Tarea#2_Cardenas_Venegas

October 14, 2022

1 Tarea datos de panel

Autores : David Càrdenas y Cristobal Venegas

Fecha : 05/10/2022

[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 sklearn
import scipy
import linearmodels.panel as lmp

%matplotlib inline
```

1.1 Carga y limpieza de datos

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.

[2]:
```
# charls data
charls = pd.read_csv('../data/charls.csv')
charls.dropna(inplace=True)
charls.reset_index(drop=True, inplace=True)
# Limpieza de datos sobre la fila 10058 en adelante
charls.drop(range(10059, 34371, 1),axis=0,inplace= True)
```

[3]:
```
charls.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10059 entries, 0 to 10058
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   cesd      10059 non-null  int64
 1   child     10059 non-null  int64
```

```
 2   drinkly    10059 non-null  object
 3   female     10059 non-null  int64
 4   hrsusu     10059 non-null  float64
 5   hsize      10059 non-null  int64
 6   inid       10059 non-null  float64
 7   intmonth   10059 non-null  int64
 8   married    10059 non-null  int64
 9   retired    10059 non-null  int64
 10  schadj     10059 non-null  int64
 11  urban      10059 non-null  int64
 12  wave       10059 non-null  int64
 13  wealth     10059 non-null  float64
 14  age        10059 non-null  int64
dtypes: float64(3), int64(11), object(1)
memory usage: 1.2+ MB
```

1.1.1 Revision de datos ¿El panel esta balanceado?

[4]:
```
print(charls["inid"].value_counts())
```

```
1.010410e+10    3
5.865932e+10    3
6.044011e+10    3
6.044011e+10    3
6.044011e+10    3
               ..
4.674623e+10    3
4.674623e+10    3
4.674623e+10    3
4.674623e+10    3
1.017910e+11    3
Name: inid, Length: 3353, dtype: int64
```

[5]:
```
charls["valores"]=1
charls.groupby("wave").sum()["valores"]
```

[5]:
```
wave
1    3353
2    3353
3    3353
Name: valores, dtype: int64
```

1.1.2 Estudio de Missing

[6]:
```
## Nos damos cuenta que hay 7 filas con ".m:missing        "
charls.groupby("drinkly").count()
```

[6]:
```
            cesd  child  female  hrsusu  hsize  inid  intmonth  married  \
drinkly
.m:missing     7      7       7       7      7     7         7        7
0.None      6791   6791    6791    6791   6791  6791      6791     6791
1.Yes       3261   3261    3261    3261   3261  3261      3261     3261

            retired  schadj  urban  wave  wealth   age  valores
drinkly
.m:missing        7       7      7     7       7     7        7
0.None         6791    6791   6791  6791    6791  6791     6791
1.Yes          3261    3261   3261  3261    3261  3261     3261
```

[7]:
```
## reemplazamos los valores 0.None por 0 y 1.Yes por 1
charls.replace({"0.None": 0, "1.Yes": 1},inplace=True)
```

[8]:
```
a=charls.loc[charls.drinkly==".m:missing"]
a
```

[8]:
```
      cesd  child     drinkly  female  hrsusu  hsize          inid  intmonth  \
4712     1      5  .m:missing       0     0.0      2  5.605921e+10         8
4813     4      3  .m:missing       1    42.0      3  5.605931e+10         7
5878    10      3  .m:missing       0    42.0      4  5.745731e+10        10
6326    10      1  .m:missing       1     0.0      3  5.820230e+10         8
6394    10      4  .m:missing       0     0.0      2  5.820232e+10         8
9142    20      2  .m:missing       1     9.0      6  7.498132e+10        10
9227     5      3  .m:missing       0    40.0      2  7.537612e+10         7

      married  retired  schadj  urban  wave   wealth  age  valores
4712        1        1       4      0     3   1200.0   74        1
4813        1        0       0      0     2   1600.0   56        1
5878        1        0       4      0     2      0.0   69        1
6326        0        0       0      1     3      0.0   50        1
6394        1        1       4      1     2    180.0   70        1
9142        1        0       0      0     2  52000.0   46        1
9227        1        0       8      0     3  20900.0   53        1
```

[9]:
```
## por ejemllo ahi hay que eliminar esas 3 porque si elimino solo el messing de␣
↪la fila 4813 se me desbalancea el panel
charls.loc[charls.inid==56059314002.0]
```

[9]:
```
      cesd  child     drinkly  female  hrsusu  hsize          inid  intmonth  \
4812    17      3           1       1    35.0      3  5.605931e+10         7
4813     4      3  .m:missing       1    42.0      3  5.605931e+10         7
4814    14      3           1       1    56.0      3  5.605931e+10         7

      married  retired  schadj  urban  wave  wealth  age  valores
4812        1        0       0      0     1  1000.0   56        1
```

```
4813        1        0       0       0     2  1600.0   56        1
4814        1        0       0       0     3 -9000.0   56        1
```

[10]:
```
lista=[4712,4813,5878,6326,6394,9142,9227]
cac=[]
a=charls.loc[charls.drinkly==".m:missing"]
for i in (lista):
    print(i)
    cac.append(a["inid"][i])

cac = list(set(cac))
cac
```

```
4712
4813
5878
6326
6394
9142
9227
```

[10]:
```
[58202320002.0,
 74981324002.0,
 57457309001.0,
 58202302001.0,
 56059314002.0,
 75376118001.0,
 56059207001.0]
```

[11]:
```
#for i in range (len(charls)):
for i in  range (len(cac)):
    b=charls.loc[charls["inid"]!=cac[i],:]
    charls = b
```

[12]:
```
charls.groupby("drinkly").count()
```

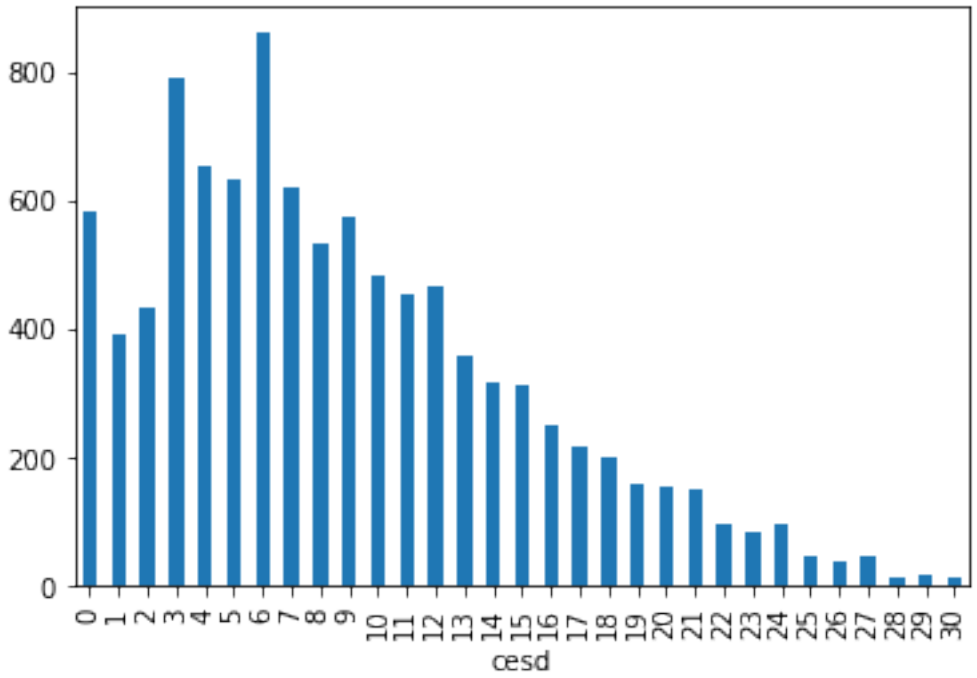
[12]:
```
          cesd  child  female  hrsusu  hsize  inid  intmonth  married  retired  \
drinkly
0         6782   6782    6782    6782   6782  6782      6782     6782     6782
1         3256   3256    3256    3256   3256  3256      3256     3256     3256

          schadj  urban  wave  wealth   age  valores
drinkly
0           6782   6782  6782    6782  6782     6782
1           3256   3256  3256    3256  3256     3256
```

[13]:
```
#Verificamos que se hayan eliminado las 7 id en cada wave de 3353 pasar a 3346
charls.groupby("wave").sum()["valores"]
```

[13]:
```
wave
1    3346
2    3346
3    3346
Name: valores, dtype: int64
```

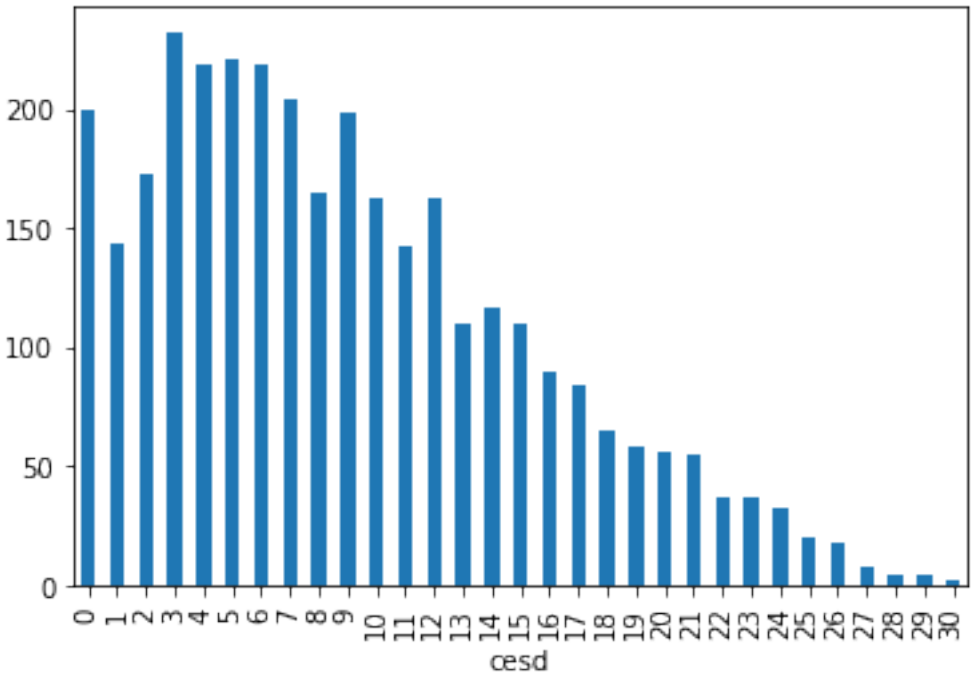
[14]:
```
##Visualizamos la cantidad de gente con sus valores cesd
charls.groupby(['cesd']).sum()["valores"].plot.bar()
```

[14]:
```
<AxesSubplot:xlabel='cesd'>
```



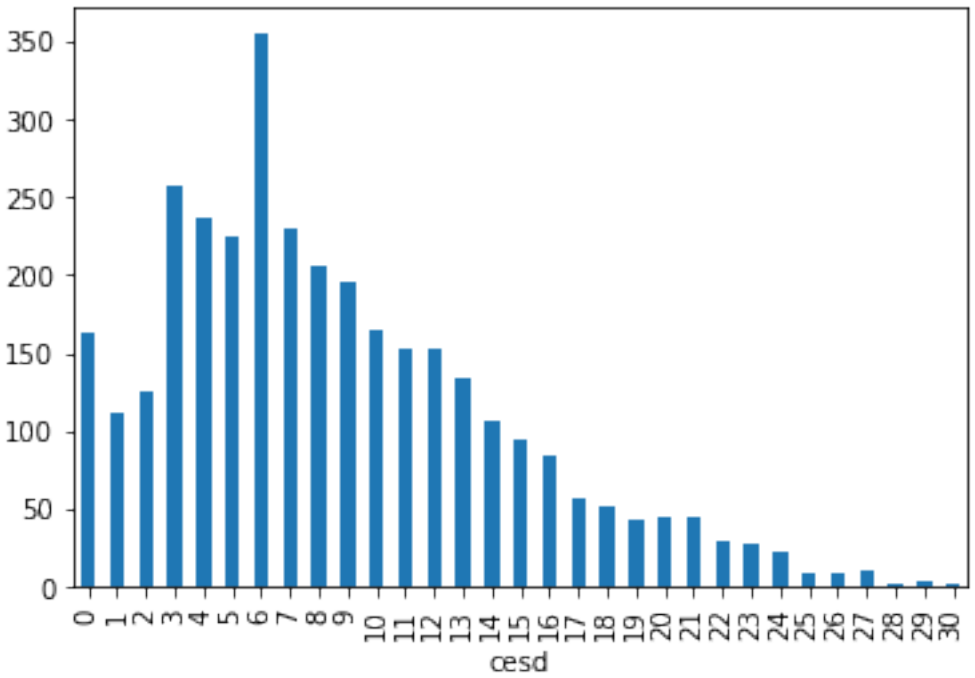
[15]:
```
## Visualizamos como se distribuye cesd por cada wave
charls.loc[charls.wave==1].groupby(["cesd"]).count()["wave"].plot.bar()
```

[15]:
```
<AxesSubplot:xlabel='cesd'>
```



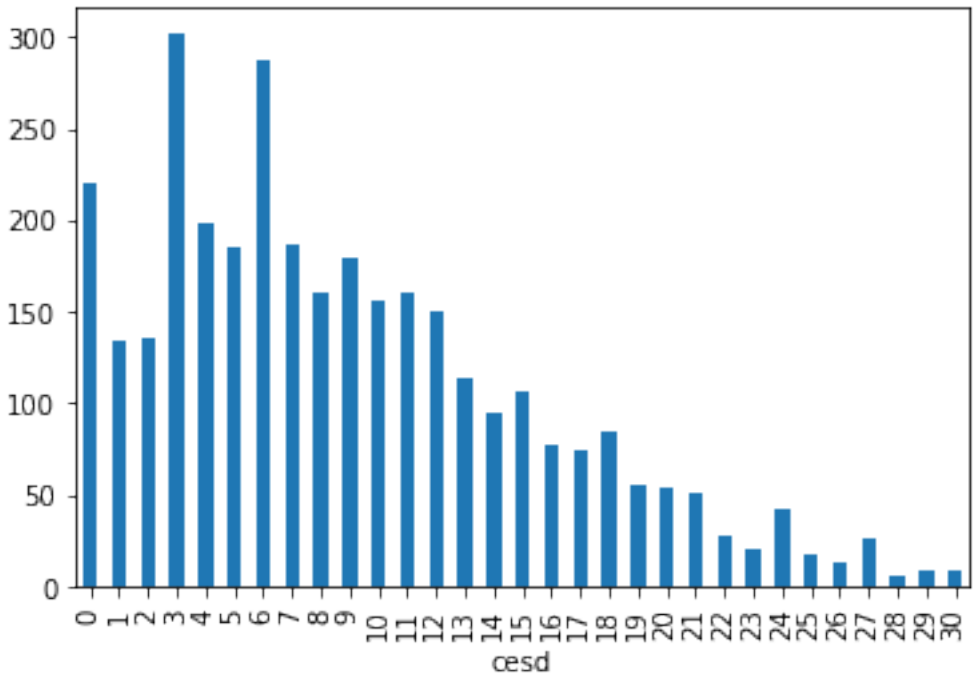
[16]:
```
charls.loc[charls.wave==2].groupby(["cesd"]).count()["wave"].plot.bar()
```

[16]: <AxesSubplot:xlabel='cesd'>



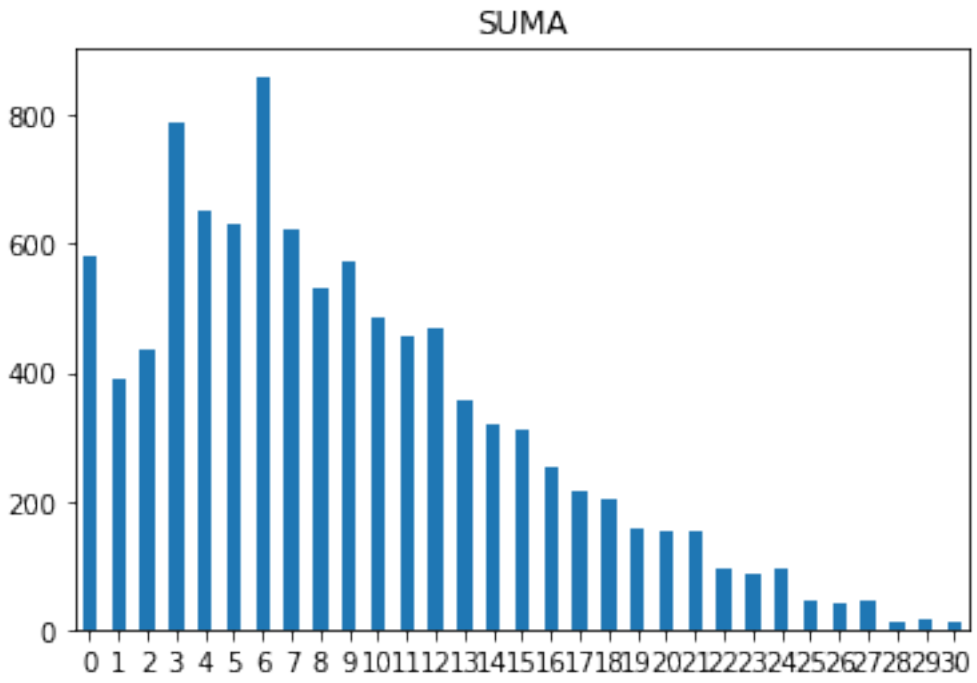
[17]:
```
charls.loc[charls.wave==3].groupby(["cesd"]).count()["wave"].plot.bar()
```

[17]: <AxesSubplot:xlabel='cesd'>



[18]:
```
charls["cesd"].value_counts().sort_index().plot.bar(y= "cesd",rot = 0 , title =␣
↪"SUMA")
```

[18]: <AxesSubplot:title={'center':'SUMA'}>



[19]:
```
charls["cesd"].mean()
```

[19]:
```
8.864614465032876
```

[20]:
```
charls.loc[charls.wave==3].groupby(["wave"]).mean()
```

[20]:
```
          cesd    child    female     hrsusu     hsize          inid  \
wave
3     8.985057  2.89211  0.542439  24.902869  3.161984  4.888676e+10

      intmonth   married   retired    schadj     urban        wealth  \
wave
3     7.295876  0.842499  0.313509  4.095039  0.315601  16176.098924

            age  valores
wave
3     58.223551      1.0
```

[21]:
```
# Modificamos la columna drinkly ya que es un objeto a int para poder agregarla␣
↪al modelo
charls['drinkly']=charls['drinkly'].astype('int')
```

1.1.3 Construccion de variables

[22]:
```
#variable construction
X=charls[['child','female','hrsusu','hsize','intmonth','married','retired','schadj','urban','w
Xm=(X.groupby(charls['inid']).transform('mean'))
Xid=charls[['inid',"wave",'cesd','child','female','hrsusu','hsize','intmonth','married','retir
Xc=pd.DataFrame(np.c_[Xid, Xm],␣
 ↪columns=['inid','wave',"cesd",'child','female','hrsusu','hsize','intmonth','married','retir
#set panel structure
Xc = Xc.set_index(["inid","wave"])
Xc.describe()
```

[22]:

	cesd	child	female	hrsusu	hsize	\
count	10038.000000	10038.000000	10038.000000	10038.000000	10038.000000	
mean	8.864614	2.768779	0.542439	27.963987	3.652919	
std	6.289969	1.436246	0.498221	27.250418	1.785352	
min	0.000000	0.000000	0.000000	0.000000	1.000000	
25%	4.000000	2.000000	0.000000	0.000000	2.000000	
50%	8.000000	2.000000	1.000000	24.000000	3.000000	
75%	13.000000	3.000000	1.000000	49.000000	5.000000	
max	30.000000	10.000000	1.000000	168.000000	13.000000	

	intmonth	married	retired	schadj	urban	\
count	10038.000000	10038.000000	10038.000000	10038.000000	10038.000000	
mean	7.592847	0.857940	0.268679	4.095039	0.315601	
std	1.101095	0.349129	0.443295	3.603898	0.464778	
min	1.000000	0.000000	0.000000	0.000000	0.000000	
25%	7.000000	1.000000	0.000000	0.000000	0.000000	
50%	7.000000	1.000000	0.000000	4.000000	0.000000	
75%	8.000000	1.000000	1.000000	4.000000	1.000000	
max	12.000000	1.000000	1.000000	16.000000	1.000000	

	…	mhrsusu	mhsize	mintmonth	mmarried	\
count	…	10038.000000	10038.000000	10038.000000	10038.000000	
mean	…	27.963987	3.652919	7.592847	0.857940	
std	…	21.282866	1.460633	0.630944	0.333064	
min	…	0.000000	1.000000	5.000000	0.000000	
25%	…	8.333333	2.333333	7.333333	1.000000	
50%	…	28.000000	3.666667	7.666667	1.000000	
75%	…	43.333333	4.666667	8.000000	1.000000	
max	…	119.000000	10.000000	10.000000	1.000000	

	mretired	mschadj	murban	mwealth	mage	\
count	10038.000000	10038.000000	10038.000000	1.003800e+04	10038.000000	
mean	0.268679	4.095039	0.315601	1.021378e+04	58.223551	
std	0.365890	3.603898	0.464778	6.284536e+04	9.232748	
min	0.000000	0.000000	0.000000	-3.250000e+05	16.000000	

```
25%         0.000000      0.000000      0.000000  8.333333e+01     51.000000
50%         0.000000      4.000000      0.000000  1.071667e+03     58.000000
75%         0.333333      4.000000      1.000000  8.666667e+03     64.000000
max         1.000000     16.000000      1.000000  2.672550e+06     89.000000

           mdrinkly
count  10038.000000
mean       0.324367
std        0.406825
min        0.000000
25%        0.000000
50%        0.000000
75%        0.666667
max        1.000000

[8 rows x 25 columns]
```

[23]:
```
charls.groupby(["wave"]).describe()
```

[23]:
```
        cesd                                                    child            \
       count      mean       std  min  25%  50%   75%   max   count      mean
wave
1     3346.0  9.021518  6.422179  0.0  4.0  8.0  13.0  30.0  3346.0  2.650926
2     3346.0  8.587268  5.822461  0.0  4.0  7.0  12.0  30.0  3346.0  2.763299
3     3346.0  8.985057  6.591911  0.0  4.0  8.0  13.0  30.0  3346.0  2.892110

      …   age       valores
      …   75%   max   count mean  std  min  25%  50%  75%  max
wave  …
1     …  64.0  89.0  3346.0  1.0  0.0  1.0  1.0  1.0  1.0  1.0
2     …  64.0  89.0  3346.0  1.0  0.0  1.0  1.0  1.0  1.0  1.0
3     …  64.0  89.0  3346.0  1.0  0.0  1.0  1.0  1.0  1.0  1.0

[3 rows x 120 columns]
```

1.1.4 Estudio de variables que cambian con el tiempo

[24]:
```
#aca distingo que variables cambian en el tiempo y cuales no,
#Por ejemplo la variable CHILD cambia en el tiempo fijense en sus MEDIAS
#la variable Female no cambia en el tiempo si , los promedios de cada wave,
#nos ayuda a definir el modelo despues mira eso es lo que tienen que considerar␣
 ↪en su analisis  SI
pd.options.display.max_columns = 1000
charls.groupby(["wave"]).describe()
```

[24]:

	cesd								child		
	count	mean	std	min	25%	50%	75%	max	count	mean	
wave											
1	3346.0	9.021518	6.422179	0.0	4.0	8.0	13.0	30.0	3346.0	2.650926	
2	3346.0	8.587268	5.822461	0.0	4.0	7.0	12.0	30.0	3346.0	2.763299	
3	3346.0	8.985057	6.591911	0.0	4.0	8.0	13.0	30.0	3346.0	2.892110	

						drinkly				\
	std	min	25%	50%	75%	max	count	mean	std	min
wave										
1	1.412551	0.0	2.0	2.0	3.0	10.0	3346.0	0.329647	0.470155	0.0
2	1.426897	0.0	2.0	2.0	3.0	10.0	3346.0	0.326659	0.469061	0.0
3	1.459165	0.0	2.0	3.0	4.0	10.0	3346.0	0.316796	0.465297	0.0

					female							\
	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
wave												
1	0.0	0.0	1.0	1.0	3346.0	0.542439	0.49827	0.0	0.0	1.0	1.0	1.0
2	0.0	0.0	1.0	1.0	3346.0	0.542439	0.49827	0.0	0.0	1.0	1.0	1.0
3	0.0	0.0	1.0	1.0	3346.0	0.542439	0.49827	0.0	0.0	1.0	1.0	1.0

	hrsusu								hsize	\
	count	mean	std	min	25%	50%	75%	max	count	
wave										
1	3346.0	30.604304	27.880731	0.0	0.0	30.0	56.0	144.0	3346.0	
2	3346.0	28.384788	26.806486	0.0	0.0	25.0	49.0	140.0	3346.0	
3	3346.0	24.902869	26.753567	0.0	0.0	18.0	42.0	168.0	3346.0	

								inid		\
	mean	std	min	25%	50%	75%	max	count	mean	
wave										
1	3.840406	1.859551	1.0	2.0	4.0	5.0	13.0	3346.0	4.888676e+10	
2	3.956366	1.946338	1.0	2.0	4.0	5.0	13.0	3346.0	4.888676e+10	
3	3.161984	1.396366	1.0	2.0	3.0	4.0	11.0	3346.0	4.888676e+10	

					\
	std	min	25%	50%	75%
wave					
1	2.298507e+10	1.010410e+10	3.110611e+10	5.630230e+10	6.403312e+10
2	2.298507e+10	1.010410e+10	3.110611e+10	5.630230e+10	6.403312e+10
3	2.298507e+10	1.010410e+10	3.110611e+10	5.630230e+10	6.403312e+10

		intmonth						\
	max	count	mean	std	min	25%	50%	75%	max
wave									
1	1.017910e+11	3346.0	7.801554	1.375662	1.0	7.0	8.0	8.0	12.0
2	1.017910e+11	3346.0	7.681112	0.911764	7.0	7.0	7.0	8.0	12.0

```
3     1.017910e+11   3346.0  7.295876  0.880121  1.0  7.0  7.0  8.0   9.0

     married                                                retired            \
       count      mean       std  min  25%  50%  75%  max   count      mean
wave
1     3346.0  0.874178  0.331698  0.0  1.0  1.0  1.0  1.0  3346.0  0.227137
2     3346.0  0.857143  0.349979  0.0  1.0  1.0  1.0  1.0  3346.0  0.265392
3     3346.0  0.842499  0.364327  0.0  1.0  1.0  1.0  1.0  3346.0  0.313509

                                        schadj                                \
           std  min  25%  50%  75%  max   count      mean       std  min  25%
wave
1     0.419044  0.0  0.0  0.0  0.0  1.0  3346.0  4.095039  3.604257  0.0  0.0
2     0.441607  0.0  0.0  0.0  1.0  1.0  3346.0  4.095039  3.604257  0.0  0.0
3     0.463988  0.0  0.0  0.0  1.0  1.0  3346.0  4.095039  3.604257  0.0  0.0

                    urban                                                    \
      50%  75%   max   count      mean       std  min  25%  50%  75%  max
wave
1     4.0  4.0  16.0  3346.0  0.315601  0.464824  0.0  0.0  0.0  1.0  1.0
2     4.0  4.0  16.0  3346.0  0.315601  0.464824  0.0  0.0  0.0  1.0  1.0
3     4.0  4.0  16.0  3346.0  0.315601  0.464824  0.0  0.0  0.0  1.0  1.0

      wealth                                                                 \
       count          mean            std       min  25%    50%     75%
wave
1     3346.0   5802.647645   54559.105027 -975000.0  0.0  300.0  2200.0
2     3346.0   8662.585923   51952.488992 -499500.0  0.0  300.0  4300.0
3     3346.0  16176.098924  154912.131064 -596000.0  0.0  500.0  7000.0

                  age                                                        \
            max   count       mean       std   min   25%   50%   75%   max
wave
1      900100.0  3346.0  58.223551  9.233668  16.0  51.0  58.0  64.0  89.0
2     1001500.0  3346.0  58.223551  9.233668  16.0  51.0  58.0  64.0  89.0
3     8001500.0  3346.0  58.223551  9.233668  16.0  51.0  58.0  64.0  89.0

     valores
       count mean  std  min  25%  50%  75%  max
wave
1     3346.0  1.0  0.0  1.0  1.0  1.0  1.0  1.0
2     3346.0  1.0  0.0  1.0  1.0  1.0  1.0  1.0
3     3346.0  1.0  0.0  1.0  1.0  1.0  1.0  1.0
```

VARIABLES QUE NO CAMBIAN EN EL TIEMPO -AGE

-URBAN

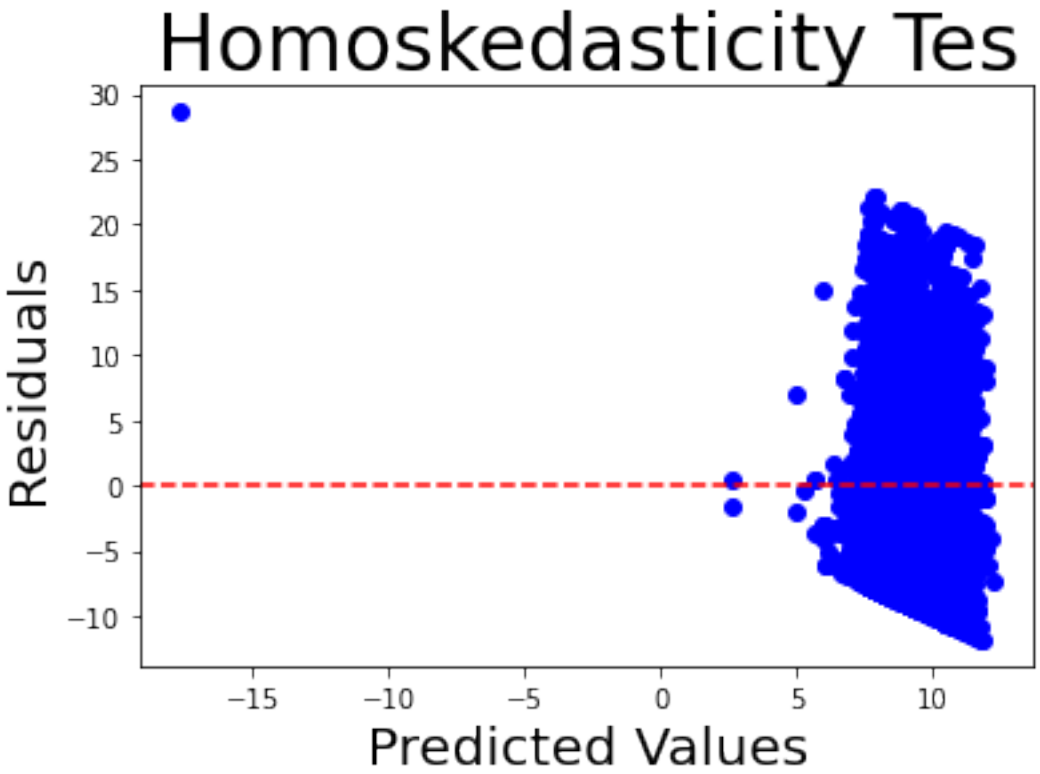
-SCHADJ

-FEMALE

1.2 Pooled OLS

[25]:
```
# Perform PooledOLS
from linearmodels import PooledOLS
import statsmodels.api as sm
exog = sm.tools.tools.
 ↪add_constant(Xc[['wealth',"child","hrsusu","hsize","intmonth","married","retired","drinkly"]
endog = Xc['cesd']
mod = PooledOLS(endog, exog)
pooledOLS_res = mod.fit(cov_type='robust')
# Store values for checking homoskedasticity graphically
fittedvals_pooled_OLS = pooledOLS_res.predict().fitted_values
residuals_pooled_OLS = pooledOLS_res.resids
```

[26]:
```
# 3A. Homoskedasticity
import matplotlib.pyplot as plt
 # 3A.1 Residuals-Plot for growing Variance Detection
fig, ax = plt.subplots()
ax.scatter(fittedvals_pooled_OLS, residuals_pooled_OLS, color = "blue")
ax.axhline(0, color = 'r', ls = '--')
ax.set_xlabel("Predicted Values", fontsize = 20)
ax.set_ylabel("Residuals", fontsize = 20)
ax.set_title("Homoskedasticity Tes", fontsize = 30)
plt.show()
```



[27]:
```
from statsmodels.stats.diagnostic import het_white, het_breuschpagan
pooled_OLS_dataset = pd.concat([Xc, residuals_pooled_OLS], axis=1)
breusch_pagan_test_results = het_breuschpagan(pooled_OLS_dataset["residual"],␣
 ↪exog)
labels = ["LM-Stat", "LM p-val", "F-Stat", "F p-val"]
print(dict(zip(labels, breusch_pagan_test_results)))
```

```
{'LM-Stat': 136.97553109867232, 'LM p-val': 1.0092237595604324e-25, 'F-Stat':
17.343250762880597, 'F p-val': 6.685546523335638e-26}
```

[28]:
```
# 3.B Non-Autocorrelation
# Durbin-Watson-Test
from statsmodels.stats.stattools import durbin_watson

durbin_watson_test_results = durbin_watson(pooled_OLS_dataset["residual"])
print(durbin_watson_test_results)
```

```
1.2938072562447387
```

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.

[29]:
```
y=Xc['cesd']
X=Xc[['wealth',"child","hrsusu","hsize","intmonth","married","retired","drinkly"]]
X=sm.add_constant(X)

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

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   cesd   R-squared:                       0.022
Model:                           OLS   Adj. R-squared:                  0.021
Method:                Least Squares   F-statistic:                     28.07
Date:               Wed, 05 Oct 2022   Prob (F-statistic):           1.34e-43
Time:                       20:30:15   Log-Likelihood:                -32591.
No. Observations:              10038   AIC:                         6.520e+04
Df Residuals:                  10029   BIC:                         6.527e+04
Df Model:                          8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          9.6791      0.500     19.376      0.000       8.700      10.658
wealth     -3.226e-06   6.26e-07     -5.157      0.000   -4.45e-06      -2e-06
child          0.2467      0.044      5.578      0.000       0.160       0.333
hrsusu         0.0042      0.003      1.438      0.151      -0.002       0.010
hsize         -0.0315      0.036     -0.886      0.375      -0.101       0.038
intmonth       0.0485      0.056      0.858      0.391      -0.062       0.159
married       -1.7480      0.186     -9.411      0.000      -2.112      -1.384
retired       -0.1679      0.181     -0.925      0.355      -0.524       0.188
drinkly       -0.8957      0.135     -6.642      0.000      -1.160      -0.631
==============================================================================
Omnibus:                      774.354   Durbin-Watson:                   1.294
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              966.774
Skew:                           0.758   Prob(JB):                    1.17e-210
Kurtosis:                       3.105   Cond. No.                     8.17e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 8.17e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[30]:
```
model=lmp.PooledOLS(y,X)
OLS=model.fit(cov_type="robust")
print(OLS)
```

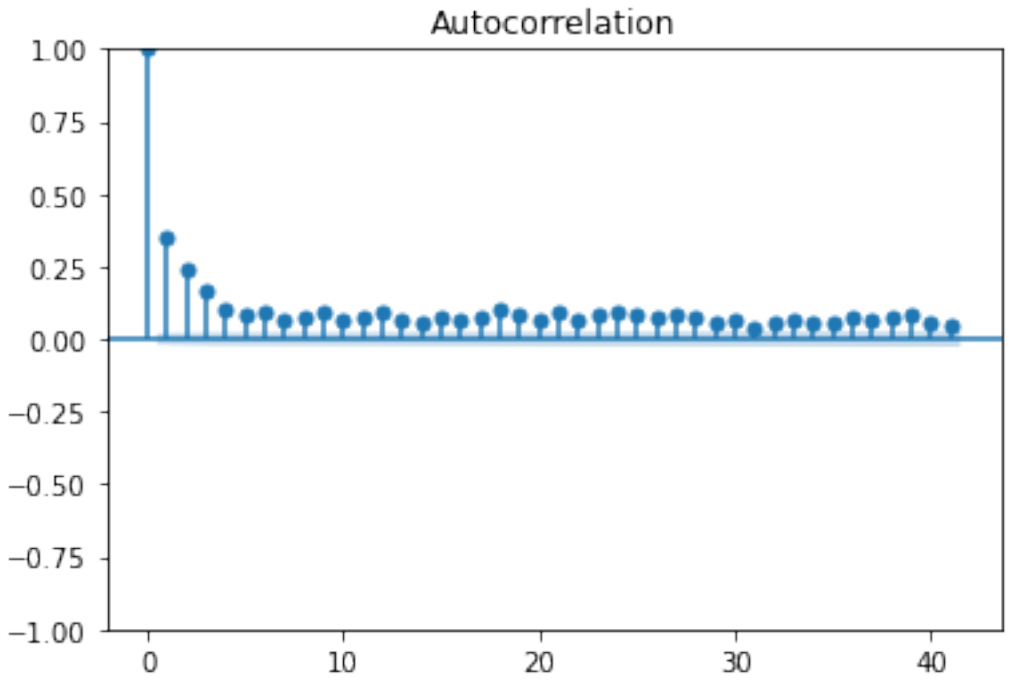
PooledOLS Estimation Summary

Dep. Variable:	cesd	R-squared:	0.0219
Estimator:	PooledOLS	R-squared (Between):	0.0370
No. Observations:	10038	R-squared (Within):	-0.0080
Date:	Wed, Oct 05 2022	R-squared (Overall):	0.0219
Time:	20:30:15	Log-likelihood	-3.259e+04
Cov. Estimator:	Robust		
		F-statistic:	28.073
Entities:	3346	P-value	0.0000
Avg Obs:	3.0000	Distribution:	F(8,10029)
Min Obs:	3.0000		
Max Obs:	3.0000	F-statistic (robust):	22.864
		P-value	0.0000
Time periods:	3	Distribution:	F(8,10029)
Avg Obs:	3346.0		
Min Obs:	3346.0		
Max Obs:	3346.0		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	9.6791	0.4842	19.992	0.0000	8.7301	10.628
wealth	-3.226e-06	2.332e-06	-1.3836	0.1665	-7.798e-06	1.345e-06
child	0.2467	0.0450	5.4853	0.0000	0.1585	0.3348
hrsusu	0.0042	0.0029	1.4668	0.1424	-0.0014	0.0098
hsize	-0.0315	0.0349	-0.9039	0.3661	-0.0999	0.0368
intmonth	0.0485	0.0549	0.8824	0.3776	-0.0592	0.1561
married	-1.7480	0.2006	-8.7118	0.0000	-2.1413	-1.3547
retired	-0.1679	0.1848	-0.9083	0.3637	-0.5302	0.1944
drinkly	-0.8957	0.1321	-6.7830	0.0000	-1.1545	-0.6368

[31]:
```
import pandas as pd
import scipy.stats as st
import statsmodels.api as sm
import statsmodels.graphics.tsaplots as tsap
from statsmodels.compat import lzip
from statsmodels.stats.diagnostic import het_white
from matplotlib import pyplot as plt
import seaborn as sns
tsap.plot_acf(x=results.resid)
plt.show()
```



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.

1.3 Efectos fijos

[32]:
```
#si agrego una variable que no cambia con el tiempo , me tira error
#solo agregar variables que cambian en el tiempo que son las siguientes:
X=Xc[['wealth',"child","hrsusu","hsize","intmonth","married","retired","drinkly"]]
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):              0.0125
No. Observations:               10038   R-squared (Within):               0.0039
Date:                Wed, Oct 05 2022   R-squared (Overall):              0.0096
Time:                        20:30:16   Log-likelihood                -2.72e+04
Cov. Estimator:              Robust
                                        F-statistic:                      3.3026
Entities:                        3346   P-value                           0.0009
Avg Obs:                       3.0000   Distribution:                  F(8,6684)
Min Obs:                       3.0000
```

```
Max Obs:                        3.0000   F-statistic (robust):             2.7983
                                         P-value                           0.0043
Time periods:                        3   Distribution:                  F(8,6684)
Avg Obs:                        3346.0
Min Obs:                        3346.0
Max Obs:                        3346.0

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const          9.8949     0.6694     14.782     0.0000      8.5827      11.207
wealth     -5.199e-07  8.429e-07    -0.6169     0.5373  -2.172e-06  1.132e-06
child          0.1493     0.0960     1.5553     0.1199     -0.0389      0.3375
hrsusu        -0.0004     0.0029    -0.1408     0.8880     -0.0061      0.0053
hsize         -0.1204     0.0442    -2.7241     0.0065     -0.2070     -0.0337
intmonth      -0.0177     0.0507    -0.3492     0.7270     -0.1170      0.0817
married       -1.1832     0.5064    -2.3362     0.0195     -2.1759     -0.1904
retired        0.3476     0.2022     1.7192     0.0856     -0.0488      0.7439
drinkly        0.2121     0.1886     1.1246     0.2608     -0.1576      0.5819
==============================================================================

F-test for Poolability: 3.8483
P-value: 0.0000
Distribution: F(3345,6684)

Included effects: Entity
```

1.4 Efectos aleatorios

[33]:
```
# FE und RE model
from linearmodels import PanelOLS
from linearmodels import RandomEffects
exog = sm.tools.tools.
  ↪add_constant(Xc[['wealth',"child","hrsusu","hsize","intmonth","married","retired","drinkly"]
endog = Xc["cesd"]
# modelo de efectos aleatorios
model_re = RandomEffects(endog, exog)
re_res = model_re.fit(cov_type="robust")
print(re_res)
```

```
                        RandomEffects Estimation Summary
================================================================================
Dep. Variable:                   cesd   R-squared:                        0.0103
Estimator:                RandomEffects   R-squared (Between):              0.0284
No. Observations:                10038   R-squared (Within):               0.0009
Date:                Wed, Oct 05 2022   R-squared (Overall):              0.0192
Time:                        20:30:16   Log-likelihood                -2.927e+04
```

```
Cov. Estimator:                Robust
                                          F-statistic:                    13.041
Entities:                        3346     P-value                         0.0000
Avg Obs:                       3.0000     Distribution:                F(8,10029)
Min Obs:                       3.0000
Max Obs:                       3.0000     F-statistic (robust):           11.129
                                          P-value                         0.0000
Time periods:                       3     Distribution:                F(8,10029)
Avg Obs:                       3346.0
Min Obs:                       3346.0
Max Obs:                       3346.0

                            Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const          9.9065     0.4652     21.295     0.0000      8.9947      10.818
wealth     -1.541e-06  1.126e-06    -1.3688     0.1711  -3.749e-06  6.659e-07
child          0.2300     0.0547     4.2010     0.0000      0.1227      0.3373
hrsusu         0.0016     0.0026     0.6058     0.5446     -0.0035      0.0066
hsize         -0.0721     0.0352    -2.0476     0.0406     -0.1412     -0.0031
intmonth       0.0039     0.0463     0.0841     0.9330     -0.0869      0.0947
married       -1.5836     0.2486    -6.3695     0.0000     -2.0710     -1.0963
retired        0.0863     0.1727     0.4999     0.6172     -0.2522      0.4249
drinkly       -0.4231     0.1429    -2.9607     0.0031     -0.7032     -0.1430
==============================================================================
```

[34]:
```
model=lmp.RandomEffects(y,X)
re=model.fit(cov_type="robust")
print(re)
```

```
                      RandomEffects Estimation Summary
================================================================================
Dep. Variable:                   cesd     R-squared:                      0.0103
Estimator:              RandomEffects     R-squared (Between):            0.0284
No. Observations:               10038     R-squared (Within):             0.0009
Date:                Wed, Oct 05 2022     R-squared (Overall):            0.0192
Time:                        20:30:16     Log-likelihood              -2.927e+04
Cov. Estimator:                Robust
                                          F-statistic:                    13.041
Entities:                        3346     P-value                         0.0000
Avg Obs:                       3.0000     Distribution:                F(8,10029)
Min Obs:                       3.0000
Max Obs:                       3.0000     F-statistic (robust):           11.129
                                          P-value                         0.0000
Time periods:                       3     Distribution:                F(8,10029)
Avg Obs:                       3346.0
Min Obs:                       3346.0
```

```
Max Obs:                        3346.0

                            Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const          9.9065     0.4652     21.295     0.0000      8.9947      10.818
wealth     -1.541e-06  1.126e-06    -1.3688     0.1711  -3.749e-06  6.659e-07
child          0.2300     0.0547     4.2010     0.0000      0.1227      0.3373
hrsusu         0.0016     0.0026     0.6058     0.5446     -0.0035      0.0066
hsize         -0.0721     0.0352    -2.0476     0.0406     -0.1412     -0.0031
intmonth       0.0039     0.0463     0.0841     0.9330     -0.0869      0.0947
married       -1.5836     0.2486    -6.3695     0.0000     -2.0710     -1.0963
retired        0.0863     0.1727     0.4999     0.6172     -0.2522      0.4249
drinkly       -0.4231     0.1429    -2.9607     0.0031     -0.7032     -0.1430
==============================================================================
```

[35]:
```
re.variance_decomposition
```

[35]:
```
Effects                   18.635811
Residual                  19.860553
Percent due to Effects     0.484093
Name: Variance Decomposition, dtype: float64
```

1.5 COMPARACION OLS , EF Y RE

[36]:
```
print(lmp.compare({"FE": fe, "RE": re, "Pooled": OLS}))
```

```
                            Model Comparison
=============================================================================
                                   FE                RE         Pooled
-----------------------------------------------------------------------------
Dep. Variable                    cesd              cesd           cesd
Estimator                   PanelOLS     RandomEffects      PooledOLS
No. Observations               10038             10038          10038
Cov. Est.                     Robust            Robust         Robust
R-squared                     0.0039            0.0103         0.0219
R-Squared (Within)            0.0039            0.0009        -0.0080
R-Squared (Between)           0.0125            0.0284         0.0370
R-Squared (Overall)           0.0096            0.0192         0.0219
F-statistic                   3.3026            13.041         28.073
P-value (F-stat)              0.0009            0.0000         0.0000
====================     ============   ===============   ============
const                         9.8949            9.9065         9.6791
                             (14.782)          (21.295)       (19.992)
wealth                    -5.199e-07        -1.541e-06     -3.226e-06
                             (-0.6169)         (-1.3688)      (-1.3836)
```

```
child                          0.1493          0.2300          0.2467
                             (1.5553)        (4.2010)        (5.4853)
hrsusu                        -0.0004          0.0016          0.0042
                            (-0.1408)        (0.6058)        (1.4668)
hsize                         -0.1204         -0.0721         -0.0315
                            (-2.7241)       (-2.0476)       (-0.9039)
intmonth                      -0.0177          0.0039          0.0485
                            (-0.3492)        (0.0841)        (0.8824)
married                       -1.1832         -1.5836         -1.7480
                            (-2.3362)       (-6.3695)       (-8.7118)
retired                        0.3476          0.0863         -0.1679
                             (1.7192)        (0.4999)       (-0.9083)
drinkly                        0.2121         -0.4231         -0.8957
                             (1.1246)        (-2.9607)       (-6.7830)
======================== ============== ================ ==============
Effects                          Entity
-------------------------------------------------------------------------

T-stats reported in parentheses
```

1.6 Test hausman

[37]:
```
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
```

[38]:
```
htest = hausman(fe, re)
print("Hausman Test: chi-2 = {0}, df = {1}, p-value = {2}".format(htest[0],␣
 ↪htest[1], htest[2]))
```

```
Hausman Test: chi-2 = 42.936489479396805, df = 8, p-value = 9.02953360063885e-07
```

[39]:
```
X=Xc[['child','hrsusu','hsize','intmonth','married','retired','wealth',"drinkly",'mchild','mhr
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:                        0.0168
```

Estimator:	RandomEffects	R-squared (Between):	0.0417
No. Observations:	10038	R-squared (Within):	0.0039
Date:	Wed, Oct 05 2022	R-squared (Overall):	0.0290
Time:	20:30:16	Log-likelihood	-2.923e+04
Cov. Estimator:	Robust		
		F-statistic:	10.711
Entities:	3346	P-value	0.0000
Avg Obs:	3.0000	Distribution:	F(16,10021)
Min Obs:	3.0000		
Max Obs:	3.0000	F-statistic (robust):	8.9346
		P-value	0.0000
Time periods:	3	Distribution:	F(16,10021)
Avg Obs:	3346.0		
Min Obs:	3346.0		
Max Obs:	3346.0		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	8.6171	1.1276	7.6421	0.0000	6.4068	10.827
child	0.1493	0.1006	1.4835	0.1380	-0.0480	0.3466
hrsusu	-0.0004	0.0029	-0.1403	0.8884	-0.0061	0.0053
hsize	-0.1204	0.0445	-2.7045	0.0069	-0.2076	-0.0331
intmonth	-0.0177	0.0493	-0.3592	0.7195	-0.1143	0.0789
married	-1.1832	0.4870	-2.4295	0.0151	-2.1378	-0.2285
retired	0.3476	0.2025	1.7166	0.0861	-0.0493	0.7445
wealth	-5.199e-07	1.532e-06	-0.3395	0.7343	-3.522e-06	2.482e-06
drinkly	0.2121	0.1907	1.1120	0.2662	-0.1618	0.5860
mchild	0.0954	0.1219	0.7829	0.4337	-0.1435	0.3342
mhrsusu	0.0087	0.0066	1.3349	0.1819	-0.0041	0.0216
mhsize	0.1311	0.0769	1.7044	0.0883	-0.0197	0.2819
mintmonth	0.2137	0.1478	1.4456	0.1483	-0.0761	0.5035
mmarried	-0.6785	0.5740	-1.1821	0.2372	-1.8035	0.4466
mretired	-0.7021	0.4050	-1.7335	0.0830	-1.4959	0.0918
mwealth	-6.697e-06	3.902e-06	-1.7164	0.0861	-1.435e-05	9.513e-07
mdrinkly	-1.5217	0.2869	-5.3037	0.0000	-2.0841	-0.9593

1.7 Comparacion de modelos

[40]:
```
print(lmp.compare({"FE": fe, "RE": re, "CRE": cre}))
```

Model Comparison

	FE	RE	CRE
Dep. Variable	cesd	cesd	cesd

Estimator	PanelOLS	RandomEffects	RandomEffects
No. Observations	10038	10038	10038
Cov. Est.	Robust	Robust	Robust
R-squared	0.0039	0.0103	0.0168
R-Squared (Within)	0.0039	0.0009	0.0039
R-Squared (Between)	0.0125	0.0284	0.0417
R-Squared (Overall)	0.0096	0.0192	0.0290
F-statistic	3.3026	13.041	10.711
P-value (F-stat)	0.0009	0.0000	0.0000
const	9.8949	9.9065	8.6171
	(14.782)	(21.295)	(7.6421)
wealth	-5.199e-07	-1.541e-06	-5.199e-07
	(-0.6169)	(-1.3688)	(-0.3395)
child	0.1493	0.2300	0.1493
	(1.5553)	(4.2010)	(1.4835)
hrsusu	-0.0004	0.0016	-0.0004
	(-0.1408)	(0.6058)	(-0.1403)
hsize	-0.1204	-0.0721	-0.1204
	(-2.7241)	(-2.0476)	(-2.7045)
intmonth	-0.0177	0.0039	-0.0177
	(-0.3492)	(0.0841)	(-0.3592)
married	-1.1832	-1.5836	-1.1832
	(-2.3362)	(-6.3695)	(-2.4295)
retired	0.3476	0.0863	0.3476
	(1.7192)	(0.4999)	(1.7166)
drinkly	0.2121	-0.4231	0.2121
	(1.1246)	(-2.9607)	(1.1120)
mchild			0.0954
			(0.7829)
mhrsusu			0.0087
			(1.3349)
mhsize			0.1311
			(1.7044)
mintmonth			0.2137
			(1.4456)
mmarried			-0.6785
			(-1.1821)
mretired			-0.7021
			(-1.7335)
mwealth			-6.697e-06
			(-1.7164)
mdrinkly			-1.5217
			(-5.3037)
Effects	Entity		

```
T-stats reported in parentheses
```

1.8 RESPUESTAS

1.8.1 Eliminacion de missing

Se opto por eliminar los datos faltantes que era una cantidad de 21 datos en total 7 por cada wave para no dificultar el analisis , si bien es cierto hay que hacer estudios de si estas perdidas de datos surgieron de forma aleatoria o no , para este caso optamos por solo eliminarlos y asi tener un panel de datos balanceados para obtener mejores resultados.

1.8.2 Pooled ols

Pooled OLS

Para aplicar el modelo pooled OLS se deben cumplir algunos supuestos que para datos de panel son complicados de satisfacer.

Podemos visualizar a través de la prueba de breusch_pagan_test la cual busca probar que, si la varianza de los errores de una regresión depende de los valores de las variables independientes , en tal caso se esta en presencia de heterocedasticidad , así , el modelo presenta heterocedasticidad implicando que los estimadores de pooled OLS no sean eficientes.

Además la prueba de Durbin Watson la cual está lejos de 2, indica que hay presencia de auto-correlación en los residuos, lo cual indica que los errores estándar expresados por el modelo están subestimados.

Posee un R cuadrado ajustado de 2.1% el cual es un valor muy bajo para este conjunto de datos, la prueba F que mide la significancia conjunta de los parámetros del modelo ha dado 28.07 con un p de 1.34e-43 por lo cual las estimaciones de los coeficientes son conjuntamente significativas

Por las pruebas de Omnibus y Jarque Bera no podemos decir que los residuos se distribuyan normalmente

A raíz del análisis, este modelo no es el mejor para modelar los datos de estudios.

Coeficientes:

Podemos ver que hay 4 variables que son no significativas para el modelo dado su valor P estas son (“hrsusu” ,”hsize”,”intmonth”,”retired”)

Se aprecia que a medida que aumenta el número de hijos el puntaje de salud mental tambien lo hace

El coeficiente para la variable married indica que si la persona está casada el puntaje de salud mental disminuye sustancialmente.

Lo mismo para la variable drinkly si la persona encuestada bebio alchol en el último mes el puntaje de escala de salud mental disminuye considerablemente.

La variable “wealth” que inda la riqueza neta es insignificativa para el modelo en terminos de la magnitud del coeficiente

El modelo entrega varios coeficientes no significativos con valores P mayores a 0.05, lo cual puede ocurrir por los problemas de no cumplimiento de algunos supuestos.

1.8.3 Efectos Fijos

Este modelo indica que hay menos variables significantes para el modelo en términos de valores P que el modelo Pooled OLS

Posee un R cuadrado de 0.3% el cual es un valor muy bajo, la prueba F que prueba si todos los coeficientes del modelo son conjuntamente significativos es de 3.3026 con un p de 0.0009 por lo cual las estimaciones de los coeficientes son conjuntamente significativas

Para este modelo la variable "hsize" tamaño del hogar es significativa y se puede indicar que a medida que aumenta el tamaño del hogar en el tiempo el puntaje de salud mental disminuye.

Lo mismo ocurre con la variable "retired" ya que si bien su valor p no es inferior a 0.05 se acerca mucho, y el cual indica que si la persona esta pensionada aumenta el puntaje de salud mental.

La variable married sigue siendo significativa, aunque disminuyo la magnitud de su coeficiente sin embargo sigue siendo considerable.

Este método es útil para estimar modelos de datos de panel donde se tienen efectos inobservables. además, la ventaja de este método es que el estimador de efectos fijos permite la correlación arbitraria entre los errores que son fijos en el tiempo y las variables explicativas en cualquier periodo, por lo que si se piensa que los errores inobservables están correlacionados serialmente con las variables explicativas es recomendable utilizar este método.

Los efectos fijos requieren de menos supuestos para ser empleados, sin embargo, tienen la desventaja de utilizar solo la variación de cada individuo en el tiempo (removiendo las diferencias promedio entre individuos) por lo que puede ocasionar la eliminación de muchas variables si es que las variables de los datos no cambian en el tiempo. Esto ocurre en este set de datos para las variables "Age", "Urban", "SCHADJ", "Female" las cuales no se pueden incorporar en el modelo de efectos fijos debido a que son constantes en el tiempo por lo que deben ser eliminadas.

1.8.4 Efectos aleatorios

Este modelo contiene variables significativas diferentes a las que tenía el modelo de efectos fijos, se puede apreciar que el coeficiente de la variable child es muy parecido al del modelo pooled OLS.

Posee un R cuadrado de 1.03% el cual es un valor muy bajo, la prueba F que prueba si todos los coeficientes del modelo son conjuntamente significativos es de 13.041 con un p de 0.0000 por lo cual las estimaciones de los coeficientes son conjuntamente significativas

Para este modelo la variable hsize igual es significativa y su coeficiente negativo indica que a medida que aumenta el tamaño de la casa disminuye el puntaje de salud mental sin embargo lo hace en una menor proporción que si se estimase con el modelo de efectos fijos.

La variable married sigue la misma línea de los modelos anteriores en donde si la persona esta casa disminuye el puntaje de salud mental.

Ocurre lo mismo con La variable "drinkly" , donde si la persona bebió alcohol el último mes, disminuye el puntaje de salud mental.

El estimador de efectos aleatorios de la sección es atractivo cuando se piensa que el efecto inobservable no está correlacionado con ninguna de las variables explicativas. Tiene la ventaja de que se pueden incluir variables que no cambian con el tiempo considerando si que no deben estar correlacionados con el efecto inobservable.

. Si se tienen buenos controles en la ecuación, se podría considerar que cualquier heterogeneidad sobrante olvidada sólo induce correlación serial en el término del error compuesto, pero no genera correlación entre los errores compuestos y las variables explicativas.

1.8.5 Comparacion

Si se quieren estudiar solo variables que cambian en el tiempo para explicar la variable dependiente Y, es recomendable utilizar el modelo efectos fijos ya que este modelo permite que haya correlación entre los efectos fijos no observados y las variables explicativas , lo cual en la practica es mas probable de que ocurra.

Ahora si se tienen demasiadas variables explicativas constantes en el tiempo y se quiere explicar la variable dependiente Y, es recomendable utilizar los efectos aleatorios por sobre los efectos fijos ya que estos no pueden incluir este tipo de variables en el análisis. Además, es preferible un modelo de efectos aleatorios que pooled OLS ya que por lo general los estimadores de los efectos aleatorios son más eficientes, esto también se evidencia en este set de datos ya que se rechaza el test de breusch_pagan probando que los estimadores de pooled OLS no son eficientes.

1.8.6 Test hausman

Investigadores recomiendan utilizar las estimaciones de efectos aleatorios a menos que la prueba de Hausman lo rechace. El rechazo de la prueba de Huasman significa que el supuesto clave del modelo efecto aleatorio de que las variables explicativas y los efectos no observados no están correlacionados es falso.

Por ende, si se rechaza la hipótesis nula de la prueba de Hausman se recomienda utilizar el modelo de efectos fijos ya que presenta estimadores consistentes y posiblemente eficientes, por otro lado los estimadores del efecto aleatorio son consistentes pero ineficientes. De no tener las pruebas necesarias para rechazar la hipótesis nula de la prueba de Hausman se recomienda utilizar el modelo de efectos aleatoria que presenta estimadores consistentes y eficientes versus los estimadores de los efectos fijos que son inconsistentes.

Para este set de datos en especifico se rechazo la hipotesis nula del Test Hausman por lo que es prefereible utilizar el modelo de efectos fijos por sobre el de efectos aleatorios

1.8.7 CRE

Como podemos observar el modelo entrega coeficientes muy parecidos a los que entrega el modelo de efectos fijos. Como la gran mayoría de las medias de las variables son no significativas al tener estadísticos t de magnitud menor a 2, el modelo no está haciendo el mejor trabajo para capturar la heterogeneidad no observada

El componente no observado presenta autocorrelación con la variable explicativa, se puede apreciar que la heterogeneidad fija en el tiempo tiene un impacto en el puntaje CESD ya que las variables explicativas están correlacionadas con estos fenómenos no observados y por ende impactan en el puntaje CESD. Además las medias al no ser todas significativas el modelo CRE no está haciendo el mejor trabajo para capturar la heterogeneidad no observada

Debido a la similitud de los coeficientes entre los efectos fijos y el modelo CRE, nos inclinamos por utilizar el modelo CRE ya que es más completo debido a que se pueden incorporar variables que

en el efecto fijo no se permiten, así pudiendo integrar mayor información que puede ser relevante para el análisis.