Unit 5 E10.2

January 9, 2021

Econometrics II, Bachelor degree in Economics

Universitat Autònoma de Barcelona

Student, Roylan Martinez Vargas

Professor, Michael Creel

NIU: 1539069

E10.2 Do citizens demand more democracy and political freedom as their incomes grow? That is, is democracy a normal good? On the textbook website, **www.pearsonglobaleditions.com/Stock_Watson**, you will find the data file **Income_Democracy**, which contains a panel data set from 195 countries for the years 1960, 1965, . . . , 2000. A detailed description is given in **Income_Democracy_Description**, available on the website.[4] The dataset contains an index of political freedom/democracy for each country in each year, together with data on the country's income and various demographic controls. (The income and demographic controls are lagged five years relative to the democracy index to allow time for democracy to adjust to changes in these variables.)

a. Is the data set a balanced panel? Explain.

b. The index of political freedom/democracy is labeled *Dem_ind*.

i. What are the minimum and maximum values of *Dem_ind* in the data set? What are the mean and standard deviation of *Dem_ind*

in the data set? What are the 10th, 25th, 50th, 75th, and 90th percentiles of its distribution?

ii. What is the value of *Dem_ind* for the United States in 2000? Averaged over all years in the data set?

iii. What is the value of *Dem_ind* for Libya in 2000? Averaged over all years in the data set?

iv. List five countries with an average value of *Dem_ind* greater than 0.95; less than 0.10; and between 0.3 and 0.7.

c. The logarithm of per capita income is labeled *Log_GDPPC*. Regress *Dem_ind* on *Log_GDPPC*. Use standard errors that are clustered by country.

i. How large is the estimated coefficient on *Log_GDPPC*? Is the coefficient statistically significant?

ii. If per capita income in a country increases by 20%, by how much is *Dem_ind* predicted to increase? What is a 95% confidence interval for the prediction? Is the predicted increase in *Dem_ind* large or small? (Explain what you mean by large or small.)

iii. Why is it important to use clustered standard errors for the regression? Do the results change if you do not use clustered standard errors?

d. i. Suggest a variable that varies across countries but plausibly varies little—or not at all—over time and that could cause omitted variable bias in the regression in (c).

ii. Estimate the regression in (c), allowing for country fixed effects. How do your answers to (c)(i) and (c)(ii) change?

iii. Exclude the data for Azerbaijan and rerun the regression. Do the results change? Why or why not?

iv. Suggest a variable that varies over time but plausibly varies little—or not at all—across countries and that could cause omitted variable bias in the regression in (c).

v. Estimate the regression in (c), allowing for time and country fixed effects. How do your answers to (c)(i) and (c)(ii) change?

vi. There are addition demographic controls in the data set. Should these variables be included in the regression? If so, how do the results change when they are included?

e. Based on your analysis, what conclusions do you draw about the effects of income on democracy?

[21]:
```
# TOOLS
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
import statsmodels.api as sm
from linearmodels import PanelOLS, PooledOLS
```

Notes

The original data includes the variables dt_n where n varies from 1 to 9 but these variables are dummy equivalent variables of the years and therefore I did not include them in this analysis.

[22]:
```
# Data without null values and with the obs column as the index
data = pd.read_csv("income_democracy.csv")
data.head(1)
```

[22]:
```
    country  year  dem_ind  log_gdppc  log_pop  age_1  age_2  age_3  age_4  \
0  Andorra  1960      NaN        NaN      NaN    NaN    NaN    NaN    NaN

   age_5  …  code  dt_1  dt_2  dt_3  dt_4  dt_5  dt_6  dt_7  dt_8  dt_9
```

```
0    NaN  …   1.0     1     0     0     0     0     0     0     0     0

[1 rows x 22 columns]
```

a.

[23]:
```
# Check if the countries have null values
data.isna().values.any()
```

[23]:
```
True
```

The prior test returns **True** if there are missing values in the data set. As we see, it returned **True** because there are missing values and therefore the data set is not balanced[1].

b.i

[24]:
```
# Maximum and minimum values of the Dem_ind variable
data.dem_ind.describe(percentiles=[.1, .25, .5, .75, .9])
```

[24]:
```
count    1266.000000
mean        0.499073
std         0.371337
min         0.000000
10%         0.000000
25%         0.166667
50%         0.500000
75%         0.833333
90%         1.000000
max         1.000000
Name: dem_ind, dtype: float64
```

The variable `dem_ind` has a minimum value of **0.000000** and a maximum value of **1.000000**. The mean is **0.499073** with a standard deviation of **0.371337**. The 10th percentile is **0.000000**, the 25th is **0.166667**, the 50th is **0.500000**, the 75th **0.833333** and the 90th **1.000000**.

b.ii

[25]:
```
# value of dem_ind for the United States in 2000
print(data[(data.country == 'United States') & (data.year == 2000)].dem_ind)

# Mean all the years
print(data[data.country == 'United States'].dem_ind.mean())
```

[1] If this test returns **False**, it does not mean the data is balanced.

```
1601    1.0
Name: dem_ind, dtype: float64
0.9855555560853746
```

The value of variable `dem_ind` for the United States in 2000 is **1.0** and the average for all the years of this country in the data set is **0.9855555560853746**

b.iii

[26]:
```
# value of dem_ind for the United States in 2000
print(data[(data.country == 'Libya') & (data.year == 2000)].dem_ind)

# Mean all the years
print(data[data.country == 'Libya'].dem_ind.mean())
```

```
917    0.0
Name: dem_ind, dtype: float64
0.10925926764806122
```

The value of variable `dem_ind` for Libya in 2000 is **0.0** and the average for all the years of this country in the data set is **0.10925926764806122**

b.iv

[27]:
```
# Making groups of the given filters by the exercise (data.groupby(['country']).
↪mean())
def five_countries():
    average, greater95, less10, between3_7 = 0, [], [], []

    for country in data.country.unique():
        average = data[data.country == country].dem_ind.mean()

        if average > 0.95 and len(greater95) != 5:
            greater95.append(country)

        elif average < 0.1 and len(less10) != 5:
            less10.append(country)

        elif 0.3 < average and average < 0.7 and len(between3_7) != 5:
            between3_7.append(country)

        if len(greater95) + len(less10) + len(between3_7) == 15:
            return greater95, between3_7, less10


five_countries()
```

[27]:
```
(['Australia', 'Austria', 'Belgium', 'Belize', 'Barbados'],
 ['Argentina', 'Armenia', 'Antigua', 'Bangladesh', 'Bulgaria'],
 ['Afghanistan', 'Angola', 'Burundi', 'Brunei', 'China'])
```

Australia, Austria, Belgium, Belize and Barbados are countries with an average value of `dem_ind` greater than **0.95**.
Argentina, Armenia, Antigua, Bangladesh, Bulgaria are countries with an average value of `dem_ind` greater between **0.3** and **0.7**.
Afghanistan, Angola, Burundi, Brunei, China are countries with an average value of `dem_ind` lower than **0.1**.

c.i

[51]:
```
# Data adjustment for the analysis
dci = data[['year', 'country', 'dem_ind','log_gdppc', 'code']].dropna()
dci['year'] = pd.to_datetime(dci.year)
dci = dci.set_index(['code', 'year'])

# panel ols
mdci = PooledOLS.from_formula('dem_ind ~ 1 + log_gdppc', data = dci).
 ↪fit(cov_type='clustered', clusters=dci.country)
print(mdci)
```

```
                          PooledOLS Estimation Summary
================================================================================
Dep. Variable:                dem_ind   R-squared:                        0.4385
Estimator:                  PooledOLS   R-squared (Between):              0.5353
No. Observations:                 958   R-squared (Within):              -0.0451
Date:                Sat, Jan 09 2021   R-squared (Overall):              0.4385
Time:                        14:33:14   Log-likelihood                   -110.72
Cov. Estimator:            Clustered
                                        F-statistic:                      746.48
Entities:                         150   P-value                           0.0000
Avg Obs:                       6.3867   Distribution:                  F(1,956)
Min Obs:                       1.0000
Max Obs:                       9.0000   F-statistic (robust):             398.65
                                        P-value                           0.0000
Time periods:                       9   Distribution:                  F(1,956)
Avg Obs:                       106.44
Min Obs:                       54.000
Max Obs:                       148.00

                             Parameter Estimates
================================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept     -1.3548     0.1001    -13.530     0.0000     -1.5513     -1.1583
```

```
log_gdppc      0.2357      0.0118      19.966      0.0000      0.2125      0.2588
==============================================================================
```

The estimated `log_gdppc` coefficient is **0.2357** and it is statisticaslly significant with a 99.99% confidence level.

c.ii

[29]:
```
0.2 * mdci.params[1], mdci.params[1]
```

[29]:
```
(0.047134621766962044, 0.2356731088348102)
```

If per capita income in a country increases by 20%, then `dem_ind` is expected to increase by **0.047134621766962044** $(0.2 * 0.2356731088348102)$

[30]:
```
# 95% Confidence interval of the predicted increase
print(mdci.conf_int(0.95)[-1:] * 0.2)
```

```
               lower     upper
log_gdppc   0.042502  0.051767
```

The confidence interval of the predicted increase of `dem_ind` is between **0.042502** and **0.051767**. The model predicts that a 20% change in `log_gdppc` leads to an increase of arround **0.047** of the variable `dem_ind`, since this variable is between 0 and 1 we can argue that a 20% change leads to a increase of 4.7 percentage points[2] and therefore we can argue that the predicted increase is relatively a small change.

c.iii

[62]:
```
# Model without clustered standard errors
mdciii = PooledOLS.from_formula('dem_ind ~ 1 + log_gdppc', data = dci).fit()
print(0.2 * mdciii.params[1], mdciii.params[1])
print(mdciii.conf_int(0.95).iloc[1] * 0.2)
```

```
0.047134621766962044 0.2356731088348102
lower    0.043749
upper    0.050520
Name: log_gdppc, dtype: float64
```

The results actually do change. The coefficient regressor of `dem_ind` does not change (it is still **0.2356731088348102**). The standard error of the expected change when `log_gdppc` changes by 20% is wider when there are clustered standard errors, since now the 95% confidence interval of the prediction goes from 0.043749 to 0.050520. I think it is important since it helps to account (prevent possible bias) for the possible differences the variable `dem_ind` has associated to the cluster (`country`).
In this example there is not too much difference but there are some cases in which the serial correlation might lead to highly misleading conclusions.

[2] Note I have written **percentage points** instead of percent

d.i

[32]:
```
# standard deviation of the variables that change over time without taking into␣
↪account code and year
ddi = data[np.delete(data.columns, [1, 12])].groupby(['country']).std().
↪transpose().dropna(axis=1)

# mean of the standard deviation of the variables
ddi['mean'] = ddi.mean(numeric_only=True, axis=1)

# variables with the minimum mean of the standard deviation of the variables
var_min_mean = ddi.index[np.where(ddi['mean'] == ddi['mean'].min())[0]]
print(var_min_mean[0])

# Check if the last result varies across countries in at least one situation
ddi2 = data[np.delete(data.columns, [1, 12])].groupby(['country']).mean().
↪transpose().dropna(axis=1)
ddi2['std'] = ddi2.std(numeric_only=True, axis=1)
print(ddi2.loc[var_min_mean[0], 'std'] != 0)
```

```
age_4
True
```

As we just computed, regardless of the variables `code`, `year` and `country`, `age_4` is the variable that varies across countries in at least one situation[3] with the minimum variation over time. It is a good candidate to cause omitted variable bias in the regression in **c**.

d.ii

[33]:
```
# Fixed effects
mdii = PanelOLS.from_formula("dem_ind ~ 1 + log_gdppc + EntityEffects", dci).
↪fit(cov_type='clustered', cluster_entity=True)
print(mdii)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                dem_ind   R-squared:                        0.0197
Estimator:                   PanelOLS   R-squared (Between):              0.3031
No. Observations:                 958   R-squared (Within):               0.0197
Date:                Sat, Jan 09 2021   R-squared (Overall):              0.2562
Time:                        14:27:40   Log-likelihood                   247.99
Cov. Estimator:            Clustered
                                        F-statistic:                      16.202
Entities:                         150   P-value                           0.0001
Avg Obs:                       6.3867   Distribution:                  F(1,807)
Min Obs:                       1.0000
```

[3] *The last* `True` *result checks if the variable actually varies across countries in at least one situation.*

```
Max Obs:                        9.0000   F-statistic (robust):             7.0957
                                         P-value                           0.0079
Time periods:                      958   Distribution:                  F(1,807)
Avg Obs:                        1.0000
Min Obs:                        1.0000
Max Obs:                        1.0000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept     -0.1153     0.2565    -0.4496     0.6531     -0.6187      0.3881
log_gdppc      0.0837     0.0314     2.6638     0.0079      0.0220      0.1454
==============================================================================

F-test for Poolability: 6.0368
P-value: 0.0000
Distribution: F(149,807)

Included effects: Entity
```

The estimated `log_gdppc` coefficient is **0.0837** and it is statisticaslly significant with a 99.21% confidence level $(1 - 0.0079)$.

[34]:
```
0.2 * mdii.params[1], mdii.params[1]
```

[34]:
```
(0.01674820065176016, 0.0837410032588008)
```

If per capita income in a country increases by 20%, then `dem_ind` is expected to increase by **0.01674820065176016** $(0.2 * 0.0837410032588008)$

[35]:
```
# 95% Confidence interval of the predicted increase
print(mdii.conf_int(0.95)[-1:] * 0.2)
```

```
              lower    upper
log_gdppc  0.004407  0.02909
```

The confidence interval of the predicted increase of `dem_ind` is between **0.004407** and **0.02909**. The model predicts that a 20% change in `log_gdppc` leads to an increase of arround **0.01674820065176016** of the variable `dem_ind`, since this variable is between 0 and 1 we can argue that a 20% change leads to a increase of 1.67 percentage points[4] and therefore we can argue that the predicted increase is even smaller than the expected percentage point increase of **(c)(ii)**.

[4] Note again that I have written **percentage points** instead of percent

d.iii

[36]:
```
# Data without Azerbaijan
diii = dci[dci.country != 'Azerbaijan']

# Fixed effects
mdiii = PanelOLS.from_formula("dem_ind ~ 1 + log_gdppc + EntityEffects", diii).
 ↪fit(cov_type='clustered', cluster_entity=True)
print(mdiii)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                 dem_ind   R-squared:                        0.0197
Estimator:                   PanelOLS   R-squared (Between):              0.3045
No. Observations:                 957   R-squared (Within):               0.0197
Date:                Sat, Jan 09 2021   R-squared (Overall):              0.2563
Time:                        14:27:42   Log-likelihood                   247.23
Cov. Estimator:            Clustered
                                        F-statistic:                      16.202
Entities:                         150   P-value                           0.0001
Avg Obs:                       6.3800   Distribution:                   F(1,807)
Min Obs:                       0.0000
Max Obs:                       9.0000   F-statistic (robust):             7.0957
                                        P-value                           0.0079
Time periods:                     958   Distribution:                   F(1,807)
Avg Obs:                       0.9990
Min Obs:                       0.0000
Max Obs:                       1.0000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept     -0.1149     0.2565    -0.4481     0.6542     -0.6184      0.3885
log_gdppc      0.0837     0.0314     2.6638     0.0079      0.0220      0.1454
==============================================================================

F-test for Poolability: 6.0639
P-value: 0.0000
Distribution: F(148,807)

Included effects: Entity
```

As we see the results remain almost the same, this can be argue because Azerbaijan just has one year observation in the sample and therefore it is a minimal loss of data. This can be easily seen by looking at the observation that went from **958** to **957** without Azerbaijan.

d.iv

[39]:
```
# mean of the variables with respect of time for every country.
div = data[np.delete(data.columns, [1] + list(np.arange(12, 22)))].
 ↪groupby(['country']).mean().transpose().dropna(axis=1)

# standard deviation of the mean of the variables for every country.
div['std'] = div.std(numeric_only=True, axis=1)

# variable with the minimum standard deviation of the mean of the variables for␣
 ↪every country
var_min_std2 = div.index[np.where(div['std'] == div['std'].min())[0]]
print(var_min_std2)

# Check if the these variables vary over time in at least one country
div2 = data[np.delete(data.columns, [1] + list(np.arange(12, 22)))].
 ↪groupby(['country']).mean().transpose().dropna(axis=1)
div2['max'] = div2.max(numeric_only=True, axis=1)
print(0 not in div2.loc[var_min_std2, :])
```

```
Index(['age_2'], dtype='object')
True
```

As we just computed, regardless of the variables `code`, `year` and `country`, the variable `age_2` is the variable that vary over time in at least one country[5] with the minimum variation across countries. It is also good candidate to cause omitted variable bias in the regression in **c**.

d.v

[64]:
```
# Fixed effects
mdv = PanelOLS.from_formula("dem_ind ~ 1 + log_gdppc + EntityEffects +␣
 ↪TimeEffects", dci).fit(cov_type='clustered', cluster_entity=True)
print(mdv)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                dem_ind   R-squared:                        0.0046
Estimator:                   PanelOLS   R-squared (Between):              0.2080
No. Observations:                 958   R-squared (Within):               0.0171
Date:                Sat, Jan 09 2021   R-squared (Overall):              0.1767
Time:                        14:52:09   Log-likelihood                    298.75
Cov. Estimator:            Clustered
                                        F-statistic:                      3.7289
Entities:                         150   P-value                           0.0538
Avg Obs:                       6.3867   Distribution:                  F(1,799)
Min Obs:                       1.0000
```

[5] *The last* `True` *result checks if the variable actually varies across countries in at least one situation.*

```
Max Obs:                        9.0000   F-statistic (robust):             1.3519
                                         P-value                           0.2453
Time periods:                        9   Distribution:                  F(1,799)
Avg Obs:                        106.44
Min Obs:                        54.000
Max Obs:                        148.00

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept      0.1307     0.3760     0.3476     0.7283     -0.6074      0.8688
log_gdppc      0.0536     0.0461     1.1627     0.2453     -0.0369      0.1441
==============================================================================

F-test for Poolability: 6.8754
P-value: 0.0000
Distribution: F(157,799)

Included effects: Entity, Time
```

The estimated `log_gdppc` coefficient is **0.0536** and it is statisticaslly significant with a 99.21% confidence level $(1 - 0.0079)$.

[65]:
```
0.2 * mdv.params[1], mdv.params[1]
```

[65]:
```
(0.010717556426210176, 0.05358778213105088)
```

If per capita income in a country increases by 20%, then `dem_ind` is expected to increase by **0.010717556426210176** $(0.2 * 0.05358778213105088)$

[66]:
```
# 95% Confidence interval of the predicted increase
print(mdv.conf_int(0.95)[-1:] * 0.2)
```

```
               lower     upper
log_gdppc -0.007376  0.028812
```

The confidence interval of the predicted increase of `dem_ind` is between **-0.007376** and **0.028812**. The model predicts that a 20% change in `log_gdppc` leads to an increase of arround **0.010717556426210176** of the variable `dem_ind`, since this variable is between 0 and 1 we can argue that a 20% change leads to a increase of 1.07 percentage points[6] and therefore we can argue that the predicted increase is even smaller than the expected percentage point increase of **(c)(ii)**.

[6] Note again that I have written **percentage points** instead of percent

d.vi

[67]:
```
# Data adjustment for the analysis
dvi = data[['year', 'country', 'dem_ind','log_gdppc', 'code', 'age_1', 'age_2',␣
 ↪'age_3', 'age_4', 'age_5',]].dropna()
dvi['year'] = pd.to_datetime(dvi.year)
dvi = dvi.set_index(['code', 'year'])

# Fixed effects
mdvi = PanelOLS.from_formula("dem_ind ~ 1 + log_gdppc + EntityEffects +␣
 ↪TimeEffects + age_1 + age_2 + age_3 + age_4 + age_5", dvi).
 ↪fit(cov_type='clustered', cluster_entity=True)
print(mdvi)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                dem_ind   R-squared:                        0.0262
Estimator:                   PanelOLS   R-squared (Between):              0.2986
No. Observations:                 932   R-squared (Within):              -0.0060
Date:                Sat, Jan 09 2021   R-squared (Overall):              0.2332
Time:                        14:52:25   Log-likelihood                   298.04
Cov. Estimator:            Clustered
                                        F-statistic:                      3.4721
Entities:                        144   P-value                           0.0022
Avg Obs:                      6.4722   Distribution:                  F(6,774)
Min Obs:                      1.0000
Max Obs:                      9.0000   F-statistic (robust):             1.4770
                                        P-value                           0.1831
Time periods:                     9   Distribution:                  F(6,774)
Avg Obs:                     103.56
Min Obs:                     54.000
Max Obs:                     142.00

                             Parameter Estimates
================================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept   1.192e+05  1.092e+05     1.0912     0.2755  -9.519e+04   3.335e+05
log_gdppc      0.0704     0.0476     1.4790     0.1395     -0.0230      0.1639
age_1      -1.192e+05  1.092e+05    -1.0913     0.2755  -3.335e+05   9.519e+04
age_2      -1.192e+05  1.092e+05    -1.0913     0.2755  -3.335e+05   9.519e+04
age_3      -1.192e+05  1.092e+05    -1.0913     0.2755  -3.335e+05   9.519e+04
age_4      -1.192e+05  1.092e+05    -1.0913     0.2755  -3.335e+05   9.519e+04
age_5      -1.192e+05  1.092e+05    -1.0912     0.2755  -3.335e+05   9.519e+04
================================================================================

F-test for Poolability: 6.6829
```

```
P-value: 0.0000
Distribution: F(151,774)

Included effects: Entity, Time
```

The results show that, fixing effects for countries and time, the variables `age1`, `age_2`, `age_3`, `age_4` and `age_5` do not seem to be relevant since their coefficient regressors is extremely small and the p-value is quite low. The cofficient regressor of `log_gdppc` has become smaller because there are more regressors in the model but it is still more relevant than the demographic regressors. The p-value of the `log_gdppc` is lower than the p-value of the demographic regressors and therefore it is more trustable.
As we see the results do change but to worst results, therefore these demographic variables should not be included in the model.

e

[84]:
```
print(mdii.params[1], mdv.params[1])
print(mdii.params[1] * 0.05, mdv.params[1] * 0.05)
print(mdii.conf_int(0.99)[-1:])
print(mdii.conf_int(0.99)[-1:] * 0.05)
```

```
0.0837410032588008 0.05358778213105088
0.00418705016294004 0.002679389106552544
              lower     upper
log_gdppc  0.002573  0.164909
              lower     upper
log_gdppc  0.000129  0.008245
```

The data shows evidence that income per capita does affect democracy. The model of the exercise **(d)(ii)** indicates that a 5% increase in the gdp[7] is related to an expected increase of the democracy index of about **0.004185** ($0.083741 * 0.05$) or equivalently 0.4185 percentage points since it the democracy index goes from 0 to 1.
This relationship between income and democracy seems to be strong but this model shows no causality, in other words, we do not know if for an increase in democracy there will be a increase in income or if an increase of democracy expects an increase of income.
The model of the exercise **(d)(v)** (fixed effects for years and country) predicts that a 5% increase in the gdp is related to an expected increase of the democracy index of about **0.0026794** ($0.053588 * 0.05$). Again it is not related to causality and the model is slightly worst than the previus one.
The conclusion is that income is definitely related with a positive improvement of the democracy index but since this data is exposed to sample specific features, the models here do not take into account many important confounding variables and the income does not completely explain all the variance in the models; the real change in democracy related to an increase in the income is probable arround but not exactly the predicted change in **(d)(ii)**, so:

$\frac{\Delta DemocracyIndex}{\%\Delta Income_{GDPPC}} \approx 0.083741$

[7] 5% increase is generally seen as as healthy achivable increase in the gdp.