49  87.628258  13.0      4      1        1       1  4.025352      3
21044   60  87.628258   4.0      4      1        0       0  4.025352      3

       intmonth  …  nrps  retage  retired  schadj  urban  wave   wealth  \
0             7  …     0      24        0       0      0     1  -5800.0
1             7  …     1      17        0       0      0     2    100.0
2             8  …     1      10        0       0      0     3 -59970.0
3             7  …     0      22        0       4      0     1  -5800.0
4             7  …     1       0        0       4      0     2    100.0
…           …  …   …      …       …      …     …    …       …
21040         8  …     1       0        0       0      0     2      0.0
21041         8  …     1       0        1       0      0     3    900.0
21042         9  …     0       0        0       4      0     2    600.0
21043         8  …     1       1        0       4      0     2   5300.0
21044         8  …     1       0        0       4      0     2   5300.0

        inid  T  sdrinkly
0          1  0         0
1          1  0         0
2          1  0         0
3          2  0         1
4          2  0         1
…       … ..       …
21040  25400  1         0
21041  25400  1         1
21042  25401  1         1
21043  25402  1         1
21044  25403  1         0

[21027 rows x 21 columns]
```

12. Estime el efecto del tratamiento usando diferencias en diferencias, comparando entre los periodos 2 y 3.

[115]:
```
#Con medias
est_Y_pre = (np.mean(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪2]),np.mean(charls["sdrinkly"][charls["T"] == 1][charls["wave"] == 2]));
print(est_Y_pre);
est_Y_post = (np.mean(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪3]),np.mean(charls["sdrinkly"][charls["T"] == 1][charls["wave"] == 3]));
print(est_Y_post);
est_Y = (est_Y_pre[0] - est_Y_post[0],est_Y_pre[1] - est_Y_post[1]);
print(est_Y);
result = round(est_Y[0] - est_Y[1],4);
print("Media:",result);
```

```
#Con desviacion estandar
est_Y_pre = (np.std(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪2]),np.std(charls["sdrinkly"][charls["T"] == 1][charls["wave"] == 2]));
print(est_Y_pre);
est_Y_post = (np.std(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪3]),np.std(charls["sdrinkly"][charls["T"] == 1][charls["wave"] == 3]));
print(est_Y_post);
est_Y = (est_Y_pre[0] - est_Y_post[0],est_Y_pre[1] - est_Y_post[1]);
print(est_Y);
result2 = round(est_Y[0] - est_Y[1],4);
print("Desviacion Estandar:",result2);

#Como la media (-0.1532) no es mas de dos veces mayor a la desviacion (-0.
 ↪0946), entonces no existe una diferencia
#significativa entre las personas que reciben y que no reciben tratamiento.
```

```
(0.366023410313192, 0.31310344827586206)
(0.3700170357751278, 0.16393442622950818)
(-0.003993625461935768, 0.14916902204635388)
Media: -0.1532
(0.48171596757414387, 0.46375605543607207)
(0.48280889491734247, 0.3702160587093755)
(-0.001092927343198602, 0.09353999672669655)
Desviacion Estandar: -0.0946
```

13. Compare el efecto del tratamiento generando grupos pseudo-equivalentes, en particular entre individuos solo con 3 hijos (tratamiento) y 2 hijos (control).

[116]:
```
#Con medias
est_Y_pre = (np.mean(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪2][charls["child"] == 2]),np.mean(charls["sdrinkly"][charls["T"] ==␣
 ↪1][charls["wave"] == 2][charls["child"] == 3]));
print(est_Y_pre);
est_Y_post = (np.mean(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪3][charls["child"] == 2]),np.mean(charls["sdrinkly"][charls["T"] ==␣
 ↪1][charls["wave"] == 3][charls["child"] == 3]));
print(est_Y_post);
est_Y = (est_Y_pre[0] - est_Y_post[0],est_Y_pre[1] - est_Y_post[1]);
print(est_Y);
result = round(est_Y[0] - est_Y[1],4);
print("Media:",result);

#Con desviacion estandar
est_Y_pre = (np.std(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪2][charls["child"] == 2]),np.std(charls["sdrinkly"][charls["T"] ==␣
 ↪1][charls["wave"] == 2][charls["child"] == 3]));
```

```
print(est_Y_pre);
est_Y_post = (np.std(charls["sdrinkly"][charls["T"] == 0][charls["wave"] ==␣
 ↪3][charls["child"] == 2]),np.std(charls["sdrinkly"][charls["T"] ==␣
 ↪1][charls["wave"] == 3][charls["child"] == 3]));
print(est_Y_post);
est_Y = (est_Y_pre[0] - est_Y_post[0],est_Y_pre[1] - est_Y_post[1]);
print(est_Y);
result2 = round(est_Y[0] - est_Y[1],4);
print("Desviacion Estandar:",result2);

#Como la media (-0.1636) no es mas de dos veces mayor a la desviacion (-0.
 ↪0944), entonces no existe una diferencia
#significativa entre las personas que reciben y que no reciben tratamiento para␣
 ↪ese numero de hijos.
```

```
(0.3637137989778535, 0.3300110741971207)
(0.3693998309382925, 0.17204932472108045)
(-0.005686031960438986, 0.15796174947604028)
Media: -0.1636
(0.4810676370438526, 0.4702167214214981)
(0.48264230631084565, 0.3774233095399672)
(-0.0015746692669930673, 0.0927934118815309)
Desviacion Estandar: -0.0944
```

14. Estime el efecto anterior usando la variable married como instrumento para determinar el efecto del tratamiento en la pregunta 12. Como se interpreta el efecto en este caso?

Tomando la variable married como elemento exógeno para la asignación del tratamiento y no dejar aleatorizar como en el experimento simple original. [si se puede hacer un test de correlación de la variable con el tratamiento]. Así, se puede atacar al tratamiento en sí sin afectar directamente el outcome ni el error por componente no observado. Se hace la misma tabla de interceptor de los factores, para los datos antes y después del tratamiento. La diferencia al interpretar estos datos va con el hecho de que se pierde un gran valor de aleatoriedad, al tener un instrumento que seleccione una muestra. Además se tiene que entender que muchas de las personas no reaccionan al instrumento, o actuarán distinto debido a él. Como además, el modelo de mínimos cuadrados no especifica correctamente el modelo con variables instrumentales que correlacionen el error estándar, se hace el uso de mínimos cuadrados en dos etapas. Se asume que married está correlacionado con la variable dependiente, pero no con el error. Suponiendo que se emplea un mínimo cuadrados por etapas, debemos hacer dos ecuaciones, una representando al estimador de toma de alcohol por periodo según retardos de la variable (regresor endógeno ) y otra para el error no observado con una variable de ruido adicional. La interceptación desde acá se vuelve similar, se analiza por mco el regresor endógeno y luego el resultado al modelo inicial. Ahí, se analizará si el coeficiente del regresor será consistente.

15. Finalmente, asuma que la intervencion se implementa en todos los individuos. Genere una nueva variable de tratamiento un nueva variable llamada tdrinkly donde el efecto es una reduccion de 50% en la prevalencia de consumo de alcohol en toda la poblacion en el tercer periodo (identica a drinkly en los periodos 1 y 2). Genere una variable cdrinkly que es identica a drinkly en los periodos 1 y 2 y use la informacion de ambos periodos para predicir

el valor esperado de drinkly en el tercer periodo, estos seran los valores de cdrinkly en el periodo 3 (contrafactual). Finalmente, estime el efecto de la intervencion en toda la poblacion comparando entre tdrinkly (datos reales) versus cdrinkly contrafactual.

[117]:
```
charls["tdrinkly"] = charls["drinkly"];

proporcion = sum(charls["drinkly"][charls["wave"] == 3])/
 ↪len(charls["drinkly"][charls["wave"] == 3]);
print("Proporcion: ", proporcion);
nueva_proporcion = proporcion/2;

aux = nueva_proporcion * len(charls["drinkly"][charls["wave"] == 3]);
aux2 = np.random.choice([i for i in charls["tdrinkly"][charls["tdrinkly"] ==␣
 ↪1][charls["wave"] == 3].index],int(aux),replace=False);
for i in aux2:
    charls.loc[i,"tdrinkly"] = 0;

proporcion = sum(charls["tdrinkly"][charls["wave"] == 3])/
 ↪len(charls["tdrinkly"][charls["wave"] == 3]);
print("Nueva proporcion: ", proporcion);
```

```
Proporcion:  0.34792477302204927
Nueva proporcion:  0.17396238651102464
```

[119]:
```
charls["cdrinkly"] = charls["drinkly"];

y = charls['drinkly'][charls['wave'] <= 2];
X = charls[['age','bnrps','cesd','child','dnrps',
            'female','hrsusu','hsize','married','nrps',
            'retage','retired','schadj','urban','wealth']][charls['wave'] <= 2];
X = sm.add_constant(X);
model = sm.Logit(y, X);
results2 = model.fit();
print(results2.summary());
print(results2.get_margeff().summary());
```

```
Optimization terminated successfully.
         Current function value: 0.519034
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                drinkly   No. Observations:                14859
Model:                          Logit   Df Residuals:                    14843
Method:                           MLE   Df Model:                           15
Date:                Mon, 28 Nov 2022   Pseudo R-squ.:                 0.1826
Time:                        19:56:58   Log-Likelihood:                -7712.3
converged:                       True   LL-Null:                       -9435.6
Covariance Type:            nonrobust   LLR p-value:                     0.000
```

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 0.1822 | 0.216 | 0.845 | 0.398 | -0.241 | 0.605 |
| age | -0.0056 | 0.003 | -1.918 | 0.055 | -0.011 | 0.000 |
| bnrps | 0.0003 | 0.001 | 0.517 | 0.605 | -0.001 | 0.002 |
| cesd | -0.0025 | 0.003 | -0.732 | 0.464 | -0.009 | 0.004 |
| child | 0.0213 | 0.017 | 1.278 | 0.201 | -0.011 | 0.054 |
| dnrps | 0.0275 | 0.078 | 0.353 | 0.724 | -0.125 | 0.180 |
| female | -2.0467 | 0.046 | -44.856 | 0.000 | -2.136 | -1.957 |
| hrsusu | 0.1224 | 0.017 | 7.296 | 0.000 | 0.090 | 0.155 |
| hsize | -0.0287 | 0.011 | -2.570 | 0.010 | -0.051 | -0.007 |
| married | 0.0021 | 0.066 | 0.031 | 0.975 | -0.128 | 0.132 |
| nrps | 0.0196 | 0.056 | 0.348 | 0.728 | -0.091 | 0.130 |
| retage | 0.0115 | 0.005 | 2.404 | 0.016 | 0.002 | 0.021 |
| retired | -0.2889 | 0.079 | -3.670 | 0.000 | -0.443 | -0.135 |
| schadj | 0.0147 | 0.006 | 2.310 | 0.021 | 0.002 | 0.027 |
| urban | 0.0161 | 0.051 | 0.313 | 0.754 | -0.085 | 0.117 |
| wealth | 1.769e-07 | 4.17e-07 | 0.424 | 0.672 | -6.41e-07 | 9.94e-07 |

Logit Marginal Effects

| | |
|---|---|
| Dep. Variable: | drinkly |
| Method: | dydx |
| At: | overall |

| | dy/dx | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| age | -0.0010 | 0.001 | -1.919 | 0.055 | -0.002 | 2.08e-05 |
| bnrps | 5.417e-05 | 0.000 | 0.517 | 0.605 | -0.000 | 0.000 |
| cesd | -0.0004 | 0.001 | -0.732 | 0.464 | -0.002 | 0.001 |
| child | 0.0037 | 0.003 | 1.278 | 0.201 | -0.002 | 0.009 |
| dnrps | 0.0047 | 0.013 | 0.353 | 0.724 | -0.022 | 0.031 |
| female | -0.3532 | 0.006 | -61.511 | 0.000 | -0.364 | -0.342 |
| hrsusu | 0.0211 | 0.003 | 7.342 | 0.000 | 0.015 | 0.027 |
| hsize | -0.0050 | 0.002 | -2.572 | 0.010 | -0.009 | -0.001 |
| married | 0.0004 | 0.011 | 0.031 | 0.975 | -0.022 | 0.023 |
| nrps | 0.0034 | 0.010 | 0.348 | 0.728 | -0.016 | 0.022 |
| retage | 0.0020 | 0.001 | 2.405 | 0.016 | 0.000 | 0.004 |
| retired | -0.0499 | 0.014 | -3.676 | 0.000 | -0.076 | -0.023 |
| schadj | 0.0025 | 0.001 | 2.311 | 0.021 | 0.000 | 0.005 |
| urban | 0.0028 | 0.009 | 0.313 | 0.754 | -0.015 | 0.020 |
| wealth | 3.053e-08 | 7.2e-08 | 0.424 | 0.672 | -1.11e-07 | 1.72e-07 |