Tarea2_Rios_Arancibia

October 14, 2022

TAREA 2: César Arancibia, Francisco Ríos

[3]:
```
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
import linearmodels.panel as lmp

%matplotlib inline
```

Las variables tienen la siguiente descripcion:

- INID: identificador unico
- wave: periodo de la encuesta (1-3)
- cesd: puntaje en la escala de salud mental (0-30)
- child: numero de hijos
- drinkly: bebio alcohol en el ultimo mes (binario)
- hrsusu: horas promedio trabajo semanal
- hsize: tamano del hogar
- intmonth: mes en que fue encuestado/a (1-12)
- married: si esta casado/a (binario)
- retired: si esta pensionado/a (binario)
- schadj: años de escolaridad
- urban: zona urbana (binario)
- wealth: riqueza neta (miles RMB)
- age: edad al entrar a la encuesta (no varia entre periodos)

[208]:
```
#Leemos el archivo
charls = pd.read_csv('C:/Users/PC/Documents/GitHub/LAB-MAA/data/charls.csv');
charls.dropna(inplace=True);
charls.reset_index(drop=True, inplace=True);

#"drinkly" viene en un formato raro; esta seccion de codigo la estandariza al␣
 ↪resto de datos
print(charls.head(5));
for i in range(len(charls)):
```

```
    if charls['drinkly'][i] == '0.None':
        charls['drinkly'][i] = 0;
    elif charls['drinkly'][i] == '1.Yes':
        charls['drinkly'][i] = 1;
    else:
        charls['drinkly'][i] = 2;

#Eliminamos filas que incluyen datos no utilizables para "inid"
charls = charls.head(10056);

#Eliminamos las filas que no tienen info para "drinkly"
##########################################
drinklys_malos = charls[charls.drinkly == 2];
drinklys_malos.reset_index(inplace=True);
print(drinklys_malos);
for i in range(len(drinklys_malos)):
    charls = charls[charls.inid != drinklys_malos['inid'][i]];
print(len(charls))
##########################################

charls.inid = charls.inid.astype(str);

#Efectuamos algunas modificaciones de tipo de dato de las columnas problematicas
charls['inid'] = pd.to_numeric(charls['inid'],errors="coerce").fillna(0).
 ↪astype('int64');
charls['drinkly'] = pd.to_numeric(charls['drinkly'],errors="coerce").fillna(0).
 ↪astype('int64');
print(charls.head(5));
print(charls.info());

#Construccion de variables
X =␣
 ↪charls[['child','drinkly','female','hrsusu','hsize','married','retired','urban','schadj','w
 ↪
X = X.astype('int64');
Xm=(X.groupby(charls['inid']).transform('mean'));
Xm = Xm.astype('int64');
Xid=charls[['inid','wave','cesd','child','drinkly','female','hrsusu','hsize','married','retire
 ↪
Xid = Xid.astype('int64');
Xc=pd.DataFrame(np.c_[Xid, Xm],␣
 ↪columns=['inid','wave','cesd','child','drinkly','female','hrsusu','hsize','married','retire
 ↪
Xc = Xc.astype('int64');
print('a')
print(Xc.head(5));
print(Xc.info());
```

```
#Setear estructura de panel
Xc = Xc.set_index(["inid","wave"]);
print(Xc.describe());
```

```
   cesd  child drinkly  female  hrsusu  hsize          inid  intmonth  \
0     6      2  0.None       1     0.0      4  1.010410e+10         7
1     7      2  0.None       1    49.0      4  1.010410e+10         7
2     5      2  0.None       1    56.0      7  1.010410e+10         8
3     0      2   1.Yes       0    63.0      4  1.010410e+10         7
4     5      2   1.Yes       0    49.0      4  1.010410e+10         7

   married  retired  schadj  urban  wave   wealth  age
0        1        0       0      0     1  -5800.0   46
1        1        0       0      0     2    100.0   46
2        1        0       0      0     3 -59970.0   46
3        1        0       4      0     1  -5800.0   48
4        1        0       4      0     2    100.0   48
C:\Users\PC\Anaconda\lib\site-packages\ipykernel_launcher.py:10:
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
  # Remove the CWD from sys.path while we load stuff.
C:\Users\PC\Anaconda\lib\site-packages\ipykernel_launcher.py:12:
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
  if sys.path[0] == '':
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


   index  cesd  child drinkly  female  hrsusu  hsize          inid  intmonth  \
0   4712     1      5       2       0     0.0      2  5.605921e+10         8
1   4813     4      3       2       1    42.0      3  5.605931e+10         7
2   5878    10      3       2       0    42.0      4  5.745731e+10        10
3   6326    10      1       2       1     0.0      3  5.820230e+10         8
4   6394    10      4       2       0     0.0      2  5.820232e+10         8
5   9142    20      2       2       1     9.0      6  7.498132e+10        10
```

```
6   9227     5      3       2       0    40.0      2  7.537612e+10         7

   married  retired  schadj  urban  wave   wealth  age
0        1        1       4      0     3   1200.0   74
1        1        0       0      0     2   1600.0   56
2        1        0       4      0     2      0.0   69
3        0        0       0      1     3      0.0   50
4        1        1       4      1     2    180.0   70
5        1        0       0      0     2  52000.0   46
6        1        0       8      0     3  20900.0   53
10035
   cesd  child  drinkly  female  hrsusu  hsize         inid  intmonth  \
0     6      2        0       1     0.0      4  10104101001         7
1     7      2        0       1    49.0      4  10104101001         7
2     5      2        0       1    56.0      7  10104101001         8
3     0      2        1       0    63.0      4  10104101002         7
4     5      2        1       0    49.0      4  10104101002         7

   married  retired  schadj  urban  wave   wealth  age
0        1        0       0      0     1  -5800.0   46
1        1        0       0      0     2    100.0   46
2        1        0       0      0     3 -59970.0   46
3        1        0       4      0     1  -5800.0   48
4        1        0       4      0     2    100.0   48
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10035 entries, 0 to 10055
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   cesd      10035 non-null  int64
 1   child     10035 non-null  int64
 2   drinkly   10035 non-null  int64
 3   female    10035 non-null  int64
 4   hrsusu    10035 non-null  float64
 5   hsize     10035 non-null  int64
 6   inid      10035 non-null  int64
 7   intmonth  10035 non-null  int64
 8   married   10035 non-null  int64
 9   retired   10035 non-null  int64
 10  schadj    10035 non-null  int64
 11  urban     10035 non-null  int64
 12  wave      10035 non-null  int64
 13  wealth    10035 non-null  float64
 14  age       10035 non-null  int64
dtypes: float64(2), int64(13)
memory usage: 1.2 MB
None
a
```

```
          inid  wave  cesd  child  drinkly  female  hrsusu  hsize  married  \
0  10104101001     1     6      2        0       1       0      4        1
1  10104101001     2     7      2        0       1      49      4        1
2  10104101001     3     5      2        0       1      56      7        1
3  10104101002     1     0      2        1       0      63      4        1
4  10104101002     2     5      2        1       0      49      4        1

   retired  …  mfemale  mhrsusu  mhsize  mmarried  mretired  murban  \
0        0  …        1       35       5         1         0       0
1        0  …        1       35       5         1         0       0
2        0  …        1       35       5         1         0       0
3        0  …        0       56       5         1         0       0
4        0  …        0       56       5         1         0       0

   mschadj  mwealth  mage  mintmonth
0        0   -21890    46          7
1        0   -21890    46          7
2        0   -21890    46          7
3        4   -21890    48          7
4        4   -21890    48          7

[5 rows x 27 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10035 entries, 0 to 10034
Data columns (total 27 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   inid       10035 non-null  int64
 1   wave       10035 non-null  int64
 2   cesd       10035 non-null  int64
 3   child      10035 non-null  int64
 4   drinkly    10035 non-null  int64
 5   female     10035 non-null  int64
 6   hrsusu     10035 non-null  int64
 7   hsize      10035 non-null  int64
 8   married    10035 non-null  int64
 9   retired    10035 non-null  int64
 10  urban      10035 non-null  int64
 11  schadj     10035 non-null  int64
 12  wealth     10035 non-null  int64
 13  age        10035 non-null  int64
 14  intmonth   10035 non-null  int64
 15  mchild     10035 non-null  int64
 16  mdrinkly   10035 non-null  int64
 17  mfemale    10035 non-null  int64
 18  mhrsusu    10035 non-null  int64
 19  mhsize     10035 non-null  int64
 20  mmarried   10035 non-null  int64
```

```
 21  mretired   10035 non-null  int64
 22  murban     10035 non-null  int64
 23  mschadj    10035 non-null  int64
 24  mwealth    10035 non-null  int64
 25  mage       10035 non-null  int64
 26  mintmonth  10035 non-null  int64
dtypes: int64(27)
memory usage: 2.1 MB
None
```

	cesd	child	drinkly	female	hrsusu	\
count	10035.000000	10035.000000	10035.000000	10035.000000	10035.000000	
mean	8.866069	2.768411	0.324165	0.542302	27.952865	
std	6.290205	1.436302	0.468085	0.498232	27.244793	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	4.000000	2.000000	0.000000	0.000000	0.000000	
50%	8.000000	2.000000	0.000000	1.000000	24.000000	
75%	13.000000	3.000000	1.000000	1.000000	49.000000	
max	30.000000	10.000000	1.000000	1.000000	168.000000	

	hsize	married	retired	urban	schadj	\
count	10035.000000	10035.000000	10035.000000	10035.000000	10035.000000	
mean	3.653413	0.858196	0.268759	0.315695	4.095067	
std	1.785335	0.348866	0.443337	0.464815	3.604436	
min	1.000000	0.000000	0.000000	0.000000	0.000000	
25%	2.000000	1.000000	0.000000	0.000000	0.000000	
50%	3.000000	1.000000	0.000000	0.000000	4.000000	
75%	5.000000	1.000000	1.000000	1.000000	4.000000	
max	13.000000	1.000000	1.000000	1.000000	16.000000	

	…	mfemale	mhrsusu	mhsize	mmarried	\
count	…	10035.000000	10035.000000	10035.000000	10035.000000	
mean	…	0.542302	27.689686	3.374589	0.833483	
std	…	0.498232	21.201417	1.454681	0.372563	
min	…	0.000000	0.000000	1.000000	0.000000	
25%	…	0.000000	8.000000	2.000000	1.000000	
50%	…	1.000000	28.000000	3.000000	1.000000	
75%	…	1.000000	43.000000	4.000000	1.000000	
max	…	1.000000	119.000000	10.000000	1.000000	

	mretired	murban	mschadj	mwealth	mage	\
count	10035.000000	10035.000000	10035.000000	1.003500e+04	10035.000000	
mean	0.139312	0.315695	4.095067	1.020623e+04	58.222422	
std	0.346289	0.464815	3.604436	6.285327e+04	9.233897	
min	0.000000	0.000000	0.000000	-3.250000e+05	16.000000	
25%	0.000000	0.000000	0.000000	8.300000e+01	51.000000	
50%	0.000000	0.000000	4.000000	1.070000e+03	58.000000	
75%	0.000000	1.000000	4.000000	8.616000e+03	64.000000	
max	1.000000	1.000000	16.000000	2.672550e+06	89.000000	

```
          mintmonth
count  10035.000000
mean       7.301644
std        0.688318
min        5.000000
25%        7.000000
50%        7.000000
75%        8.000000
max       10.000000

[8 rows x 25 columns]
```

[209]:
```
#Verificacion de que "inid" se repite solo 3 veces
print(charls["inid"].value_counts());
print('----');

#Verificacion de que "drinkly" solo toma valores 0 y 1
print(charls["drinkly"].value_counts());
print("----");
```

```
56302123001    3
31106113001    3
10206109001    3
51606216001    3
75376101002    3
              ..
60440126001    3
54054206002    3
10206313002    3
94004116002    3
56059316001    3
Name: inid, Length: 3345, dtype: int64
----
0    6782
1    3253
Name: drinkly, dtype: int64
----
```

[210]:
```
#Visualizacion de las variables
print(Xc.head(9));
print(Xc.info());
```

```
                 cesd  child  drinkly  female  hrsusu  hsize  married  \
inid        wave
10104101001 1       6      2        0       1       0      4        1
            2       7      2        0       1      49      4        1
            3       5      2        0       1      56      7        1
```

```
10104101002 1        0      2        1       0      63      4        1
            2        5      2        1       0      49      4        1
            3        6      2        1       0      56      7        1
10104102001 1        6      1        0       1       0      6        1
            2        7      2        0       1      35      6        1
            3        6      2        0       1      24      4        1

                  retired  urban  schadj  …  mfemale  mhrsusu  mhsize  \
inid        wave                          …
10104101001 1           0      0       0  …        1       35       5
            2           0      0       0  …        1       35       5
            3           0      0       0  …        1       35       5
10104101002 1           0      0       4  …        0       56       5
            2           0      0       4  …        0       56       5
            3           0      0       4  …        0       56       5
10104102001 1           0      0       0  …        1       19       5
            2           0      0       0  …        1       19       5
            3           0      0       0  …        1       19       5

                  mmarried  mretired  murban  mschadj  mwealth  mage  \
inid        wave
10104101001 1            1         0       0        0   -21890    46
            2            1         0       0        0   -21890    46
            3            1         0       0        0   -21890    46
10104101002 1            1         0       0        4   -21890    48
            2            1         0       0        4   -21890    48
            3            1         0       0        4   -21890    48
10104102001 1            1         0       0        0      583    57
            2            1         0       0        0      583    57
            3            1         0       0        0      583    57

                  mintmonth
inid        wave
10104101001 1             7
            2             7
            3             7
10104101002 1             7
            2             7
            3             7
10104102001 1             7
            2             7
            3             7

[9 rows x 25 columns]
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 10035 entries, (10104101001, 1) to (94004308001, 3)
Data columns (total 25 columns):
 #   Column     Non-Null Count  Dtype
```

```
---  ------     --------------  -----
 0   cesd       10035 non-null  int64
 1   child      10035 non-null  int64
 2   drinkly    10035 non-null  int64
 3   female     10035 non-null  int64
 4   hrsusu     10035 non-null  int64
 5   hsize      10035 non-null  int64
 6   married    10035 non-null  int64
 7   retired    10035 non-null  int64
 8   urban      10035 non-null  int64
 9   schadj     10035 non-null  int64
 10  wealth     10035 non-null  int64
 11  age        10035 non-null  int64
 12  intmonth   10035 non-null  int64
 13  mchild     10035 non-null  int64
 14  mdrinkly   10035 non-null  int64
 15  mfemale    10035 non-null  int64
 16  mhrsusu    10035 non-null  int64
 17  mhsize     10035 non-null  int64
 18  mmarried   10035 non-null  int64
 19  mretired   10035 non-null  int64
 20  murban     10035 non-null  int64
 21  mschadj    10035 non-null  int64
 22  mwealth    10035 non-null  int64
 23  mage       10035 non-null  int64
 24  mintmonth  10035 non-null  int64
dtypes: int64(25)
memory usage: 2.0 MB
None
```

0.1 Graficos

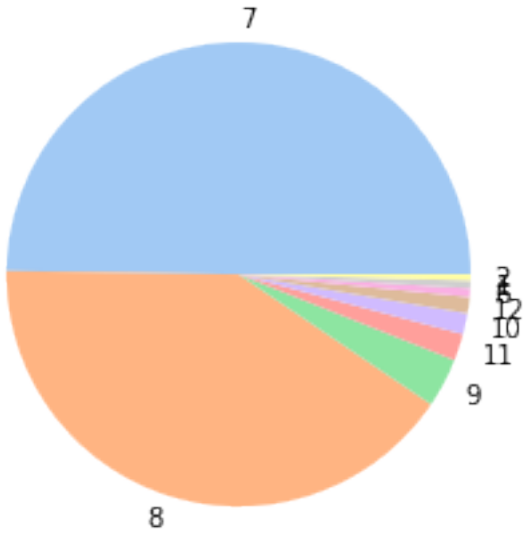
[211]:
```
import seaborn
col = seaborn.color_palette('pastel');

#Visualizacion de la variable "intmonth"
var = 'intmonth';

print(charls[var].value_counts());
graf = charls[var].value_counts().to_frame().reset_index();
graf.columns = [var, 'rep'];
plt.pie(graf.rep, labels=graf.intmonth, colors=col);
plt.show();

#Se descubre que la gran mayoría de datos corresponden al mes 7 y al mes 8,␣
 ↪pero existen suficientes datos
#de otros meses como para impedir que se consideren atípicos.
```

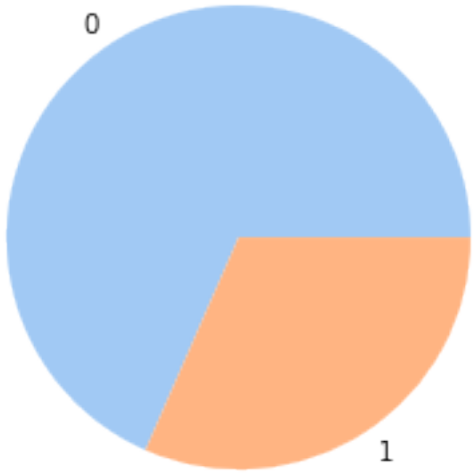
```
7     4994
8     4093
9      344
11     188
10     147
12     113
6       60
1       54
2       42
Name: intmonth, dtype: int64
```



[212]:
```
#Visualizacion de la variable "urban"
var = 'urban';

print(charls[var].value_counts());
graf = charls[var].value_counts().to_frame().reset_index();
graf.columns = [var, 'rep'];
plt.pie(graf.rep, labels=graf.urban, colors=col);
plt.show();
```

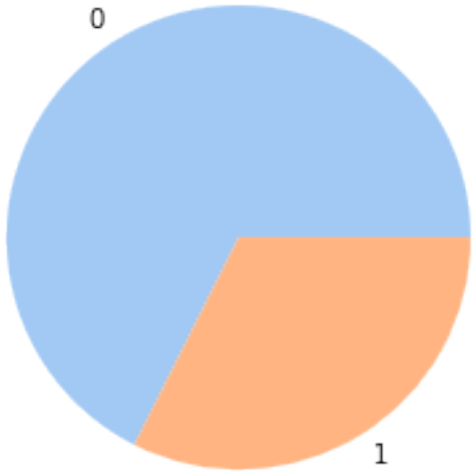
```
0    6867
1    3168
Name: urban, dtype: int64
```



[188]:
```
#Verificamos que "drinkly" ahora sea binaria
var = 'drinkly';

print(charls[var].value_counts());
graf = charls[var].value_counts().to_frame().reset_index();
graf.columns = [var, 'rep'];
plt.pie(graf.rep, labels=graf.drinkly, colors=col);
plt.show();
```

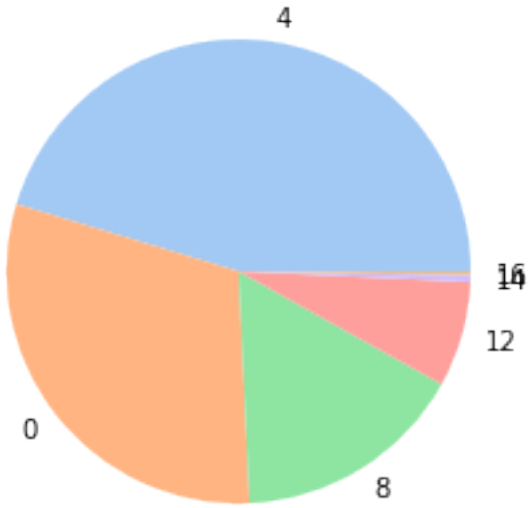
```
0    6782
1    3253
Name: drinkly, dtype: int64
```



[189]:
```
#Visualizacion de la variable "schadj"
var = 'schadj';

print(charls[var].value_counts());
graf = charls[var].value_counts().to_frame().reset_index();
graf.columns = [var, 'rep'];
plt.pie(graf.rep, labels=graf.schadj, colors=col);
plt.show();
```

```
4     4548
0     3054
8     1623
12     738
14      45
16      27
Name: schadj, dtype: int64
```



0.2 Pooled OLS

[242]:
```
#Se establece la variable dependiente
y=Xc['cesd'];

#Se establecen las variables independientes a considerar en este modelo
X=Xc[['child','drinkly','female','hrsusu','hsize','married','retired','urban','schadj','wealth
 ↪
#X=Xc[['child','drinkly','hrsusu','hsize','married','retired','wealth']];
X=sm.add_constant(X);

model = sm.OLS(y, X);
results = model.fit();
print(results.summary());

#Graficar residuos para intmonth y así demostrar heterocedasticidad
print(sm.graphics.plot_regress_exog(results, 'intmonth', fig=plt.
 ↪figure(figsize=(10,8))));

#Se concluye que intmonth presenta heterocedasticidad.
```

OLS Regression Results

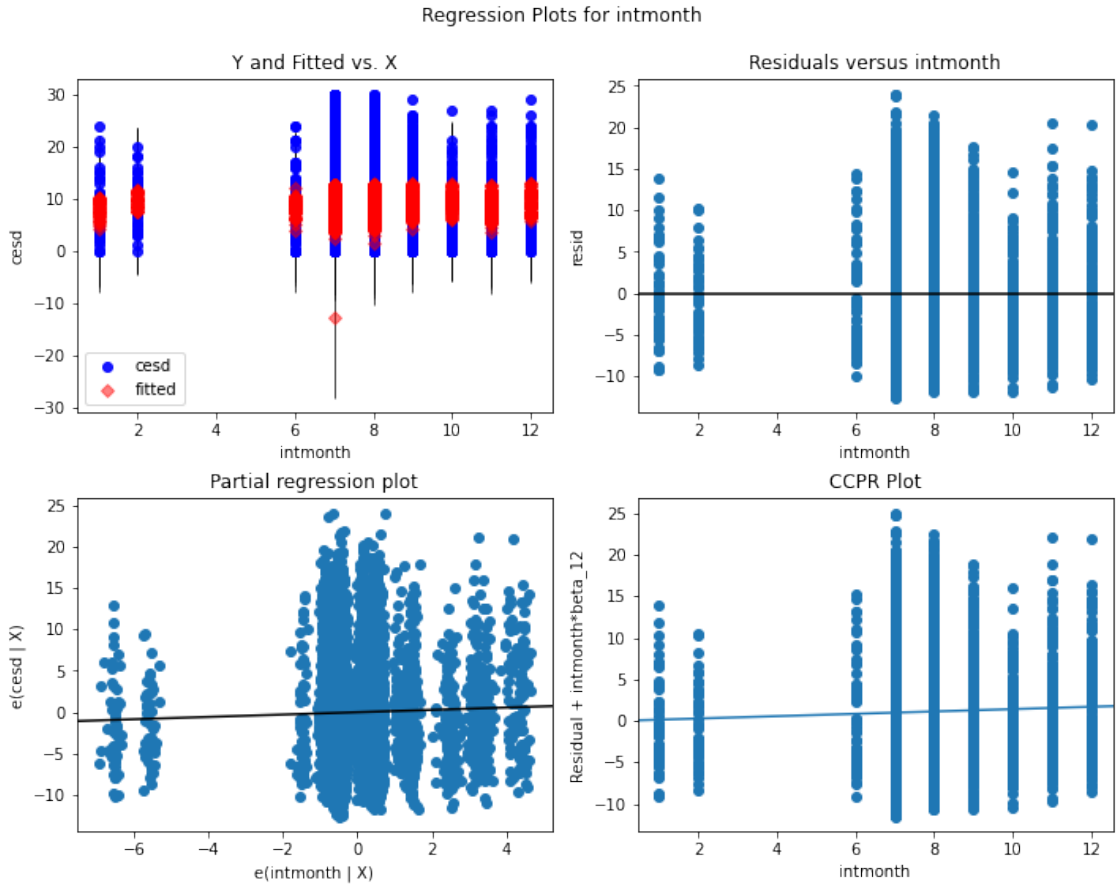
Dep. Variable:	cesd	R-squared:	0.073
Model:	OLS	Adj. R-squared:	0.072
Method:	Least Squares	F-statistic:	65.53
Date:	Wed, 05 Oct 2022	Prob (F-statistic):	2.21e-154

Time:	19:49:24	Log-Likelihood:	-32314.
No. Observations:	10035	AIC:	6.465e+04
Df Residuals:	10022	BIC:	6.475e+04
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	10.0525	0.732	13.731	0.000	8.617	11.488
child	0.0891	0.049	1.829	0.067	-0.006	0.185
drinkly	-0.0019	0.144	-0.013	0.990	-0.284	0.280
female	1.5318	0.143	10.701	0.000	1.251	1.812
hrsusu	0.0058	0.003	2.011	0.044	0.000	0.011
hsize	-0.0785	0.035	-2.226	0.026	-0.148	-0.009
married	-1.3933	0.187	-7.467	0.000	-1.759	-1.028
retired	0.4884	0.182	2.679	0.007	0.131	0.846
urban	-1.8529	0.139	-13.368	0.000	-2.125	-1.581
schadj	-0.1954	0.019	-10.367	0.000	-0.232	-0.158
wealth	-2.322e-06	6.11e-07	-3.798	0.000	-3.52e-06	-1.12e-06
age	-0.0127	0.008	-1.521	0.128	-0.029	0.004
intmonth	0.1416	0.056	2.543	0.011	0.032	0.251

Omnibus:	664.468	Durbin-Watson:	1.337
Prob(Omnibus):	0.000	Jarque-Bera (JB):	804.243
Skew:	0.692	Prob(JB):	2.30e-175
Kurtosis:	3.094	Cond. No.	1.22e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.22e+06. This might indicate that there are strong multicollinearity or other numerical problems.
Figure(720x576)



[244]:
```
#Se ejecuta Pooled OLS
model=lmp.PooledOLS(y,X);
PooledOLS=model.fit(cov_type="robust");
print(PooledOLS);
```

PooledOLS Estimation Summary

Dep. Variable:	cesd	R-squared:	0.0728
Estimator:	PooledOLS	R-squared (Between):	0.1097
No. Observations:	10035	R-squared (Within):	-0.0003
Date:	Wed, Oct 05 2022	R-squared (Overall):	0.0728
Time:	19:50:46	Log-likelihood	-3.231e+04
Cov. Estimator:	Robust		
		F-statistic:	65.532
Entities:	3345	P-value	0.0000
Avg Obs:	3.0000	Distribution:	F(12,10022)
Min Obs:	3.0000		
Max Obs:	3.0000	F-statistic (robust):	64.113
		P-value	0.0000

Time periods:	3	Distribution:	F(12,10022)
Avg Obs:	3345.0		
Min Obs:	3345.0		
Max Obs:	3345.0		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	10.052	0.7381	13.619	0.0000	8.6056	11.499
child	0.0891	0.0498	1.7877	0.0738	-0.0086	0.1867
drinkly	-0.0019	0.1424	-0.0130	0.9896	-0.2810	0.2773
female	1.5318	0.1436	10.665	0.0000	1.2502	1.8133
hrsusu	0.0058	0.0028	2.0498	0.0404	0.0003	0.0113
hsize	-0.0785	0.0346	-2.2701	0.0232	-0.1463	-0.0107
married	-1.3933	0.2022	-6.8903	0.0000	-1.7896	-0.9969
retired	0.4884	0.1859	2.6277	0.0086	0.1241	0.8528
urban	-1.8529	0.1332	-13.915	0.0000	-2.1139	-1.5919
schadj	-0.1954	0.0183	-10.666	0.0000	-0.2313	-0.1595
wealth	-2.322e-06	1.96e-06	-1.1851	0.2360	-6.164e-06	1.519e-06
age	-0.0127	0.0086	-1.4844	0.1377	-0.0295	0.0041
intmonth	0.1416	0.0542	2.6139	0.0090	0.0354	0.2477

0.3 First differences

[246]:
```
#Se retiran variables problematicas (las que son fijas en el tiempo) y se ejecuta
X=Xc[['child','drinkly','hrsusu','married','retired','hsize','wealth']];
model=lmp.FirstDifferenceOLS(y,X)
fd=model.fit(cov_type="robust")
print(fd);
```

FirstDifferenceOLS Estimation Summary

Dep. Variable:	cesd	R-squared:	0.0031
Estimator:	FirstDifferenceOLS	R-squared (Between):	-0.2010
No. Observations:	6690	R-squared (Within):	0.0034
Date:	Wed, Oct 05 2022	R-squared (Overall):	-0.1780
Time:	19:55:57	Log-likelihood	-2.163e+04
Cov. Estimator:	Robust		
		F-statistic:	3.0065
Entities:	3345	P-value	0.0037
Avg Obs:	3.0000	Distribution:	F(7,6683)
Min Obs:	3.0000		
Max Obs:	3.0000	F-statistic (robust):	2.6024
		P-value	0.0111
Time periods:	3	Distribution:	F(7,6683)

```
Avg Obs:                       3345.0
Min Obs:                       3345.0
Max Obs:                       3345.0

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
child          0.1344     0.1010     1.3309     0.1833     -0.0636      0.3324
drinkly        0.0358     0.1848     0.1938     0.8464     -0.3264      0.3980
hrsusu        -0.0012     0.0029    -0.4113     0.6809     -0.0069      0.0045
married       -1.2037     0.5580    -2.1571     0.0310     -2.2975     -0.1098
retired        0.1851     0.2038     0.9084     0.3637     -0.2144      0.5847
hsize         -0.1386     0.0454    -3.0506     0.0023     -0.2276     -0.0495
wealth      2.173e-07  6.248e-07     0.3478     0.7280  -1.007e-06   1.442e-06
==============================================================================
```

0.4 Fixed Effects

[279]:
```
#Se retiran variables problematicas y se ejecuta
X=Xc[['child','drinkly','hrsusu','married','retired','hsize','wealth']];
X=sm.add_constant(X)
model=lmp.PanelOLS(y,X, entity_effects=True)
fe=model.fit(cov_type="robust")
print(fe)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                   cesd   R-squared:                        0.0039
Estimator:                   PanelOLS   R-squared (Between):              0.0128
No. Observations:               10035   R-squared (Within):               0.0039
Date:                Wed, Oct 05 2022   R-squared (Overall):              0.0098
Time:                        20:46:44   Log-likelihood                 -2.72e+04
Cov. Estimator:                Robust
                                        F-statistic:                      3.7479
Entities:                        3345   P-value                           0.0005
Avg Obs:                       3.0000   Distribution:                  F(7,6683)
Min Obs:                       3.0000
Max Obs:                       3.0000   F-statistic (robust):             3.1735
                                        P-value                           0.0024
Time periods:                       3   Distribution:                  F(7,6683)
Avg Obs:                       3345.0
Min Obs:                       3345.0
Max Obs:                       3345.0

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
```

```
------------------------------------------------------------------------------
const          9.7600     0.5457     17.886     0.0000      8.6903      10.830
child          0.1512     0.0958     1.5787     0.1144     -0.0365      0.3390
drinkly        0.2126     0.1886     1.1275     0.2596     -0.1570      0.5823
hrsusu        -0.0003     0.0029    -0.1172     0.9067     -0.0060      0.0054
married       -1.1847     0.5062    -2.3402     0.0193     -2.1771     -0.1923
retired        0.3501     0.2022     1.7313     0.0834     -0.0463      0.7464
hsize         -0.1215     0.0439    -2.7657     0.0057     -0.2077     -0.0354
wealth     -5.151e-07  8.421e-07    -0.6117     0.5407  -2.166e-06   1.136e-06
==============================================================================

F-test for Poolability: 3.8479
P-value: 0.0000
Distribution: F(3344,6683)

Included effects: Entity
```

0.5 Random Effects

[258]:
```
#Se ejecuta modelo de random effects con fin de comparar con fixed effects
#X=Xc[['child','drinkly','hrsusu','married','retired','hsize','wealth']];
model=lmp.RandomEffects(y,X)
re=model.fit(cov_type="robust")
print(re)
```

```
                        RandomEffects Estimation Summary
================================================================================
Dep. Variable:                   cesd   R-squared:                        0.0104
Estimator:                RandomEffects   R-squared (Between):              0.0286
No. Observations:                10035   R-squared (Within):               0.0009
Date:                Wed, Oct 05 2022   R-squared (Overall):              0.0193
Time:                        20:00:42   Log-likelihood                -2.926e+04
Cov. Estimator:                Robust
                                        F-statistic:                      14.989
Entities:                        3345   P-value                           0.0000
Avg Obs:                       3.0000   Distribution:                 F(7,10027)
Min Obs:                       3.0000
Max Obs:                       3.0000   F-statistic (robust):             12.794
                                        P-value                           0.0000
Time periods:                       3   Distribution:                 F(7,10027)
Avg Obs:                       3345.0
Min Obs:                       3345.0
Max Obs:                       3345.0

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
```

```
const          9.9409     0.3217     30.903     0.0000      9.3103      10.571
child          0.2308     0.0548     4.2156     0.0000      0.1235      0.3381
drinkly       -0.4197     0.1429    -2.9371     0.0033     -0.6999     -0.1396
hrsusu         0.0017     0.0026     0.6453     0.5188     -0.0034      0.0067
married       -1.5952     0.2489    -6.4098     0.0000     -2.0830     -1.1073
retired        0.0874     0.1727     0.5063     0.6127     -0.2511      0.4260
hsize         -0.0720     0.0351    -2.0545     0.0400     -0.1408     -0.0033
wealth     -1.535e-06  1.122e-06    -1.3682     0.1713  -3.734e-06   6.642e-07
==============================================================================
```

[280]:
```
#Se ejecuta modelo de random effects con fin de comparar con Correlated Random Effects
X=Xc[['child','drinkly','female','hrsusu','hsize','married','retired','urban','schadj','wealth
X=sm.add_constant(X)
model=lmp.RandomEffects(y,X)
reALT=model.fit(cov_type="robust")
print(reALT)
```

```
                        RandomEffects Estimation Summary
================================================================================
Dep. Variable:                    cesd   R-squared:                        0.0404
Estimator:                RandomEffects   R-squared (Between):              0.1066
No. Observations:                 10035   R-squared (Within):               0.0032
Date:                Wed, Oct 05 2022   R-squared (Overall):              0.0719
Time:                        20:46:47   Log-likelihood                -2.924e+04
Cov. Estimator:                Robust
                                        F-statistic:                      35.120
Entities:                        3345   P-value                           0.0000
Avg Obs:                       3.0000   Distribution:                F(12,10022)
Min Obs:                       3.0000
Max Obs:                       3.0000   F-statistic (robust):             35.388
                                        P-value                           0.0000
Time periods:                       3   Distribution:                F(12,10022)
Avg Obs:                       3345.0
Min Obs:                       3345.0
Max Obs:                       3345.0

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const          11.110     0.8536     13.015     0.0000      9.4367      12.783
child          0.1133     0.0591     1.9163     0.0554     -0.0026      0.2291
drinkly        0.0981     0.1498     0.6547     0.5127     -0.1956      0.3918
female         1.5418     0.1909     8.0768     0.0000      1.1676      1.9160
hrsusu         0.0022     0.0026     0.8738     0.3822     -0.0028      0.0073
hsize         -0.1006     0.0350    -2.8739     0.0041     -0.1692     -0.0320
```

```
married       -1.3449     0.2502    -5.3749     0.0000     -1.8353     -0.8544
retired        0.3962     0.1746     2.2696     0.0233      0.0540      0.7384
urban         -1.8231     0.1821    -10.009     0.0000     -2.1802     -1.4661
schadj        -0.2011     0.0246    -8.1699     0.0000     -0.2493     -0.1528
wealth     -1.228e-06  1.061e-06    -1.1572     0.2472  -3.307e-06   8.518e-07
age           -0.0166     0.0110    -1.4987     0.1340     -0.0382      0.0051
intmonth       0.0398     0.0462     0.8606     0.3895     -0.0508      0.1304
==============================================================================
```

[259]:
```
re.variance_decomposition
```

[259]:
```
Effects                  18.639661
Residual                 19.861379
Percent due to Effects    0.484134
Name: Variance Decomposition, dtype: float64
```

0.6 Model comparison

[263]:
```
print(lmp.compare({"FE": fe, "RE": re, "Pooled": PooledOLS}))
```

```
                            Model Comparison
=============================================================================
                                    FE                RE         Pooled
-----------------------------------------------------------------------------
Dep. Variable                     cesd              cesd           cesd
Estimator                     PanelOLS     RandomEffects      PooledOLS
No. Observations                 10035             10035          10035
Cov. Est.                       Robust            Robust         Robust
R-squared                       0.0039            0.0104         0.0728
R-Squared (Within)              0.0039            0.0009        -0.0003
R-Squared (Between)             0.0128            0.0286         0.1097
R-Squared (Overall)             0.0098            0.0193         0.0728
F-statistic                     3.7479            14.989         65.532
P-value (F-stat)                0.0005            0.0000         0.0000
====================     ============   ===============   ============
const                           9.7600            9.9409         10.052
                              (17.886)          (30.903)       (13.619)
child                           0.1512            0.2308         0.0891
                              (1.5787)          (4.2156)       (1.7877)
drinkly                         0.2126           -0.4197        -0.0019
                              (1.1275)         (-2.9371)      (-0.0130)
hrsusu                         -0.0003            0.0017         0.0058
                             (-0.1172)          (0.6453)       (2.0498)
married                        -1.1847           -1.5952        -1.3933
                             (-2.3402)         (-6.4098)      (-6.8903)
retired                         0.3501            0.0874         0.4884
                              (1.7313)          (0.5063)       (2.6277)
```

```
hsize                        -0.1215          -0.0720          -0.0785
                            (-2.7657)        (-2.0545)        (-2.2701)
wealth                     -5.151e-07       -1.535e-06       -2.322e-06
                            (-0.6117)        (-1.3682)        (-1.1851)
female                                                          1.5318
                                                               (10.665)
urban                                                          -1.8529
                                                              (-13.915)
schadj                                                         -0.1954
                                                              (-10.666)
age                                                            -0.0127
                                                              (-1.4844)
intmonth                                                        0.1416
                                                               (2.6139)
====================== ============= ================ ==============
Effects                         Entity
------------------------------------------------------------------------

T-stats reported in parentheses
```

[283]:
```
#Se ejecuta test de Hausman
import numpy.linalg as la
from scipy import stats

def hausman(fe, re):
    diff = fe.params-re.params
    psi = fe.cov - re.cov
    dof = diff.size -1
    W = diff.dot(la.inv(psi)).dot(diff)
    pval = stats.chi2.sf(W, dof)
    return W, dof, pval
```

[284]:
```
htest = hausman(fe, re)
print("Hausman Test: chi-2 = {0}, df = {1}, p-value = {2}".format(htest[0],␣
 ↪htest[1], htest[2]))
#Se concluye que no se rechaza hipotesis nula, y por lo tanto Fixed Effects es␣
 ↪mejor modelo
```

```
Hausman Test: chi-2 = 41.63022350740858, df = 7, p-value = 6.125793499234899e-07
```

0.7 Correlated Random Effects

[281]:
```
#Se agregan promedios y se ejecuta modelo CRE
X=Xc[['child','drinkly','female','hrsusu','hsize','married','retired','urban','schadj','wealth
 ↪
X=sm.add_constant(X);
model=lmp.RandomEffects(y,X);
```

```
cre=model.fit(cov_type="robust");
print(cre);
```

RandomEffects Estimation Summary

Dep. Variable:	cesd	R-squared:	0.0436
Estimator:	RandomEffects	R-squared (Between):	0.1136
No. Observations:	10035	R-squared (Within):	0.0038
Date:	Wed, Oct 05 2022	R-squared (Overall):	0.0768
Time:	20:46:51	Log-likelihood	-2.925e+04
Cov. Estimator:	Robust		
		F-statistic:	22.800
Entities:	3345	P-value	0.0000
Avg Obs:	3.0000	Distribution:	F(20,10014)
Min Obs:	3.0000		
Max Obs:	3.0000	F-statistic (robust):	22.618
		P-value	0.0000
Time periods:	3	Distribution:	F(20,10014)
Avg Obs:	3345.0		
Min Obs:	3345.0		
Max Obs:	3345.0		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	8.1897	1.2876	6.3605	0.0000	5.6658	10.714
child	0.1757	0.0984	1.7855	0.0742	-0.0172	0.3687
drinkly	0.3322	0.1750	1.8977	0.0578	-0.0109	0.6753
female	1.4688	0.2004	7.3276	0.0000	1.0759	1.8617
hrsusu	-2.106e-05	0.0029	-0.0073	0.9942	-0.0057	0.0056
hsize	-0.1148	0.0442	-2.5938	0.0095	-0.2015	-0.0280
married	-1.1274	0.4277	-2.6360	0.0084	-1.9657	-0.2890
retired	0.4088	0.1888	2.1659	0.0303	0.0388	0.7789
urban	-1.9030	0.1888	-10.082	0.0000	-2.2730	-1.5331
schadj	-0.1828	0.0256	-7.1453	0.0000	-0.2329	-0.1326
wealth	-5.209e-07	1.424e-06	-0.3657	0.7146	-3.313e-06	2.271e-06
age	-0.0058	0.0122	-0.4783	0.6324	-0.0298	0.0181
intmonth	0.0047	0.0489	0.0959	0.9236	-0.0911	0.1005
mchild	-0.1306	0.1204	-1.0847	0.2781	-0.3666	0.1054
mdrinkly	-0.7516	0.2583	-2.9103	0.0036	-1.2579	-0.2454
mhrsusu	0.0109	0.0055	1.9632	0.0497	1.646e-05	0.0218
mhsize	0.0685	0.0756	0.9063	0.3648	-0.0797	0.2168
mmarried	-0.3083	0.4483	-0.6876	0.4917	-1.1871	0.5706
mretired	0.1761	0.3202	0.5499	0.5824	-0.4516	0.8038
mwealth	-4.658e-06	3.597e-06	-1.2952	0.1953	-1.171e-05	2.392e-06
mintmonth	0.3387	0.1331	2.5448	0.0109	0.0778	0.5996

```
==============================================================================
```

[282]:
```
#Se muestra comparacion
print(lmp.compare({"FE": fe, "RE": reALT, "CRE": cre}))
```

```
                          Model Comparison
==============================================================================
                                    FE                RE                CRE
------------------------------------------------------------------------------
Dep. Variable                     cesd              cesd              cesd
Estimator                     PanelOLS      RandomEffects     RandomEffects
No. Observations                 10035             10035             10035
Cov. Est.                       Robust            Robust            Robust
R-squared                       0.0039            0.0404            0.0436
R-Squared (Within)              0.0039            0.0032            0.0038
R-Squared (Between)             0.0128            0.1066            0.1136
R-Squared (Overall)             0.0098            0.0719            0.0768
F-statistic                     3.7479            35.120            22.800
P-value (F-stat)                0.0005            0.0000            0.0000
====================     ============   ===============   ===============
const                           9.7600            11.110            8.1897
                              (17.886)          (13.015)          (6.3605)
child                           0.1512            0.1133            0.1757
                              (1.5787)          (1.9163)          (1.7855)
drinkly                         0.2126            0.0981            0.3322
                              (1.1275)          (0.6547)          (1.8977)
hrsusu                         -0.0003            0.0022        -2.106e-05
                             (-0.1172)          (0.8738)         (-0.0073)
married                        -1.1847           -1.3449           -1.1274
                             (-2.3402)         (-5.3749)         (-2.6360)
retired                         0.3501            0.3962            0.4088
                              (1.7313)          (2.2696)          (2.1659)
hsize                          -0.1215           -0.1006           -0.1148
                             (-2.7657)         (-2.8739)         (-2.5938)
wealth                     -5.151e-07        -1.228e-06        -5.209e-07
                             (-0.6117)         (-1.1572)         (-0.3657)
female                                            1.5418            1.4688
                                                (8.0768)          (7.3276)
urban                                            -1.8231           -1.9030
                                               (-10.009)         (-10.082)
schadj                                           -0.2011           -0.1828
                                               (-8.1699)         (-7.1453)
age                                              -0.0166           -0.0058
                                               (-1.4987)         (-0.4783)
intmonth                                          0.0398            0.0047
                                                (0.8606)          (0.0959)
mchild                                                              -0.1306
                                                                  (-1.0847)
```

```
mdrinkly                                                          -0.7516
                                                                 (-2.9103)
mhrsusu                                                            0.0109
                                                                  (1.9632)
mhsize                                                             0.0685
                                                                  (0.9063)
mmarried                                                          -0.3083
                                                                 (-0.6876)
mretired                                                           0.1761
                                                                  (0.5499)
mwealth                                                         -4.658e-06
                                                                  (-1.2952)
mintmonth                                                          0.3387
                                                                  (2.5448)
====================== ============= ================ =================
Effects                          Entity
-------------------------------------------------------------------------

T-stats reported in parentheses
```

Tarea 2

Instrucciones

Los resultados de los ejericicios propuestos se deben entregar como un notebook por correo electronico a *juan.caro@uni.lu* el dia 3/10 hasta las 21:00.

Es importante considerar que el código debe poder ejecutarse en cualquier computadora con la data original del repositorio. Recordar la convencion para el nombre de archivo ademas de incluir en su documento titulos y encabezados por seccion. La data a utilizar es **charls.csv**.

Las variables tienen la siguiente descripcion:

- INID: identificador unico
- wave: periodo de la encuesta (1-3)
- cesd: puntaje en la escala de salud mental (0-30)
- child: numero de hijos
- drinkly: bebio alcohol en el ultimo mes (binario)
- hrsusu: horas promedio trabajo semanal
- hsize: tamano del hogar
- intmonth: mes en que fue encuestado/a (1-12)
- married: si esta casado/a (binario)
- retired: si esta pensionado/a (binario)
- schadj: años de escolaridad
- urban: zona urbana (binario)
- wealth: riqueza neta (miles RMB)
- age: edad al entrar a la encuesta (no varia entre periodos)

Preguntas:

1. Cargar la base de datos *charls.csv* en el ambiente. Identifique los tipos de datos que se encuentran en la base, realice estadisticas descriptivas sobre las variables importantes (Hint:

Revisar la distribuciones, datos faltantes, outliers, etc.) y limpie las variables cuando sea necesario.

R = En primera instancia, el análisis de la base de datos nos deja entrever una atrición MAR, con valores aleatorios faltantes, los cuales son eliminados. En específico, los datos faltantes o "missing" corresponden a 7 instancias de la variable "drinkly" y a más de 20.000 filas ocasionadas por problemas en el formato de la variable "inid". Para el caso de "drinkly" no solo se eliminaron los datos asociados a esta variable faltante, sino que también se eliminaron todas las filas con un inid igual al problemático. De esta forma, se evitó que quedaran entidades con 2 o menos "waves" de datos.

2. Ejecute un modelo Pooled OLS para explicar el puntaje en la escala de salud mental (CESD). Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

R = Para utilizar el método Pooled OLS se debe asumir que el efecto del tiempo es igual para todos los individuos y el error individual no afecta el puntaje de salud mental estudiado. En este caso se añaden todas las variables de la base de datos. De esta manera se obtiene que solo la cantidad de hijos, el haber bebido en el último mes, la edad y la riqueza del individuo no representan importancia para la estimación de la variable dependiente debido a que el modelo arroja que no son significativas. Con un parámetro positivo, se infiere que tener hijos provocaría un empeoramiento en la salud mental del individuo, mientras que al tener parámetros positivos el haber bebido, tener más edad y tener más riquezas empeoraría la salud mental asumiendo que 0 es un puntaje más favorable.

3. Ejecute un modelo de efectos fijos para explicar el puntaje en la escala de salud mental (CESD). Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

R = Para ejecutar el modelo de efectos fijos se eliminaron las variables que no varían en el tiempo, ya que al hacer las diferencias de cada parámetro individual y su promedio, estas siempre darían cero. Estas variables a no considerar son "female", "urban", "schadj" y "age", representando si la entidad es mujer, si vive en zona urbana, sus años de escolaridad y su edad, respectivamente. Puede apreciarse fácilmente que es difícil que estas variables varíen en el tiempo del periodo de la encuesta ("age" en particular no varía porque así está construida la base de datos). Se asume también, que las características no observables y heterogéneas por individuo son las que están relacionadas con nuestras variables explicativas, pero no correlacionadas. En este caso, las variables significativas serían solo "hsize" y "married", representando que la explicación de la salud mental está explicada por estas dos variables, que representan el tamaño del hogar y el estado civil respectivamente. Ambas variables mejorarían la salud mental, puesto que tienen parámetros negativos.

4. Ejecute un modelo de efectos aleatorios para explicar el puntaje en la escala de salud mental (CESD). Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

R = Contrariamente al método anterior, ahora se asume que los errores entre los individuos son aleatorios y realmente sí están correlacionados entre sí. Este caso sí acepta en el modelo las variables fijas en el tiempo. Sin embargo, por fines comparativos, se utilizaron las mismas variables que en el inciso anterior. Tras ejecutar este modelo, se descubre que las variables significativas son "child", "married", "drinkly" y "hsize", lo cual nos dice que la salud mental está mayormente explicada por los efectos del número de hijos (que empeora el puntaje de salud mental puesto que tiene un parámetro positivo), del estado civil (con parámetro negativo indicando que mejora la salud

mental), si bebió en el último mes (mejorando la salud mental) y el tamaño del hogar, mejorando la salud mental.

5. Comente los resultados obtenidos en 2, 3 y 4. ¿Cuáles y por qué existen las diferencias entre los resultados?. En su opinión, ¿Cuál sería el más adecuado para responder la pregunta de investgación y por qué? ¿Qué variables resultaron ser robustas a la especificación?

R = Para comparar la estimación de RE y FE, se utiliza el test de Hausman. La hipótesis nula indicaría que los coeficientes entre los métodos no varían significativamente, dando así que ambos métodos son consistentes, pero que el método RE es eficiente. Mientras que la hipótesis alternativa indicaría que FE es consistente y RE no. El valor de la prueba p significativo (menor a 0.05) ayudaría a decidir que el mejor estimador es el de efectos fijos al no rechazar la hipótesis nula. Esto nos podría indicar que las diferencias y errores entre los individuos del panel no son aleatorios, y no existe correlación entre los individuos.

6. Ejecute un modelo de efectos aleatorios correlacionados (CRE) para explicar el puntaje en la escala de salud mental (CESD). Seleccione las variables dependientes a incluir en el modelo final e interprete su significado. Es este modelo adecuado, dada la data disponible, para modelar el componente no observado?

R = El modelo de CRE es usado a veces como alternativa al test de Haussmann. En este caso se utilizan todas las variables, más las medias de aquellas variables explicativas no fijas en el tiempo. Con esto se busca una mejor estimación sobre cual es el modelo adecuado para los datos de panel. El valor de los estimadores es similar al modelo fixed, prefiriendo nuevamente sobre Random effect. Las variables significativas son “female”, “urban”, “hsize”, “married”, “retired”, “schadj”, “mdrinkly”, “mhrsusu” e “mintmonth”.

7. Usando el modelo CRE, prediga la distribucion del componente no observado. Que puede inferir respecto de la heterogeneidad fija en el tiempo y su impacto en el puntaje CESD?

R = Una forma de observar la distribución del error, es graficando los residuos. Se cumple mejor el objetivo de la respuesta, analizando específicamente la homocedasticidad de dichos residuos. El modelo presenta heteroscedasticidad en el error, lo que podría ser indicador de que el experimento no es aplicado ni realizado de la misma manera por cada observación, alejando los resultados de una línea media. La apreciación personal de la dupla de trabajo es que la variable “intmonth” juega un papel importante en el error sistemático de cada prueba, afectando en la respuesta el periodo del mes en que se realice, tal y como se muestra en los gráficos representados en la sección de OLS.

8. Usando sus respuestas anteriores, que modelo prefiere? que se puede inferir en general respecto del efecto de las variables explicativas sobre el puntaje CESD?

R = El mejor modelo es el de correlated random effects. Podemos notar que hay variables fijas y no fijas en el tiempo que son importantes para la estimación del puntaje de la prueba de salud mental. Por ejemplo, “female” es una variable significativa que aumenta en 1.4688 unidades el puntaje de cesd por cada punto extra sí misma. Contrariamente, “hsize”, representando el tamaño del hogar, provoca que el puntaje de cesd disminuya en 0.1148 unidades a medida que aumenta su propio valor.