Stern's Replication-Final Version

June 27, 2024

1 Replication of *'Modelling the emissions-income relationship using long-run growth rates'* by David I. Stern, Reyer Gerlagh, Paul J. Burke

The idea of this work is to replicate the regression output tables presented in the paper. The sequence of the tables and it's corresponding regression equations are given below for your convenience.

1.0.1 Data

The data file and the RATS (Regression Analysis of Time Series) code have been downloaded from Stern's official website (http://www.sterndavidi.com/datasite.html).

[12]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML


df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='CO2 1971-2010')

# Display the first few rows of the DataFrame
print(df.head())
```

```
  ISO3CODE                CNAME     GDPPC1971  LNGDPPC1971  DMLNGDPPC1971  \
0      AGO               Angola   2390.478101     7.779249      -0.319958
1      ALB              Albania   3728.588935     8.223785       0.124578
2      ARG            Argentina   3047.466797     8.022066      -0.077141
3      ATG  Antigua and Barbuda   4391.901990     8.387518       0.288311
4      AUS            Australia  17143.668520     9.749384       1.650177

    GRGDPPC  DMMULGDP    MULGDP         EPC1971  LNEPC1971  …  DMLNFFPC1971  \
0  0.006774 -0.002167  0.052697    563.898623   6.334874  …      4.464793
1  0.019730  0.002458  0.162253   1988.773445   7.595273  …      3.001371
2  0.011180 -0.000862  0.089685   3648.630633   8.202107  …      3.234411
```

```
3  0.028881  0.008327  0.242239   6391.381435   8.762706  …     -4.230234
4  0.017044  0.028126  0.166172  11762.885257   9.372705  …      6.174614

   PWTPOP1971        AREA  lNPOPD1971  DMLNPOPD1971  MAF  REF  LAM  OECD90  \
0    6.047736  1246700.0    1.579184     -2.043424    1    0    0       0
1    2.188650    27400.0    4.380497      0.757889    0    1    0       0
2   24.376109  2736690.0    2.186854     -1.435754    0    0    1       0
3    0.066554      440.0    5.018994      1.396386    0    0    1       0
4   12.987847  7682300.0    0.525095     -3.097513    0    0    0       1

   ASIA
0     0
1     0
2     0
3     0
4     0

[5 rows x 33 columns]
```

[6]:
```
pip install openpyxl
```

```
Collecting openpyxl
  Downloading openpyxl-3.1.4-py2.py3-none-any.whl (251 kB)
Collecting et-xmlfile
  Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Note: you may need to restart the kernel to use updated packages.
Installing collected packages: et-xmlfile, openpyxl
Successfully installed et-xmlfile-1.1.0 openpyxl-3.1.4
```

1.1 Table 2 : General model (CO2 emission)

The regression shown below is of the general model for CO2 emission as in regression equation 2 in the original paper.

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} X_{j,i} + \epsilon_i$$

where hats indicate long-run growth rates, $ E$ is the log of emissions per-capita, and G is the log of GDP per capita. X_i is a vector of explanatory and control variables. The sample mean has been deducted from each continuous level variable in this vector.

The third term, $\hat{G}_I G_{i,0}$, is the interaction between the rate of economic growth and the initial level of log income per capita. The EKC turning point is calculated with the assumption that $\alpha_1 > 0$ and $\beta_1 < 0$ when $\frac{\partial \hat{E}}{\partial \hat{G}} = 0$ and $\tau = exp\left(-\frac{\alpha_1}{\beta_1} + \mu_G\right)$, where μ_G is the cross-country mean of initial GDP per capita variable prior to the estimation

The following variables are contained in $X_{j,i}$: Dummy variable for countries that are centrally planned or market economy, dummies for countries that have legal origins in the UK, France, Germany or any Scandinavian countries, average summer and winter temperatures, de-meaned log of fossil fuel consumption in 1971 and de-meaned log of population density in 1971.

[5]:
```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2




df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='CO2 1971-2010')

# Run OLS regression
reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC3')
print(reg1.summary())

#Different Types of Robust Standard Errors
#'HC0': The original White (1980) heteroskedasticity-consistent standard errors.
#'HC1': A small sample correction to HC0.
#'HC2': Another small sample correction, which is more reliable when the sample␣
↪size is small.
#'HC3': MacKinnon and White's (1985) heteroskedasticity-consistent standard␣
↪errors, which is more robust in small samples.

#print(reg1.resid())

#Create a matrix of independent varibales which can be used for the residual␣
↪tests
exog_het = reg1.model.exog

# Perform the Breusch-Pagan Test for Heteroskedasticity. Note exog_het is the␣
↪matrix of explanatory (independent) variables
bp_test = het_breuschpagan(reg1.resid,exog_het)

print("Breusch-Pagan Test for Heteroskedasticity:") # H0: Variance of the␣
↪residuals is constant (homoskedastic)
```

```
print (f"LM Statistics: {bp_test[0]}") # The Lagrange Multiplier (LM) test␣
↪follows a chi-sq distribution and the degree of freedom is the number of␣
↪independent variables. The LM statistic is calculated based on the residuals␣
↪from the regression model and the independent variables
print (f"LM Test p-value: {bp_test[1]}") # If p-value is less than 0.05 (5%␣
↪level) then we reject the null and vice-versa
#print (f"F-Statistics: {bp_test[2]}") # F-stat is the ratio of the mean␣
↪squared error of the model and the mean squared error of the residual.
#print (f"F-Statistics p-value: {bp_test[3]}")

# Performing the White test for heteroskedasticity
white_test = het_white(reg1.resid, exog_het) # H0: Homoskedasticity

# Extracting the test statistic and p-value
white_stat = white_test[0]
white_pvalue = white_test[1]

print(f"White test statistic: {white_stat}")
print(f"White test p-value: {white_pvalue}")

#print("White Test for Heteroskedasticity:") # Another way of writing the above␣
↪command of white test and p value
#print(f"T Statistics: {white_test[0]}")
#print(f"P-value: {white_test[1]}")
#print(f"F-statistics: {white_test[2]}")
#print(f"F-test p-value: {white_test[3]}")

# Interpretation
if white_pvalue < 0.05:  # Typically, we use a significance level of 0.05
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. No evidence␣
↪of heteroskedasticity.")

r1s= reg1.resid ** 2 # This command enables python to store the squared of␣
↪residuals in r1s

# Assuming df is your DataFrame and PWTPOP1971 is the column you want to invert
df['invpop'] = 1 / df['PWTPOP1971']

df['lpop1971'] = np.log (df['PWTPOP1971']) # To create a new varible with log

df['rls']=reg1.resid**2 # stored the residual square in the data file.

# To check the addition of the new variable invop: print(df.head())
```

```python
# Trying to run an auxiliary regression following the codes of Stern:

#reg2 = smf.ols("r1s ~ invpop", data=df).fit() (Error: Number of rows mismatch
 ↪between data argument and r1s (136 versus 134))
#print(reg2.summary())

# Test for the EKC turning point.
alpha1 = reg1.params['GRGDPPC'] # Storing the estimates of GRGDPPC
beta1 = reg1.params['DMMULGDP']
turning_point= np.exp(- alpha1/beta1 + 8.099206859) # Deriving the EKC turning
 ↪point level : From the paper (Note \mu_G = 8.09 = ln(6340.99); where 6340.99
 ↪is cross-country mean of GDPPC1971 (initial GDP per capita 1971))
print(f"EKC turning point:{turning_point}")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.727
Model:                            OLS   Adj. R-squared:                  0.700
Method:                 Least Squares   F-statistic:                     19.56
Date:                Thu, 27 Jun 2024   Prob (F-statistic):           5.22e-23
Time:                        11:29:36   Log-Likelihood:                 390.24
No. Observations:                 134   AIC:                            -754.5
Df Residuals:                     121   BIC:                            -716.8
Df Model:                          12
Covariance Type:                  HC3
===============================================================================
=
                    coef    std err          z      P>|z|      [0.025
0.975]
-------------------------------------------------------------------------------
-
Intercept        -0.0025      0.003     -0.949      0.343      -0.008
0.003
GRGDPPC           0.8901      0.107      8.282      0.000       0.679
1.101
DMMULGDP         -0.1330      0.075     -1.780      0.075      -0.280
0.013
DMLNGDPPC1971     0.0167      0.004      4.659      0.000       0.010
0.024
DMLNEPC1971      -0.0154      0.002     -7.005      0.000      -0.020
-0.011
CPE              -0.0054      0.006     -0.917      0.359      -0.017
0.006
legorfr           0.0007      0.003      0.238      0.812      -0.005
0.006
legorge           0.0006      0.005      0.127      0.899      -0.009
```

```
0.010
legorsc          -0.0038      0.005     -0.707      0.479      -0.014
0.007
DMSUMT            0.0008      0.000      2.223      0.026      0.0001
0.002
DMWINT           -0.0004      0.000     -1.674      0.094      -0.001
6.42e-05
DMLNFFPC1971      0.0011      0.000      2.889      0.004       0.000
0.002
DMLNPOPD1971     -0.0003      0.001     -0.245      0.806      -0.003
0.002
==============================================================================
Omnibus:                        6.565   Durbin-Watson:                   2.078
Prob(Omnibus):                  0.038   Jarque-Bera (JB):                6.948
Skew:                          -0.364   Prob(JB):                       0.0310
Kurtosis:                       3.845   Cond. No.                         896.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
Breusch-Pagan Test for Heteroskedasticity:
LM Statistics: 13.513843530901335
LM Test p-value: 0.33282345059096746
White test statistic: 99.90689760380853
White test p-value: 0.013479168535001177
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
EKC turning point:2648714.923934659
```

1.2 Table 3 : Restricted model: CO2 emission 1971-2010 ~ Ordas Criado et al. (2011) (Column 2)

By applying restrictions to the general model shown above, Ordas Criado et al. (2011) is estimated. Equation (3) depicts the regression equation:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model is: $\beta_1 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[43]:
```python
# Table 3 CO2 Regression Results

import numpy as np
import matplotlib.pyplot as plt
```

```python
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='CO2 1971-2010')


# Ordas-Criado Model

reg2= smf.ols("GREPC ~ GRGDPPC + DMLNGDPPC1971 + DMLNEPC1971 + CPE + legorfr +␣
↪legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')
print(reg2.summary())

# For conducting different statistical tests we have to follow the following␣
↪steps:
# 1. Create a matrix of exogenous variables

exog_het = reg2.model.exog

# 2. Run the white test: H0: Homoskedasticity

white_test = het_white(reg2.resid, exog_het)

white_pvalue = white_test[1]
# 3. Print the results

print(f"White test for heteroskedasticity:")
print(f"T-stats for white test: {white_test[0]}")
print(f"P-value: {white_test[1]}")
print(f" F statistics: {white_test[2]}")
print(f" p_value: {white_test[3]}")

if white_pvalue < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. No evidence␣
↪of heteroskedasticity")

# 3. Run the Breusch-Pagan test: H0: Homoskedasticity
```

```
# The het_breuschpagan function returns a tuple with four elements:
# a) Lagrange Multiplier (LM) Statistic: The first element, accessible with␣
↪bp_test[0], is the test statistic for the Breusch-Pagan test.
# b) p-value: The second element, accessible with bp_test[1], is the p-value␣
↪associated with the LM statistic.
# c) f-value: The third element, accessible with bp_test[2], is the f-value of␣
↪the test.
# d) f p-value: The fourth element, accessible with bp_test[3], is the p-value␣
↪associated with the f-statistic.

bp_test = het_breuschpagan(reg2.resid, exog_het)

bp_pvalue= bp_test[1]

print(f"BP test:")
print(f"LM Statistic (Chi-square): {bp_test[0]}")
print(f"LM Test p-value: {bp_test[1]}")
print(f" F statistics: {bp_test[2]}")
print(f" p_value: {bp_test[3]}")

if bp_pvalue < 0.05:
      print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
      print("Fail to reject the null hypothesis of homoskedasticity. Evidence␣
↪of no heteroskedasticity.")

# 4. Run the Harvey test (Harvey test doesn't exist in any statistical packages␣
↪in python. Thus it has to do it manually)

    # Step 1: Calculate the log of the squared residuals

df['log_sq_residuals'] = np.log(reg2.resid**2)

    # Step 2: Regress log_squared_residuals on the independent variables

harvey_model = smf.ols("log_sq_residuals ~ GRGDPPC + DMLNGDPPC1971 +␣
↪DMLNEPC1971 + CPE + legorfr + legorge + legorsc + DMSUMT + DMWINT +␣
↪DMLNFFPC1971 + DMLNPOPD1971", data=df).fit()

    # Step 3: Obtain the R-squared value from this regression

r_sq = harvey_model.rsquared

    # Step 4: Calculate the LM statistic
```

```
n = len(df) # this command helps in identifying the number of obs in the␣
↪dataset. len() ~ determines the length of various objects.
lm_stat_harvey = n * r_sq # The LM statistics is calulated as N * r_squared

    # Step 5: Calculate the p-value

p_value_harvey = chi2.sf(lm_stat_harvey, df=11) # degree of freedom (df) = 11␣
↪for eleven independent variables

    # Step 6: Print the results

print("Harvey-Godfrey test for heteroskedasticity:") # H0: Homoskedasticity
print(f" LM Statistics: {lm_stat_harvey}")
print(f"LM test p-value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no hoteroskedasticity.")


# 5. Performing the Likelihood test (LR test) against the general model.␣
↪(follows a chi-sq distribution)
# The Likelihood Ratio (LR) test is used to compare the goodness-of-fit of two␣
↪nested models. The H0: The restricted model provides a good fit.

reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

    # Calculating the log-likelihood values of both the models

llf_restricted = reg2.llf
llf_unrestricted = reg1.llf

    # Calculating the LR statistics
lr_statistic = -2 * (llf_restricted - llf_unrestricted)

    # Degrees of freedom (df): number of restrictions (difference in number of␣
↪parameters). In this case it should be 1
df_diff = reg1.df_model - reg2.df_model # Note: reg1 is the unrestricted model␣
↪and reg2 is a restricted model

    # Calculating the p_value
```

```python
p_value = chi2.sf(lr_statistic, df_diff)

print("LR test against the general model:")
print(f"LR Statistic: {lr_statistic}")
print(f"p-value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit:␣
↪The unrestricted model fits significantly better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
↪good fit: No significant improvement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.718
Model:                            OLS   Adj. R-squared:                  0.692
Method:                 Least Squares   F-statistic:                     24.79
Date:                Mon, 17 Jun 2024   Prob (F-statistic):           3.85e-26
Time:                        21:09:35   Log-Likelihood:                 388.02
No. Observations:                 134   AIC:                            -752.0
Df Residuals:                     122   BIC:                            -717.3
Df Model:                          11
Covariance Type:                  HC0
================================================================================
=
                    coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-
Intercept        -0.0039      0.002     -1.717      0.086      -0.008
0.001
GRGDPPC           0.9670      0.083     11.606      0.000       0.804
1.130
DMLNGDPPC1971     0.0156      0.003      4.976      0.000       0.009
0.022
DMLNEPC1971      -0.0157      0.002     -8.108      0.000      -0.019
-0.012
CPE              -0.0038      0.005     -0.813      0.416      -0.013
0.005
legorfr           0.0008      0.003      0.322      0.748      -0.004
0.006
legorge           0.0015      0.004      0.376      0.707      -0.006
0.009
legorsc          -0.0033      0.004     -0.779      0.436      -0.012
0.005
DMSUMT            0.0010      0.000      3.213      0.001       0.000
0.002
```

```
DMWINT            -0.0004      0.000     -1.820      0.069      -0.001
2.83e-05
DMLNFFPC1971       0.0014      0.000      4.283      0.000       0.001
0.002
DMLNPOPD1971      -0.0006      0.001     -0.573      0.567      -0.003
0.002
==============================================================================
Omnibus:                        7.830   Durbin-Watson:                   2.052
Prob(Omnibus):                  0.020   Jarque-Bera (JB):                8.966
Skew:                          -0.392   Prob(JB):                       0.0113
Kurtosis:                       3.995   Cond. No.                         747.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
White test for heteroskedasticity:
T-stats for white test: 92.3245807849109
P-value: 0.007504469440336944
 F statistics: 2.5369039604491936
 p_value: 8.793027341162336e-05
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
BP test:
LM Statistic (Chi-square): 15.490049093711212
LM Test p-value: 0.16114301505547435
 F statistics: 1.4496565478110688
 p_value: 0.1595272495911336
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
Harvey-Godfrey test for heteroskedasticity:
 LM Statistics: 11.957825790720378
LM test p-value: 0.3668216687373419
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
hoteroskedasticity.
LR test against the general model:
LR Statistic: 4.432836971044708
p-value: 0.035253863973579394
Reject the null hypothesis of restricted model provides a good fit: The
unrestricted model fits significantly better.
```

1.3 Table 3 : Restricted model: CO2 emission 1971-2010 ~ Green Solow Model (Column 3)

By applying restrictions to the general model shown above, Green Solow model (GSM) is estimated. Equation (4) shows the regression equation:

$$\hat{E}_i = \alpha_0 + \beta_2 E_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model are: $\alpha_1 = \beta_1 = \beta_3 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[13]:
```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='CO2 1971-2010')

# Table 3 CO2 Regression Results

# Green Solow model

reg3= smf.ols("GREPC ~ DMLNEPC1971 + CPE + legorfr + legorge + legorsc + DMSUMT␣
↪+ DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).fit(cov_type = 'HC0')
print(reg3.summary())

# White test for heteroskedasticity
 # Step 1: Create a matrix of exogenous variables
exog_het = reg3.model.exog

 # Follow the above steps for the White test.
white_test = het_white(reg3.resid, exog_het)

print(f"White test for heteroskedasticity: {white_test[0]}")
print(f"p-value: {white_test[1]}")
print(f" F stat: {white_test[2]}")
print(f"p-value : {white_test[3]}")

if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
```

```
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# Breusch-Pagan Test for heteroskedasticity
bp_test = het_breuschpagan(reg3.resid,exog_het)

print(f"Breusch-Pagan test - LM (chi-sq) stat: {bp_test[0]}")
print(f"p-value: {bp_test[1]}")
print(f"F stat: {bp_test[2]}")
print(f"p-value of F stat: {bp_test[3]}")

if bp_test[1] < 0.05:
    print("Rejecting the null of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null of homoskedasticity. Evidence of no␣
↪heteroskedasticity.")

# Harvey-Godfrey test. Note the python packages donot have the harvey-test␣
↪module, so have to do it manually.
# Step 1: Calculate the log of squared residuals

df['log_sq_resid'] = np.log(reg3.resid ** 2)

# Step 2: Regress the log of residuals on the independent variables

harvey_reg = smf.ols("log_sq_resid ~ DMLNEPC1971 + CPE + legorfr + legorge +␣
↪legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')

# Step 3: Calculate the r-sq value from the regression

r_sq = harvey_reg.rsquared

# Step 4: Calculate the LM statistics:

n=len(df)   # Explain the number of obs

lm_stat = n * r_sq  # the harvey test statistic is calculated as N * R^2


# Step 5: Calculate the p_value

p_value_harvey = chi2.sf(lm_stat,df=9)

print("Harvey test:")
print(f"LM (chi-sq) test statistics: {lm_stat}")
```

```python
print(f"p_value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")


# LR statistics: It is used to find the goodness of fit of restricted model␣
↪against the unrestricted model.
# The unrestricted model:

reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

# Calculate the log likelihood ratio of both the models

llf_unrestricted = reg1.llf
llf_restricted = reg3.llf

# Caculating the LR Statistics

lr_stat = - 2 * (llf_restricted - llf_unrestricted)

# Degrees of freedom (difference in the number of parameters between the␣
↪restricted and unrestricted model)

df_diff = reg1.df_model - reg3.df_model

# LR stat follows a chi-sq distribution

p_value = chi2.sf(lr_stat, df_diff)

print (f"LR test statistics against the general model: {lr_stat}")
print(f"p_value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit:␣
↪The unrestricted model fits significantly better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
↪good fit: No significan timprovement in fit for the unrestricted model.")
```

```
OLS Regression Results
```

```
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.348
Model:                            OLS   Adj. R-squared:                  0.301
Method:                 Least Squares   F-statistic:                     6.711
Date:                Tue, 18 Jun 2024   Prob (F-statistic):           8.53e-08
Time:                        21:59:33   Log-Likelihood:                 331.95
No. Observations:                 134   AIC:                            -643.9
Df Residuals:                     124   BIC:                            -614.9
Df Model:                           9
Covariance Type:                  HC0
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        0.0142      0.004      3.823      0.000       0.007       0.021
DMLNEPC1971     -0.0098      0.002     -6.299      0.000      -0.013      -0.007
CPE             -0.0053      0.006     -0.887      0.375      -0.017       0.006
legorfr         -0.0018      0.004     -0.411      0.681      -0.010       0.007
legorge         -0.0032      0.007     -0.448      0.654      -0.017       0.011
legorsc          0.0021      0.009      0.242      0.809      -0.015       0.019
DMSUMT           0.0010      0.000      2.526      0.012       0.000       0.002
DMWINT          -0.0009      0.000     -3.333      0.001      -0.001      -0.000
DMLNFFPC1971     0.0016      0.001      3.197      0.001       0.001       0.003
DMLNPOPD1971     0.0047      0.001      3.875      0.000       0.002       0.007
==============================================================================
Omnibus:                       11.030   Durbin-Watson:                   2.023
Prob(Omnibus):                  0.004   Jarque-Bera (JB):               14.594
Skew:                           0.473   Prob(JB):                     0.000677
Kurtosis:                       4.311   Cond. No.                         78.8
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
White test for heteroskedasticity: 40.282801456426604
p-value: 0.6316941370388863
 F stat: 0.8694361589339198
p-value : 0.6917665958023151
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
Breusch-Pagan test - LM (chi-sq) stat: 22.50092776639823
p-value: 0.007419972477546516
F stat: 2.780406835222417
p-value of F stat: 0.005313026608224283
Rejecting the null of homoskedasticity. Evidence of heteroskedasticity.
Harvey test:
LM (chi-sq) test statistics: 16.874183517906896
p_value: 0.05072309299337895
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
```

```
LR test statistics against the general model: 116.57928678402527
p_value: 4.207836015582442e-25
Reject the null hypothesis of restricted model provides a good fit: The
unrestricted model fits significantly better.
```

1.4 Table 3 : Restricted model: CO2 emission 1971-2010 ~ Basic EKC Model (Column 4)

By applying restrictions to the general model shown above, the basic EKC model is estimated. Equation (5) shows the regression equation:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model are: $\beta_2 = \beta_3 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[7]:
```python
# Table 3 EKC model

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='CO2 1971-2010')

reg4 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + CPE + legorfr + legorge + legorsc␣
↪+ DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')
print(reg4.summary())

# Finding the EKC income per capita turning point:
# Step 1: Storing the estimates from the regression

reg4.alpha1 = reg4.params['GRGDPPC']
reg4.beta1 = reg4.params['DMMULGDP']

EKC_turning_point = np.exp( - reg4.alpha1 / reg4.beta1 + 8.099206859)
```

```
# Note: The EKC turning point level : From the paper (Note \mu_G = 8.09 =␣
↪ln(6340.99); where 6340.99 is cross-country mean of GDPPC1971 (initial GDP␣
↪per capita 1971))

#print(reg4.alpha1)
#print(reg4.beta1)
print(f"EKC turning point:{EKC_turning_point}")

# White test for heteroskedasticity

reg4_exog = reg4.model.exog

white_test = het_white(reg4.resid, reg4_exog)

print(f"White test for heteroskedasticity: {white_test[0]}")
print(f"P_value (of t-stat): {white_test[1]}")
print(f"F Test stat: {white_test[2]}")
print(f"P-value (of F-test): {white_test[3]}")

if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedsaticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# Breusch-Pagan test for heteroskedasticity

bp_test = het_breuschpagan(reg4.resid, reg4.model.exog)

print(f"BP Test statistics: {bp_test[0]}")
print(f"p_value (Chi-sq distribution): {bp_test[1]}")
print(f"F stat: {bp_test[2]}")
print(f"p_value: {bp_test[3]}")

if bp_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedsaticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to rejet the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# Harvey-Godfrey test for heteroskedasticity (n * R^2)
# Step 1: Calculate the log of squared residuals from reg4
df['log_sq_resid'] = np.log(reg4.resid ** 2)
# Step 2: Regress the log of squared residuals on the independent variables in␣
↪the model
```

```
harvey_reg =  smf.ols(" log_sq_resid ~ GRGDPPC + DMMULGDP + CPE + legorfr +␣
↪legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')
# Step 3: Store the r-sq value from the regression
r_sq_harvey = harvey_reg.rsquared
# calculate the number of obs
n = len(df)
# Calculate the Harvey test statistics: n * R^2
harvey_test_stat = n * r_sq_harvey
# Calculate the chi_sq stat:
p_value = chi2.sf(harvey_test_stat , df = 10) # 10 independent variables in the␣
↪model (in harvey_reg)

print(f"harvey_test (LM stat): {harvey_test_stat}")
print(f"p-value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis of heteroskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of heteroskedasticity. Evidence␣
↪of no heteroskedasticity.")

# LR test stat against the general model:

# Estimating the unrestricted model here again to obtain the number of␣
↪paprameters from the unrestricted model
reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

llf_unrestricted = reg1.llf
llf_restricted = reg4.llf # Here llf stands for log-likelihood function.

lr_stat = 2 * (llf_unrestricted - llf_restricted)

df_diff = reg1.df_model - reg4.df_model

p_value_lr = chi2.sf(lr_stat, df_diff)


print(f"LR statistics: {lr_stat}")
print(f"p_value (chi-sq): {p_value_lr}")

if p_value_lr < 0.05:
```

```
    print("Reject the null hypothesis of restricted model provides a good fit.␣
 ↪The unrestricted model fits the model better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
 ↪good fit. No significant improvement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.478
Model:                            OLS   Adj. R-squared:                  0.435
Method:                 Least Squares   F-statistic:                     10.77
Date:                Thu, 27 Jun 2024   Prob (F-statistic):           5.73e-13
Time:                        11:30:27   Log-Likelihood:                 346.81
No. Observations:                 134   AIC:                            -671.6
Df Residuals:                     123   BIC:                            -639.7
Df Model:                          10
Covariance Type:                  HC0
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       -0.0011      0.004     -0.324      0.746      -0.008       0.006
GRGDPPC          0.7964      0.125      6.386      0.000       0.552       1.041
DMMULGDP        -0.2648      0.091     -2.921      0.003      -0.442      -0.087
CPE             -0.0167      0.009     -1.863      0.062      -0.034       0.001
legorfr          0.0036      0.004      0.958      0.338      -0.004       0.011
legorge         -0.0057      0.007     -0.824      0.410      -0.019       0.008
legorsc         -0.0072      0.008     -0.885      0.376      -0.023       0.009
DMSUMT           0.0003      0.000      0.689      0.491      -0.001       0.001
DMWINT       -8.524e-05      0.000     -0.340      0.734      -0.001       0.000
DMLNFFPC1971    -0.0005      0.000     -1.103      0.270      -0.001       0.000
DMLNPOPD1971     0.0001      0.001      0.113      0.910      -0.002       0.002
==============================================================================
Omnibus:                       15.185   Durbin-Watson:                   2.044
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               42.703
Skew:                           0.293   Prob(JB):                     5.34e-10
Kurtosis:                       5.703   Cond. No.                         878.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
EKC turning point:66626.62091065194
White test for heteroskedasticity: 62.623501547779185
P_value (of t-stat): 0.17172887541964668
F Test stat: 1.3243300262956288
P-value (of F-test): 0.12649741419268612
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
BP Test statistics: 15.008299877096468
```

```
p_value (Chi-sq distribution): 0.13175955060780753
F stat: 1.551386258853481
p_value: 0.12926593356847557
Fail to rejet the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
harvey_test (LM stat): 20.846861140784867
p-value: 0.022187086598483008
Reject the null hypothesis of heteroskedasticity. Evidence of
heteroskedasticity.
LR statistics: 86.85371224185042
p_value (chi-sq): 1.3802444842278738e-19
Reject the null hypothesis of restricted model provides a good fit. The
unrestricted model fits the model better.
```

1.5 Table 3 : Restricted model: CO2 emission 1971-2010 ~ IPAT Model (Column 5)

By applying restrictions to the general model shown above, the IPAT model is estimated. Equation (6) shows the regression equation:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model are: $\beta_1 = \beta_2 = \beta_3 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[4]:
```python
# IPAT (Table 3)

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx',sheet_name='CO2 1971-2010')

reg5 = smf.ols("GREPC ~ GRGDPPC + CPE + legorfr + legorge + legorsc + DMSUMT +␣
↪DMWINT + DMLNFFPC1971 + DMLNPOPD1971",data=df).fit(cov_type='HC0')
print(reg5.summary())
```

```
#  White test for heteroskedasticity:
reg5_exog = reg5.model.exog

white_test = het_white(reg5.resid, reg5_exog)

print(f"White test statistics: {white_test[0]}")
print(f"P_value: {white_test[1]}")
print(f"F stat: {white_test[2]}")
print(f"p_value: {white_test[3]}")

if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# BP test for heteroskedasticity.

bp_test = het_breuschpagan(reg5.resid, reg5_exog)

print(f"bp_test_stat: {bp_test[0]}")
print(f"p_value: {bp_test[1]}")

if bp_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to accept the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# Harvey-test - Has to be manually calculated
# calculate the log of squared residuals and store it in a column
df['log_sq_resid'] = np.log(reg5.resid ** 2)
# Regress the log od squared residuals on the independent variables
harvey_reg = smf.ols("log_sq_resid ~ GRGDPPC + CPE + legorfr + legorge +␣
↪legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971",data=df).
↪fit(cov_type='HC0')
# Obtain the R^2 from the harvey regression
rsquared_harvey = harvey_reg.rsquared
# Obtain the no of obs. of the data
n=len(df)
# Harvey test stat
harvey_stat = n * rsquared_harvey
# Calculate the p_value of the harvey test - it follows a chi-sq distribution
```

```
p_value_harvey = chi2.sf(harvey_stat, df = 9) # Degree of freedom is the no of␣
↪independent variables in the harvey regression.

print(f"Harvey test: {harvey_stat}")
print(f"p_value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# LR test : Comparing the restricted model with the unrestricted model
# Unrestricted model (General Model)
reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

# collect the llf stat from both the models
llf_unrestricted = reg1.llf
llf_restricted = reg5.llf

# Calculate the lr stat:
lr_stat= 2 * (llf_unrestricted - llf_restricted)

# Calculate the p_value:
df_diff = llf_unrestricted - llf_restricted

p_value_lr = chi2.sf(lr_stat, df_diff)

print(f"LR statistics: {lr_stat}")
print(f"p_value: {p_value_lr}")

if p_value_lr < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit:␣
↪The unrestricted model fits significantly better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
↪good fit: No significan timprovement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.431
Model:                            OLS   Adj. R-squared:                  0.389
Method:                 Least Squares   F-statistic:                     12.37
Date:                Fri, 21 Jun 2024   Prob (F-statistic):           7.94e-14
```

```
Time:                        21:25:25   Log-Likelihood:                 341.04
No. Observations:                 134   AIC:                            -662.1
Df Residuals:                     124   BIC:                            -633.1
Df Model:                           9
Covariance Type:                  HC0
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        -0.0053      0.003     -1.648      0.099      -0.012       0.001
GRGDPPC           0.9890      0.118      8.377      0.000       0.758       1.220
CPE              -0.0114      0.008     -1.410      0.158      -0.027       0.004
legorfr           0.0049      0.004      1.250      0.211      -0.003       0.013
legorge          -0.0034      0.007     -0.453      0.651      -0.018       0.011
legorsc          -0.0088      0.009     -0.933      0.351      -0.027       0.010
DMSUMT            0.0005      0.000      1.355      0.176      -0.000       0.001
DMWINT            0.0001      0.000      0.396      0.692      -0.000       0.001
DMLNFFPC1971     -0.0004      0.000     -0.874      0.382      -0.001       0.000
DMLNPOPD1971     -0.0012      0.001     -0.869      0.385      -0.004       0.002
==============================================================================
Omnibus:                       16.819   Durbin-Watson:                   2.023
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               51.697
Skew:                           0.319   Prob(JB):                     5.94e-12
Kurtosis:                       5.975   Cond. No.                         719.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
White test statistics: 59.56368515878082
P_value: 0.058762025963511044
F stat: 1.6185794620785032
p_value: 0.02790738794517342
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
bp_test_stat: 15.885667741819455
p_value: 0.06930787287573381
Fail to accept the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
Harvey test: 14.714180180131681
p_value: 0.09909212354157193
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
LR statistics: 98.3834528217285
p_value: 3.9388383719571115e-05
Reject the null hypothesis of restricted model provides a good fit: The
unrestricted model fits significantly better.
```

1.6 Table 5 : General model period 1: CO2 emission 1971-1990

In this regression, the authors have estimated the general model for CO2 emission as in equation (2) in the paper but only for a truncated period 1971-1990.

Rewriting equation (2) for convenience:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

where hats indicate long-run growth rates, \$ E\$ is the log of emissions per-capita, and G is the log of GDP per capita. X_i is a vector of explanatory and control variables. The sample mean has been deducted from each continuous level variable in this vector.

[3]:

```
# TABLE 5 CO2 - 1971-1990

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx',sheet_name='CO2 1971-1990')

reg6 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')
print(reg6.summary())

# Turning point -  EKC
alpha1 = reg6.params['GRGDPPC']
beta1 = reg6.params['DMMULGDP']
EKC_turning_point = np.exp( - alpha1 / beta1 + 8.099206859)
print(f"EKC Turning point:{EKC_turning_point}")

# White_test
reg6_exog = reg6.model.exog

white_test=het_white(reg6.resid, reg6_exog)

print(f"White test stat: {white_test[0]} ")
print(f"P_value: {white_test[1]}")
```

```
if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.")

# Breusch-Pagan test
bp_test = het_breuschpagan(reg6.resid, reg6_exog)

print(f"BP test stat: {bp_test[0]}")
print(f"p_value: {bp_test[1]}")

if bp_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.")

# Harvey Test:
# Calculate the log of squared residuals from the original regression.
df['log_sq_resid'] = np.log(reg6.resid ** 2)
# Run the harvey regression
harvey_test = smf.ols("log_sq_resid ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 +␣
 ↪DMLNEPC1971 + CPE + legorfr + legorge + legorsc + DMSUMT + DMWINT +␣
 ↪DMLNFFPC1971 + DMLNPOPD1971", data=df).fit(cov_type='HC0')

rsquared = harvey_test.rsquared

n=len(df)

harvey_stat = n * rsquared # Harvey test: n * R^2 from harvey test

p_value_harvey = chi2.sf(harvey_stat, df=12) # 12 nos of independent variables

print(f"Harvey-Godfrey test for heteroskedasticity: {harvey_stat}")
print(f"p_value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪heteroskedasticity.")
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.')
```

```
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.586
Model:                            OLS   Adj. R-squared:                  0.545
Method:                 Least Squares   F-statistic:                     16.09
Date:                Thu, 27 Jun 2024   Prob (F-statistic):           6.90e-20
Time:                        10:48:24   Log-Likelihood:                 311.99
No. Observations:                 134   AIC:                            -598.0
Df Residuals:                     121   BIC:                            -560.3
Df Model:                          12
Covariance Type:                  HC0
================================================================================
=
                  coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-
Intercept       -0.0014      0.004     -0.339      0.734      -0.010
0.007
GRGDPPC          0.8177      0.099      8.259      0.000       0.624
1.012
DMMULGDP        -0.0448      0.065     -0.688      0.492      -0.173
0.083
DMLNGDPPC1971    0.0245      0.006      4.408      0.000       0.014
0.035
DMLNEPC1971     -0.0234      0.004     -6.589      0.000      -0.030
-0.016
CPE             -0.0093      0.012     -0.796      0.426      -0.032
0.014
legorfr          0.0049      0.005      1.051      0.293      -0.004
0.014
legorge          0.0075      0.010      0.715      0.475      -0.013
0.028
legorsc         -0.0101      0.009     -1.109      0.268      -0.028
0.008
DMSUMT           0.0017      0.001      2.504      0.012       0.000
0.003
DMWINT          -0.0010      0.000     -3.060      0.002      -0.002
-0.000
DMLNFFPC1971     0.0020      0.001      2.838      0.005       0.001
0.003
DMLNPOPD1971     0.0017      0.002      0.954      0.340      -0.002
0.005
==============================================================================
Omnibus:                        2.242   Durbin-Watson:                   2.295
Prob(Omnibus):                  0.326   Jarque-Bera (JB):                2.028
Skew:                           0.003   Prob(JB):                        0.363
Kurtosis:                       3.603   Cond. No.                         550.
==============================================================================
```

```
Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
EKC Turning point:274175751573.85257
White test stat: 106.15309225202549
P_value: 0.004365902246960979
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
BP test stat: 23.802254612861898
p_value: 0.02163769164636923
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Harvey-Godfrey test for heteroskedasticity: 11.358018236214125
p_value: 0.4985110168932617
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
```

1.7 Table 6 : General model period 2: CO2 emission 1990-2010

In this regression, the authors have estimated the general model for CO2 emission as in equation (2) in the paper but only for a truncated period 1990-2010.

Rewriting equation (2) for convenience:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

where hats indicate long-run growth rates, \$ E\$ is the log of emissions per-capita, and G is the log of GDP per capita. X_i is a vector of explanatory and control variables. The sample mean has been deducted from each continuous level variable in this vector.

[1]:
```
# TABLE 6 CO2 - 1990-2010

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx',sheet_name='CO2 1990-2010')

# print(df.head())
# Note in the regression the change of initial date to 1990 from 1971. But the␣
↪log of fossial fuel consumption is calculated in 1971 not in 1990.
```

```
reg7 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1990 + DMLNEPC1990 + CPE␣
↪+ LEGORFR + LEGORGE + LEGORSC + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1990 ", data=df).fit(cov_type='HC0')
print(reg7.summary())

# EKC turning point:
alpha1 = reg7.params['GRGDPPC']
beta1 = reg7.params['DMMULGDP']
EKC_turning_point = np.exp(- alpha1 / beta1 + 8.341672) # Here \mu_G is just␣
↪the average of the column LNGDPPC1990
print(f"EKC turning point: {EKC_turning_point}")

# White test for heteroskedasticity
white_test = het_white(reg7.resid, reg7.model.exog)

print(f"White test for heteroskedasticity : {white_test[0]}")
print(f"p-value: {white_test[1]}")
if white_test[1] < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.')

# Breusch-Pagan test for heteroskedasticity
bp_test = het_breuschpagan(reg7.resid, reg7.model.exog)

print(f"Breusch-Pagan Test: {bp_test[0]}")
print(f"p_value: {bp_test[1]}")

if bp_test[1] < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.')

# Harvey-Godfrey test:
# Calulate the log of squared residuals from the original regression: reg7
df['log_sq_resid'] = np.log(reg7.resid ** 2)
#print(df.head())
# Regress the log of squared residuals with the independent variables of the␣
↪model (Auxiliary regression)
harvey_reg = smf.ols("log_sq_resid ~ GRGDPPC + DMMULGDP + DMLNGDPPC1990 +␣
↪DMLNEPC1990 + CPE + LEGORFR + LEGORGE + LEGORSC + DMSUMT + DMWINT +␣
↪DMLNFFPC1971 + DMLNPOPD1990 ",data=df).fit(cov_type = 'HC0')
```

```python
# Store the r-square value from the harvey_reg
harvey_rsquared = harvey_reg.rsquared
# Calculate the number of independent variable in the model
n = len(df)
# Find the Harvey test stat and the p-value (follows a chi-sq distribution)
harvey_test = n * harvey_rsquared
p_value_harvey = chi2.sf(harvey_test, df=12 ) # the degree of freedom is equal␣
↪to the number of independent variables in the model.

print(f"Harvey Test Stat: {harvey_test}")
print(f"p value : {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.580
Model:                            OLS   Adj. R-squared:                  0.538
Method:                 Least Squares   F-statistic:                     38.10
Date:                Wed, 26 Jun 2024   Prob (F-statistic):           2.17e-35
Time:                        10:16:21   Log-Likelihood:                 335.95
No. Observations:                 134   AIC:                            -645.9
Df Residuals:                     121   BIC:                            -608.2
Df Model:                          12
Covariance Type:                  HC0
================================================================================
=
                   coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-
Intercept       -0.0036      0.003     -1.191      0.234      -0.009
0.002
GRGDPPC          0.9279      0.136      6.848      0.000       0.662
1.193
DMMULGDP        -0.1095      0.060     -1.836      0.066      -0.227
0.007
DMLNGDPPC1990    0.0137      0.004      3.338      0.001       0.006
0.022
DMLNEPC1990     -0.0133      0.003     -3.814      0.000      -0.020
-0.006
CPE             -0.0043      0.010     -0.455      0.649      -0.023
```

```
0.014
LEGORFR           -0.0027      0.004     -0.754      0.451      -0.010
0.004
LEGORGE           -0.0031      0.007     -0.473      0.636      -0.016
0.010
LEGORSC            0.0007      0.006      0.125      0.900      -0.011
0.012
DMSUMT             0.0009      0.001      1.591      0.112      -0.000
0.002
DMWINT         -6.769e-05      0.000     -0.188      0.851      -0.001
0.001
DMLNFFPC1971       0.0010      0.001      1.953      0.051   -3.82e-06
0.002
DMLNPOPD1990      -0.0013      0.002     -0.765      0.444      -0.005
0.002
==============================================================================
Omnibus:                       36.492   Durbin-Watson:                   2.010
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               91.734
Skew:                          -1.065   Prob(JB):                     1.20e-20
Kurtosis:                       6.448   Cond. No.                         891.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
EKC turning point: 20005157.212090995
White test for heteroskedasticity : 107.9285530174746
p-value: 0.0031082454974498777
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Breusch-Pagan Test: 17.140654481707116
p_value: 0.14438252588514103
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
Harvey Test Stat: 26.704368966986404
p value : 0.008520708310142702
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
```

1.8 Table 6 : General model period 2: SO2 emission 1988-2005

In this regression, the authors have estimated the general model for SO2 emission as in equation (2) in the paper but only for a truncated period 1988-2005.

Rewriting equation (2) for convenience:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

where hats indicate long-run growth rates, $ E$ is the log of emissions per-capita, and G is the log of GDP per capita. X_i is a vector of explanatory and control variables. The sample mean has been deducted from each continuous level variable in this vector.

[14]:

```python
# Table 6 - SO2 1988-2005

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx',sheet_name='SO2 1988-2005')

# print(df.head())
# Note in the regression the change of initial date to 1990 from 1971. But the␣
↪log of fossial fuel consumption is calculated in 1971 not in 1990.
reg7 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1988 + DMLNEPC1988 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1988 ", data=df).fit(cov_type='HC0')
print(reg7.summary())

# EKC turning point:
alpha1 = reg7.params['GRGDPPC']
beta1 = reg7.params['DMMULGDP']
EKC_turning_point = np.exp(- alpha1 / beta1 + 8.5637) # Here \mu_G is just the␣
↪average of the column LNGDPPC1988
print(f"EKC turning point: {EKC_turning_point}")

# White test for heteroskedasticity
white_test = het_white(reg7.resid, reg7.model.exog)

print(f"White test for heteroskedasticity : {white_test[0]}")
print(f"p-value: {white_test[1]}")
if white_test[1] < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.')

# Breusch-Pagan test for heteroskedasticity
bp_test = het_breuschpagan(reg7.resid, reg7.model.exog)

print(f"Breusch-Pagan Test: {bp_test[0]}")
print(f"p_value: {bp_test[1]}")
```

```python
if bp_test[1] < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.')

# Harvey-Godfrey test:
# Calulate the log of squared residuals from the original regression: reg7
df['log_sq_resid'] = np.log(reg7.resid ** 2)
#print(df.head())
# Regress the log of squared residuals with the independent variables of the␣
↪model (Auxiliary regression)
harvey_reg = smf.ols("log_sq_resid ~ GRGDPPC + DMMULGDP + DMLNGDPPC1988 +␣
↪DMLNEPC1988 + CPE + legorfr + legorge + legorsc + DMSUMT + DMWINT +␣
↪DMLNFFPC1971 + DMLNPOPD1988 ",data=df).fit(cov_type = 'HC0')
# Store the r-square value from the harvey_reg
harvey_rsquared = harvey_reg.rsquared
# Calculate the number of independent variable in the model
n = len(df)
# Find the Harvey test stat and the p-value (follows a chi-sq distribution)
harvey_test = n * harvey_rsquared
p_value_harvey = chi2.sf(harvey_test, df=12 ) # the degree of freedom is equal␣
↪to the number of independent variables in the model.

print(f"Harvey Test Stat: {harvey_test}")
print(f"p value : {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.624
Model:                            OLS   Adj. R-squared:                  0.562
Method:                 Least Squares   F-statistic:                     12.29
Date:                Wed, 26 Jun 2024   Prob (F-statistic):           3.41e-13
Time:                        11:05:47   Log-Likelihood:                 173.99
No. Observations:                  86   AIC:                            -322.0
Df Residuals:                      73   BIC:                            -290.1
Df Model:                          12
Covariance Type:                  HC0
```

```
==============================================================================
=
                    coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-
Intercept        -0.0223      0.008     -2.893      0.004      -0.037
-0.007
GRGDPPC           0.7678      0.190      4.041      0.000       0.395
1.140
DMMULGDP         -0.5485      0.189     -2.903      0.004      -0.919
-0.178
DMLNGDPPC1988     0.0210      0.005      3.891      0.000       0.010
0.032
DMLNEPC1988      -0.0135      0.004     -3.250      0.001      -0.022
-0.005
CPE              -0.0062      0.023     -0.272      0.786      -0.051
0.039
legorfr          -0.0099      0.008     -1.261      0.207      -0.025
0.005
legorge          -0.0264      0.017     -1.583      0.113      -0.059
0.006
legorsc          -0.0733      0.017     -4.306      0.000      -0.107
-0.040
DMSUMT            0.0045      0.001      4.473      0.000       0.003
0.007
DMWINT           -0.0007      0.001     -1.171      0.242      -0.002
0.000
DMLNFFPC1971     -0.0007      0.001     -0.513      0.608      -0.003
0.002
DMLNPOPD1988     -0.0094      0.003     -3.233      0.001      -0.015
-0.004
==============================================================================
Omnibus:                        0.373   Durbin-Watson:                   1.817
Prob(Omnibus):                  0.830   Jarque-Bera (JB):                0.268
Skew:                          -0.136   Prob(JB):                        0.875
Kurtosis:                       2.965   Cond. No.                         917.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
EKC turning point: 21238.08513049237
White test for heteroskedasticity : 76.44044721348092
p-value: 0.2258895650380592
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
Breusch-Pagan Test: 9.977120443058821
p_value: 0.6179679539055802
```

```
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
Harvey Test Stat: 10.424952369779191
p value : 0.578732888079803
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
```

1.9 Table 5 : General model period 1: SO2 emission 1971-1988

In this regression, the authors have estimated the general model for SO2 emission as in equation (2) in the paper but only for a truncated period 1971-1988.

Rewriting equation (2) for convenience:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

where hats indicate long-run growth rates. E is the log of emissions per-capita, and G is the log of GDP per capita. X_i is a vector of explanatory and control variables. The sample mean has been deducted from each countinuous levels variables in this vector.

[15]:
```python
# Table 5 - SO2 1971-1988

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx',sheet_name='SO2 1971-1988')

# print(df.head())
# Note in the regression the change of initial date to 1990 from 1971. But the␣
↪log of fossial fuel consumption is calculated in 1971 not in 1990.
reg7 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971 ", data=df).fit(cov_type='HC0')
print(reg7.summary())

# EKC turning point:
alpha1 = reg7.params['GRGDPPC']
beta1 = reg7.params['DMMULGDP']
```

```python
EKC_turning_point = np.exp(- alpha1 / beta1 + 8.318) # Here \mu_G is just the␣
↪average of the column LNGDPPC1971
print(f"EKC turning point: {EKC_turning_point}")

# White test for heteroskedasticity
white_test = het_white(reg7.resid, reg7.model.exog)

print(f"White test for heteroskedasticity : {white_test[0]}")
print(f"p-value: {white_test[1]}")
if white_test[1] < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.')

# Breusch-Pagan test for heteroskedasticity
bp_test = het_breuschpagan(reg7.resid, reg7.model.exog)

print(f"Breusch-Pagan Test: {bp_test[0]}")
print(f"p_value: {bp_test[1]}")

if bp_test[1] < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
else:
    print('Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.')

# Harvey-Godfrey test:
# Calulate the log of squared residuals from the original regression: reg7
df['log_sq_resid'] = np.log(reg7.resid ** 2)
#print(df.head())
# Regress the log of squared residuals with the independent variables of the␣
↪model (Auxiliary regression)
harvey_reg = smf.ols("log_sq_resid ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 +␣
↪DMLNEPC1971 + CPE + legorfr + legorge + legorsc + DMSUMT + DMWINT +␣
↪DMLNFFPC1971 + DMLNPOPD1971 ",data=df).fit(cov_type = 'HC0')
# Store the r-square value from the harvey_reg
harvey_rsquared = harvey_reg.rsquared
# Calculate the number of independent variable in the model
n = len(df)
# Find the Harvey test stat and the p-value (follows a chi-sq distribution)
harvey_test = n * harvey_rsquared
p_value_harvey = chi2.sf(harvey_test, df=12 ) # the degree of freedom is equal␣
↪to the number of independent variables in the model.
```

```python
print(f"Harvey Test Stat: {harvey_test}")
print(f"p value : {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.635
Model:                            OLS   Adj. R-squared:                  0.585
Method:                 Least Squares   F-statistic:                     8.892
Date:                Wed, 26 Jun 2024   Prob (F-statistic):           7.62e-11
Time:                        11:11:58   Log-Likelihood:                 172.87
No. Observations:                 100   AIC:                            -319.7
Df Residuals:                      87   BIC:                            -285.9
Df Model:                          12
Covariance Type:                  HC0
================================================================================
=
                    coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-
Intercept         0.0076      0.009      0.812      0.417      -0.011
0.026
GRGDPPC           0.8340      0.191      4.373      0.000       0.460
1.208
DMMULGDP         -0.1498      0.141     -1.060      0.289      -0.427
0.127
DMLNGDPPC1971     0.0246      0.012      2.101      0.036       0.002
0.048
DMLNEPC1971      -0.0304      0.009     -3.248      0.001      -0.049
-0.012
CPE               0.0728      0.024      2.985      0.003       0.025
0.121
legorfr          -0.0241      0.009     -2.602      0.009      -0.042
-0.006
legorge          -0.0573      0.022     -2.647      0.008      -0.100
-0.015
legorsc          -0.0714      0.023     -3.139      0.002      -0.116
-0.027
DMSUMT            0.0031      0.001      2.495      0.013       0.001
0.006
```

```
DMWINT            -0.0022      0.001      -3.667      0.000      -0.003
-0.001
DMLNFFPC1971      -0.0017      0.001      -1.444      0.149      -0.004
0.001
DMLNPOPD1971      -0.0120      0.004      -2.942      0.003      -0.020
-0.004
==============================================================================
Omnibus:                       48.133   Durbin-Watson:                   1.853
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              203.768
Skew:                           1.523   Prob(JB):                     5.65e-45
Kurtosis:                       9.295   Cond. No.                         547.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
EKC turning point: 1072271.276296573
White test for heteroskedasticity : 99.16947725009595
p-value: 0.010104134059967036
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Breusch-Pagan Test: 26.56389227610282
p_value: 0.008924297940185963
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Harvey Test Stat: 12.60396061309577
p value : 0.39847104634769565
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
```

1.10 Table 4 : Restricted model: SO2 emission 1971-2005 ~ Ordas Criado et al. (2011) (Column 2)

By applying restrictions to the general model shown above, Ordas Criado et al. (2011) is estimated. Equation (3) depicts the regression equation:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model is: $\beta_1 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[16]:
```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

```python
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='SO2 1971-2005')

reg2= smf.ols("GREPC ~ GRGDPPC + DMLNGDPPC1971 + DMLNEPC1971 + CPE + legorfr +␣
↪legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')
print(reg2.summary())

# For conducting different statistical tests we have to follow the following␣
↪steps:
# 1. Create a matrix of exogenous variables

exog_het = reg2.model.exog

# 2. Run the white test: H0: Homoskedasticity

white_test = het_white(reg2.resid, exog_het)

white_pvalue = white_test[1]
# 3. Print the results

print(f"White test for heteroskedasticity:")
print(f"T-stats for white test: {white_test[0]}")
print(f"P-value: {white_test[1]}")
print(f" F statistics: {white_test[2]}")
print(f" p_value: {white_test[3]}")

if white_pvalue < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. No evidence␣
↪of heteroskedasticity")

# 3. Run the Breusch-Pagan test: H0: Homoskedasticity

# The het_breuschpagan function returns a tuple with four elements:
# a) Lagrange Multiplier (LM) Statistic: The first element, accessible with␣
↪bp_test[0], is the test statistic for the Breusch-Pagan test.
# b) p-value: The second element, accessible with bp_test[1], is the p-value␣
↪associated with the LM statistic.
```

```
# c) f-value: The third element, accessible with bp_test[2], is the f-value of␣
↪the test.
# d) f p-value: The fourth element, accessible with bp_test[3], is the p-value␣
↪associated with the f-statistic.

bp_test = het_breuschpagan(reg2.resid, exog_het)

bp_pvalue= bp_test[1]

print(f"BP test:")
print(f"LM Statistic (Chi-square): {bp_test[0]}")
print(f"LM Test p-value: {bp_test[1]}")
print(f" F statistics: {bp_test[2]}")
print(f" p_value: {bp_test[3]}")

if bp_pvalue < 0.05:
      print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
      print("Fail to reject the null hypothesis of homoskedasticity. Evidence␣
↪of no heteroskedasticity.")

# 4. Run the Harvey test (Harvey test doesn't exist in any statistical packages␣
↪in python. Thus it has to do it manually)

    # Step 1: Calculate the log of the squared residuals

df['log_sq_residuals'] = np.log(reg2.resid**2)

    # Step 2: Regress log_squared_residuals on the independent variables

harvey_model = smf.ols("log_sq_residuals ~ GRGDPPC + DMLNGDPPC1971 +␣
↪DMLNEPC1971 + CPE + legorfr + legorge + legorsc + DMSUMT + DMWINT +␣
↪DMLNFFPC1971 + DMLNPOPD1971", data=df).fit()

    # Step 3: Obtain the R-squared value from this regression

r_sq = harvey_model.rsquared

    # Step 4: Calculate the LM statistic

n = len(df) # this command helps in identifying the number of obs in the␣
↪dataset. len() ~ determines the length of various objects.
lm_stat_harvey = n * r_sq # The LM statistics is calulated as N * r_squared

    # Step 5: Calculate the p-value
```

```
p_value_harvey = chi2.sf(lm_stat_harvey, df=11) # degree of freedom (df) = 11␣
↪for eleven independent variables

    # Step 6: Print the results

print("Harvey-Godfrey test for heteroskedasticity:") # H0: Homoskedasticity
print(f" LM Statistics: {lm_stat_harvey}")
print(f"LM test p-value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no hoteroskedasticity.")


# 5. Performing the Likelihood test (LR test) against the general model.␣
↪(follows a chi-sq distribution)
# The Likelihood Ratio (LR) test is used to compare the goodness-of-fit of two␣
↪nested models. The H0: The restricted model provides a good fit.

reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

    # Calculating the log-likelihood values of both the models

llf_restricted = reg2.llf
llf_unrestricted = reg1.llf

    # Calculating the LR statistics
lr_statistic = -2 * (llf_restricted - llf_unrestricted)

    # Degrees of freedom (df): number of restrictions (difference in number of␣
↪parameters). In this case it should be 1
df_diff = reg1.df_model - reg2.df_model # Note: reg1 is the unrestricted model␣
↪and reg2 is a restricted model

    # Calculating the p_value
p_value = chi2.sf(lr_statistic, df_diff)

print("LR test against the general model:")
print(f"LR Statistic: {lr_statistic}")
print(f"p-value: {p_value}")
```

```python
if p_value < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit:␣
 ↪The unrestricted model fits significantly better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
 ↪good fit: No significant improvement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.735
Model:                            OLS   Adj. R-squared:                  0.702
Method:                 Least Squares   F-statistic:                     17.98
Date:                Wed, 26 Jun 2024   Prob (F-statistic):           3.83e-18
Time:                        11:31:24   Log-Likelihood:                 225.08
No. Observations:                 100   AIC:                            -426.2
Df Residuals:                      88   BIC:                            -394.9
Df Model:                          11
Covariance Type:                  HC0
================================================================================
=
                   coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-
Intercept       -0.0129      0.006     -2.212      0.027      -0.024
-0.001
GRGDPPC          0.9723      0.161      6.056      0.000       0.658
1.287
DMLNGDPPC1971    0.0179      0.005      3.482      0.000       0.008
0.028
DMLNEPC1971     -0.0220      0.004     -5.988      0.000      -0.029
-0.015
CPE              0.0336      0.013      2.559      0.010       0.008
0.059
legorfr         -0.0141      0.006     -2.386      0.017      -0.026
-0.003
legorge         -0.0329      0.010     -3.214      0.001      -0.053
-0.013
legorsc         -0.0431      0.016     -2.712      0.007      -0.074
-0.012
DMSUMT           0.0033      0.001      4.582      0.000       0.002
0.005
DMWINT          -0.0011      0.000     -2.892      0.004      -0.002
-0.000
DMLNFFPC1971    -0.0010      0.001     -1.169      0.242      -0.003
0.001
DMLNPOPD1971    -0.0097      0.002     -4.368      0.000      -0.014
```

```
-0.005
==============================================================================
Omnibus:                        5.778   Durbin-Watson:                   1.728
Prob(Omnibus):                  0.056   Jarque-Bera (JB):                5.166
Skew:                           0.524   Prob(JB):                       0.0756
Kurtosis:                       3.376   Cond. No.                         774.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
White test for heteroskedasticity:
T-stats for white test: 80.39682298732005
P-value: 0.04055479364221336
 F statistics: 2.6657890661271875
 p_value: 0.0007695731167671187
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
BP test:
LM Statistic (Chi-square): 19.756826591929567
LM Test p-value: 0.048789231219805405
 F statistics: 1.9696954397810522
 p_value: 0.0410117574185589
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Harvey-Godfrey test for heteroskedasticity:
 LM Statistics: 16.358572378400943
LM test p-value: 0.1283323721092265
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
hoteroskedasticity.
LR test against the general model:
LR Statistic: 3.3711394882227523
p-value: 0.06634785046250796
Fail to reject the null hypothesis of restricted model provides a good fit: No
significant improvement in fit for the unrestricted model.
```

1.11 Table 4 : Restricted model: SO2 emission 1971-2005 ~ Green Solow Model (Column 3)

By applying restrictions to the general model shown above, Green Solow model (GSM) is estimated. Equation (4) shows the regression equation:

$$\hat{E}_i = \alpha_0 + \beta_2 E_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model are: $\alpha_1 = \beta_1 = \beta_3 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[17]:

```python
# Table 4 SO2 1971-2005 Regression Results

# Green Solow model

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='SO2 1971-2005')

reg3= smf.ols("GREPC ~ DMLNEPC1971 + CPE + legorfr + legorge + legorsc + DMSUMT␣
 ↪+ DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).fit(cov_type = 'HC0')
print(reg3.summary())

# White test for heteroskedasticity
 # Step 1: Create a matrix of exogenous variables
exog_het = reg3.model.exog

 # Follow the above steps for the White test.
white_test = het_white(reg3.resid, exog_het)

print(f"White test for heteroskedasticity: {white_test[0]}")
print(f"p-value: {white_test[1]}")
print(f" F stat: {white_test[2]}")
print(f"p-value : {white_test[3]}")

if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.")

# Breusch-Pagan Test for heteroskedasticity
bp_test = het_breuschpagan(reg3.resid,exog_het)

print(f"Breusch-Pagan test - LM (chi-sq) stat: {bp_test[0]}")
print(f"p-value: {bp_test[1]}")
print(f"F stat: {bp_test[2]}")
print(f"p-value of F stat: {bp_test[3]}")
```

```python
if bp_test[1] < 0.05:
    print("Rejecting the null of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null of homoskedasticity. Evidence of no␣
↪heteroskedasticity.")

# Harvey-Godfrey test. Note the python packages donot have the harvey-test␣
↪module, so have to do it manually.
# Step 1: Calculate the log of squared residuals

df['log_sq_resid'] = np.log(reg3.resid ** 2)

# Step 2: Regress the log of residuals on the independent variables

harvey_reg = smf.ols("log_sq_resid ~ DMLNEPC1971 + CPE + legorfr + legorge +␣
↪legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')

# Step 3: Calculate the r-sq value from the regression

r_sq = harvey_reg.rsquared

# Step 4: Calculate the LM statistics:

n=len(df)   # Explain the number of obs

lm_stat = n * r_sq  # the harvey test statistic is calculated as N * R^2


# Step 5: Calculate the p_value

p_value_harvey = chi2.sf(lm_stat,df=9)

print("Harvey test:")
print(f"LM (chi-sq) test statistics: {lm_stat}")
print(f"p_value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")
```

```
# LR statistics: It is used to find the goodness of fit of restricted model␣
↪against the unrestricted model.
# The unrestricted model:

reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

# Calculate the log likelihood ratio of both the models

llf_unrestricted = reg1.llf
llf_restricted = reg3.llf

# Caculating the LR Statistics

lr_stat = - 2 * (llf_restricted - llf_unrestricted)

# Degrees of freedom (difference in the number of parameters between the␣
↪restricted and unrestricted model)

df_diff = reg1.df_model - reg3.df_model

# LR stat follows a chi-sq distribution

p_value = chi2.sf(lr_stat, df_diff)

print (f"LR test statistics against the general model: {lr_stat}")
print(f"p_value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit:␣
↪The unrestricted model fits significantly better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
↪good fit: No significan timprovement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.605
Model:                            OLS   Adj. R-squared:                  0.565
Method:                 Least Squares   F-statistic:                     10.22
Date:                Wed, 26 Jun 2024   Prob (F-statistic):           1.07e-10
Time:                        11:38:07   Log-Likelihood:                 204.97
No. Observations:                 100   AIC:                            -389.9
Df Residuals:                      90   BIC:                            -363.9
Df Model:                           9
```

```
Covariance Type:                  HC0
================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept         0.0082      0.006      1.381      0.167      -0.003       0.020
DMLNEPC1971      -0.0176      0.003     -6.188      0.000      -0.023      -0.012
CPE               0.0219      0.010      2.236      0.025       0.003       0.041
legorfr          -0.0213      0.007     -2.982      0.003      -0.035      -0.007
legorge          -0.0342      0.010     -3.329      0.001      -0.054      -0.014
legorsc          -0.0341      0.021     -1.645      0.100      -0.075       0.007
DMSUMT            0.0035      0.001      4.711      0.000       0.002       0.005
DMWINT           -0.0019      0.000     -4.098      0.000      -0.003      -0.001
DMLNFFPC1971 -3.701e-05      0.001     -0.041      0.968      -0.002       0.002
DMLNPOPD1971     -0.0050      0.003     -1.947      0.051      -0.010    3.26e-05
==============================================================================
Omnibus:                        6.475   Durbin-Watson:                   1.966
Prob(Omnibus):                  0.039   Jarque-Bera (JB):                7.419
Skew:                           0.353   Prob(JB):                       0.0245
Kurtosis:                       4.132   Cond. No.                         74.6
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
White test for heteroskedasticity: 69.74169865974989
p-value: 0.0060800548846500395
 F stat: 3.001701821699393
p-value : 6.566761639620607e-05
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Breusch-Pagan test - LM (chi-sq) stat: 22.711000979026164
p-value: 0.006879045105691347
F stat: 2.9384519487518665
p-value of F stat: 0.004208768486539984
Rejecting the null of homoskedasticity. Evidence of heteroskedasticity.
Harvey test:
LM (chi-sq) test statistics: 5.700805440724463
p_value: 0.7694491335840247
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
LR test statistics against the general model: 43.59137824180459
p_value: 1.843034714071239e-09
Reject the null hypothesis of restricted model provides a good fit: The
unrestricted model fits significantly better.
```

1.12 Table 4 : Restricted model: SO2 emission 1971-2005 ~ Basic EKC Model (Column 4)

By applying restrictions to the general model shown above, the basic EKC model is estimated. Equation (5) shows the regression equation:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model are: $\beta_2 = \beta_3 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[8]:
```
# Table 4 SO2 1971-2005 Regression Results

# EKC model

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='SO2 1971-2005')

reg4 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + CPE + legorfr + legorge + legorsc␣
↪+ DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
↪fit(cov_type='HC0')
print(reg4.summary())

# Finding the EKC income per capita turning point:
# Step 1: Storing the estimates from the regression

reg4.alpha1 = reg4.params['GRGDPPC']
reg4.beta1 = reg4.params['DMMULGDP']

EKC_turning_point = np.exp( - reg4.alpha1 / reg4.beta1 + 8.3187336) # Log of␣
↪LNGDPPC1971

#print(reg4.alpha1)
#print(reg4.beta1)
print(f"EKC turning point:{EKC_turning_point}")

# White test for heteroskedasticity
```

```
reg4_exog = reg4.model.exog

white_test = het_white(reg4.resid, reg4_exog)

print(f"White test for heteroskedasticity: {white_test[0]}")
print(f"P_value (of t-stat): {white_test[1]}")
print(f"F Test stat: {white_test[2]}")
print(f"P-value (of F-test): {white_test[3]}")

if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedsaticity. Evidence of␣
 ↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.")

# Breusch-Pagan test for heteroskedasticity

bp_test = het_breuschpagan(reg4.resid, reg4.model.exog)

print(f"BP Test statistics: {bp_test[0]}")
print(f"p_value (Chi-sq distribution): {bp_test[1]}")
print(f"F stat: {bp_test[2]}")
print(f"p_value: {bp_test[3]}")

if bp_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedsaticity. Evidence of␣
 ↪heteroskedasticity.")
else:
    print("Fail to rejet the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.")

# Harvey-Godfrey test for heteroskedasticity (n * R^2)
# Step 1: Calculate the log of squared residuals from reg4
df['log_sq_resid'] = np.log(reg4.resid ** 2)
# Step 2: Regress the log of squared residuals on the independent variables in␣
 ↪the model
harvey_reg =  smf.ols(" log_sq_resid ~ GRGDPPC + DMMULGDP + CPE + legorfr +␣
 ↪legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971", data=df).
 ↪fit(cov_type='HC0')
# Step 3: Store the r-sq value from the regression
r_sq_harvey = harvey_reg.rsquared
# calculate the number of obs
n = len(df)
# Calculate the Harvey test statistics: n * R^2
harvey_test_stat = n * r_sq_harvey
```

```python
# Calculate the chi_sq stat:
p_value = chi2.sf(harvey_test_stat , df = 10) # 10 independent variables in the␣
↪model (in harvey_reg)

print(f"harvey_test (LM stat): {harvey_test_stat}")
print(f"p-value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis of heteroskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of heteroskedasticity. Evidence␣
↪of no heteroskedasticity.")

# LR test stat against the general model:

# Estimating the unrestricted model here again to obtain the number of␣
↪paprameters from the unrestricted model
reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

llf_unrestricted = reg1.llf
llf_restricted = reg4.llf # Here llf stands for log-likelihood function.

lr_stat = 2 * (llf_unrestricted - llf_restricted)

df_diff = reg1.df_model - reg4.df_model

p_value_lr = chi2.sf(lr_stat, df_diff)


print(f"LR statistics: {lr_stat}")
print(f"p_value (chi-sq): {p_value_lr}")

if p_value_lr < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit.␣
↪The unrestricted model fits the model better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
↪good fit. No significant improvement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.529
Model:                            OLS   Adj. R-squared:                  0.476
Method:                 Least Squares   F-statistic:                     14.23
```

```
Date:                Thu, 27 Jun 2024   Prob (F-statistic):           1.08e-14
Time:                        11:32:28   Log-Likelihood:                 196.25
No. Observations:                 100   AIC:                            -370.5
Df Residuals:                      89   BIC:                            -341.8
Df Model:                          10
Covariance Type:                  HC0
=================================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept      -0.0030      0.010     -0.307      0.759      -0.022       0.016
GRGDPPC         0.8795      0.221      3.974      0.000       0.446       1.313
DMMULGDP       -0.5578      0.157     -3.564      0.000      -0.865      -0.251
CPE             0.0067      0.013      0.522      0.601      -0.018       0.032
legorfr        -0.0250      0.010     -2.610      0.009      -0.044      -0.006
legorge        -0.0417      0.012     -3.509      0.000      -0.065      -0.018
legorsc        -0.0536      0.021     -2.492      0.013      -0.096      -0.011
DMSUMT          0.0032      0.001      3.151      0.002       0.001       0.005
DMWINT         -0.0007      0.000     -1.348      0.178      -0.002       0.000
DMLNFFPC1971   -0.0028      0.001     -2.396      0.017      -0.005      -0.001
DMLNPOPD1971   -0.0127      0.004     -3.010      0.003      -0.021      -0.004
==============================================================================
Omnibus:                       53.566   Durbin-Watson:                   2.031
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              249.431
Skew:                           1.693   Prob(JB):                     6.87e-55
Kurtosis:                       9.956   Cond. No.                         881.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
EKC turning point:19838.54727680577
White test for heteroskedasticity: 64.09531979328712
P_value (of t-stat): 0.1030160275013332
F Test stat: 1.6801432714889248
P-value (of F-test): 0.03610453324727679
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
BP Test statistics: 16.296637382261036
p_value (Chi-sq distribution): 0.09144986354240889
F stat: 1.732786690595694
p_value: 0.08554107801825975
Fail to rejet the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
harvey_test (LM stat): 11.194721737508557
p-value: 0.3425500693706504
Fail to reject the null hypothesis of heteroskedasticity. Evidence of no
heteroskedasticity.
LR statistics: 61.03891614240899
p_value (chi-sq): 5.566314864214063e-14
```

Reject the null hypothesis of restricted model provides a good fit. The unrestricted model fits the model better.

1.13 Table 4 : Restricted model: SO2 emission 1971-2005 ~ IPAT Model (Column 5)

By applying restrictions to the general model shown above, the IPAT model is estimated. Equation (6) shows the regression equation:

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

The restrictions on the general model are: $\beta_1 = \beta_2 = \beta_3 = 0$

The variables in the above regression equation have the same meaning as mentioned above. This regression has considered all the explanatory variables (X_i), but the results are not specified in the table.

[20]:
```
# Table 4 SO2 1971-2005 Regression Results

# IPAT

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx',sheet_name='SO2 1971-2005')

reg5 = smf.ols("GREPC ~ GRGDPPC + CPE + legorfr + legorge + legorsc + DMSUMT +␣
↪DMWINT + DMLNFFPC1971 + DMLNPOPD1971",data=df).fit(cov_type='HC0')
print(reg5.summary())

#  White test for heteroskedasticity:
reg5_exog = reg5.model.exog

white_test = het_white(reg5.resid, reg5_exog)

print(f"White test statistics: {white_test[0]}")
print(f"P_value: {white_test[1]}")
print(f"F stat: {white_test[2]}")
```

```python
print(f"p_value: {white_test[3]}")

if white_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# BP test for heteroskedasticity.

bp_test = het_breuschpagan(reg5.resid, reg5_exog)

print(f"bp_test_stat: {bp_test[0]}")
print(f"p_value: {bp_test[1]}")

if bp_test[1] < 0.05:
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to accept the null hypothesis of homoskedasticity. Evidence of␣
↪no heteroskedasticity.")

# Harvey-test - Has to be manually calculated
# calculate the log of squared residuals and store it in a column
df['log_sq_resid'] = np.log(reg5.resid ** 2)
# Regress the log od squared residuals on the independent variables
harvey_reg = smf.ols("log_sq_resid ~ GRGDPPC + CPE + legorfr + legorge +␣
↪legorsc + DMSUMT + DMWINT + DMLNFFPC1971 + DMLNPOPD1971",data=df).
↪fit(cov_type='HC0')
# Obtain the R^2 from the harvey regression
rsquared_harvey = harvey_reg.rsquared
# Obtain the no of obs. of the data
n=len(df)
# Harvey test stat
harvey_stat = n * rsquared_harvey
# Calculate the p_value of the harvey test - it follows a chi-sq distribution
p_value_harvey = chi2.sf(harvey_stat, df = 9) # Degree of freedom is the no of␣
↪independent variables in the harvey regression.

print(f"Harvey test: {harvey_stat}")
print(f"p_value: {p_value_harvey}")

if p_value_harvey < 0.05:
    print('Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.')
```

```
else:
    print("Fail to reject the null hypothesis of homoskedasticity. Evidence of␣
 ↪no heteroskedasticity.")

# LR test : Comparing the restricted model with the unrestricted model
# Unrestricted model (General Model)
reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
 ↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
 ↪DMLNPOPD1971", data=df).fit(cov_type='HC0')

# collect the llf stat from both the models
llf_unrestricted = reg1.llf
llf_restricted = reg5.llf

# Calculate the lr stat:
lr_stat= 2 * (llf_unrestricted - llf_restricted)

# Calculate the p_value:
df_diff = llf_unrestricted - llf_restricted

p_value_lr = chi2.sf(lr_stat, df_diff)

print(f"LR statistics: {lr_stat}")
print(f"p_value: {p_value_lr}")

if p_value_lr < 0.05:
    print("Reject the null hypothesis of restricted model provides a good fit:␣
 ↪The unrestricted model fits significantly better.")
else:
    print("Fail to reject the null hypothesis of restricted model provides a␣
 ↪good fit: No significan timprovement in fit for the unrestricted model.")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.479
Model:                            OLS   Adj. R-squared:                  0.426
Method:                 Least Squares   F-statistic:                     10.38
Date:                Wed, 26 Jun 2024   Prob (F-statistic):           7.70e-11
Time:                        12:22:36   Log-Likelihood:                 191.15
No. Observations:                 100   AIC:                            -362.3
Df Residuals:                      90   BIC:                            -336.2
Df Model:                           9
Covariance Type:                  HC0
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       -0.0106      0.010     -1.028      0.304      -0.031       0.010
```

```
GRGDPPC          1.2176      0.262      4.644      0.000       0.704       1.731
CPE              0.0211      0.013      1.634      0.102      -0.004       0.046
legorfr         -0.0231      0.010     -2.319      0.020      -0.043      -0.004
legorge         -0.0361      0.013     -2.685      0.007      -0.062      -0.010
legorsc         -0.0555      0.022     -2.518      0.012      -0.099      -0.012
DMSUMT           0.0045      0.001      4.782      0.000       0.003       0.006
DMWINT          -0.0005      0.001     -0.956      0.339      -0.002       0.001
DMLNFFPC1971    -0.0026      0.001     -2.113      0.035      -0.005      -0.000
DMLNPOPD1971    -0.0153      0.004     -3.556      0.000      -0.024      -0.007
==============================================================================
Omnibus:                       41.410   Durbin-Watson:                   2.036
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              137.759
Skew:                           1.374   Prob(JB):                     1.22e-30
Kurtosis:                       8.050   Cond. No.                         748.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
White test statistics: 61.3182673061752
P_value: 0.03448818771142843
F stat: 2.0644459944892377
p_value: 0.005537988848957214
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
bp_test_stat: 18.950183512066154
p_value: 0.025619821158774553
Reject the null hypothesis of homoskedasticity. Evidence of heteroskedasticity.
Harvey test: 6.63112288718729
p_value: 0.67546299669092
Fail to reject the null hypothesis of homoskedasticity. Evidence of no
heteroskedasticity.
LR statistics: 71.24415196763232
p_value: 0.0003622576264703981
Reject the null hypothesis of restricted model provides a good fit: The
unrestricted model fits significantly better.
```

1.14 Table 2 : General model (SO2 emission)

The regression shown below is of the general model for SO2 emission as in regression equation 2 in the original paper.

$$\hat{E}_i = \alpha_0 + \alpha_1 \hat{G}_i + \beta_1 \hat{G}_i G_{i,0} + \beta_2 E_{i,0} + \beta_3 G_{i,0} + \sum_{j=4}^{k} \beta_j X_{j,i} + \epsilon_i$$

where hats indicate long-run growth rates, \$ E\$ is the log of emissions per-capita, and G is the log of GDP per capita. X_i is a vector of explanatory and control variables. The sample mean has been deducted from each continuous level variable in this vector.

The third term, $\hat{G}_I G_{i,0}$, is the interaction between the rate of economic growth and the initial level of log income per capita. The EKC turning point is calculated with the assumption that $\alpha_1 > 0$ and $\beta_1 < 0$ when $\frac{\partial \hat{E}}{\partial \hat{G}} = 0$ and $\tau = exp\left(-\frac{\alpha_1}{\beta_1} + \mu_G\right)$, where μ_G is the cross-country mean of initial GDP per capita variable prior to the estimation

The following variables are contained in $X_{j,i}$: Dummy variable for countries that are centrally planned or market economy, dummies for countries that have legal origins in the UK, France, Germany or any Scandinavian countries, average summer and winter temperatures, de-meaned log of fossil fuel consumption in 1971 and de-meaned log of population density in 1971.

[4]:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from sklearn.linear_model import LinearRegression
from scipy.stats import chi2

df = pd.read_excel(r'D:\Stern_etal2017EDE.xlsx', sheet_name='SO2 1971-2005')

# Run OLS regression
reg1 = smf.ols("GREPC ~ GRGDPPC + DMMULGDP + DMLNGDPPC1971 + DMLNEPC1971 + CPE␣
↪+ legorfr + legorge + legorsc + DMSUMT + DMWINT + DMLNFFPC1971 +␣
↪DMLNPOPD1971", data=df).fit(cov_type='HC3')
print(reg1.summary())

#Different Types of Robust Standard Errors
#'HC0': The original White (1980) heteroskedasticity-consistent standard errors.
#'HC1': A small sample correction to HC0.
#'HC2': Another small sample correction, which is more reliable when the sample␣
↪size is small.
#'HC3': MacKinnon and White's (1985) heteroskedasticity-consistent standard␣
↪errors, which is more robust in small samples.

#print(reg1.resid())

#Create a matrix of independent varibales which can be used for the residual␣
↪tests
exog_het = reg1.model.exog

# Perform the Breusch-Pagan Test for Heteroskedasticity. Note exog_het is the␣
↪matrix of explanatory (independent) variables
bp_test = het_breuschpagan(reg1.resid,exog_het)
```

```
print("Breusch-Pagan Test for Heteroskedasticity:") # H0: Variance of the␣
↪residuals is constant (homoskedastic)
print (f"LM Statistics: {bp_test[0]}") # The Lagrange Multiplier (LM) test␣
↪follows a chi-sq distribution and the degree of freedom is the number of␣
↪independent variables. The LM statistic is calculated based on the residuals␣
↪from the regression model and the independent variables
print (f"LM Test p-value: {bp_test[1]}") # If p-value is less than 0.05 (5%␣
↪level) then we reject the null and vice-versa
#print (f"F-Statistics: {bp_test[2]}") # F-stat is the ratio of the mean␣
↪squared error of the model and the mean squared error of the residual.
#print (f"F-Statistics p-value: {bp_test[3]}")

# Performing the White test for heteroskedasticity
white_test = het_white(reg1.resid, exog_het) # H0: Homoskedasticity

# Extracting the test statistic and p-value
white_stat = white_test[0]
white_pvalue = white_test[1]

print(f"White test statistic: {white_stat}")
print(f"White test p-value: {white_pvalue}")

#print("White Test for Heteroskedasticity:") # Another way of writing the above␣
↪command of white test and p value
#print(f"T Statistics: {white_test[0]}")
#print(f"P-value: {white_test[1]}")
#print(f"F-statistics: {white_test[2]}")
#print(f"F-test p-value: {white_test[3]}")

# Interpretation
if white_pvalue < 0.05:  # Typically, we use a significance level of 0.05
    print("Reject the null hypothesis of homoskedasticity. Evidence of␣
↪heteroskedasticity.")
else:
    print("Fail to reject the null hypothesis of homoskedasticity. No evidence␣
↪of heteroskedasticity.")

r1s= reg1.resid ** 2 # This command enables python to store the squared of␣
↪residuals in r1s

# Assuming df is your DataFrame and PWTPOP1971 is the column you want to invert
df['invpop'] = 1 / df['PWTPOP1971']

df['lpop1971'] = np.log (df['PWTPOP1971']) # To create a new varible with log

df['rls']=reg1.resid**2 # stored the residual square in the data file.
```

```
# To check the addition of the new variable invop: print(df.head())

# Trying to run an auxiliary regression following the codes of Stern:

#reg2 = smf.ols("r1s ~ invpop", data=df).fit() (Error: Number of rows mismatch␣
 ↪between data argument and r1s (136 versus 134))
#print(reg2.summary())

# Test for the EKC turning point.
alpha1 = reg1.params['GRGDPPC'] # Storing the estimates of GRGDPPC
beta1 = reg1.params['DMMULGDP']
turning_point= np.exp(- alpha1/beta1 + 8.3187336) # Deriving the EKC turning␣
 ↪point level : log of LNGDPPC1971
print(f"EKC Truning point:{turning_point}")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  GREPC   R-squared:                       0.744
Model:                            OLS   Adj. R-squared:                  0.709
Method:                 Least Squares   F-statistic:                     16.45
Date:                Thu, 27 Jun 2024   Prob (F-statistic):           1.28e-17
Time:                        11:04:57   Log-Likelihood:                 226.77
No. Observations:                 100   AIC:                            -427.5
Df Residuals:                      87   BIC:                            -393.7
Df Model:                          12
Covariance Type:                  HC3
===============================================================================
=
                   coef    std err          z      P>|z|      [0.025
0.975]
-------------------------------------------------------------------------------
-
Intercept       -0.0099      0.007     -1.500      0.134      -0.023
0.003
GRGDPPC          0.8543      0.187      4.563      0.000       0.487
1.221
DMMULGDP        -0.2668      0.156     -1.715      0.086      -0.572
0.038
DMLNGDPPC1971    0.0197      0.007      3.026      0.002       0.007
0.032
DMLNEPC1971     -0.0209      0.005     -3.963      0.000      -0.031
-0.011
CPE              0.0292      0.018      1.610      0.107      -0.006
0.065
legorfr         -0.0151      0.007     -2.318      0.020      -0.028
-0.002
```

```
legorge          -0.0356      0.013     -2.744      0.006      -0.061
-0.010
legorsc          -0.0446      0.021     -2.177      0.029      -0.085
-0.004
DMSUMT            0.0028      0.001      3.035      0.002       0.001
0.005
DMWINT           -0.0010      0.000     -2.188      0.029      -0.002
-0.000
DMLNFFPC1971     -0.0015      0.001     -1.358      0.174      -0.004
0.001
DMLNPOPD1971     -0.0093      0.003     -3.293      0.001      -0.015
-0.004
==============================================================================
Omnibus:                        6.532   Durbin-Watson:                   1.745
Prob(Omnibus):                  0.038   Jarque-Bera (JB):                5.950
Skew:                           0.531   Prob(JB):                       0.0511
Kurtosis:                       3.548   Cond. No.                         893.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
Breusch-Pagan Test for Heteroskedasticity:
LM Statistics: 21.556851039733115
LM Test p-value: 0.042797192474353325
White test statistic: 86.7873337082127
White test p-value: 0.07264959660445151
Fail to reject the null hypothesis of homoskedasticity. No evidence of
heteroskedasticity.
EKC Truning point:100732.05913166571
```

1.15 Breusch Pagan, Harvey test and LR test as in Stern's code:

The method adopted by Stern et al. to estimate the Breusch-Pagan and Harvey tests for heteroskedasticity for all the estimations is not reproducible using their codes. These are the following codes used by the authors:

set invpop = 1/PWTPOP1971

set lpop1971 = log(PWTPOP1971)

source regwhitetest.src

@regwhitetest(type=full)

set r1s = r1**2

LINREG(noprint) r1s

#Constant INVPOP

cdf(title="Breusch-Pagan Test for INVPOP") chisqr %trsquared 1

```
legorge          -0.0356      0.013      -2.744      0.006      -0.061
-0.010
legorsc          -0.0446      0.021      -2.177      0.029      -0.085
-0.004
DMSUMT            0.0028      0.001       3.035      0.002       0.001
0.005
DMWINT           -0.0010      0.000      -2.188      0.029      -0.002
-0.000
DMLNFFPC1971     -0.0015      0.001      -1.358      0.174      -0.004
0.001
DMLNPOPD1971     -0.0093      0.003      -3.293      0.001      -0.015
-0.004
==============================================================================
Omnibus:                        6.532   Durbin-Watson:                   1.745
Prob(Omnibus):                  0.038   Jarque-Bera (JB):                5.950
Skew:                           0.531   Prob(JB):                       0.0511
Kurtosis:                       3.548   Cond. No.                         893.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
Breusch-Pagan Test for Heteroskedasticity:
LM Statistics: 21.556851039733115
LM Test p-value: 0.042797192474353325
White test statistic: 86.7873337082127
White test p-value: 0.07264959660445151
Fail to reject the null hypothesis of homoskedasticity. No evidence of
heteroskedasticity.
EKC Truning point:100732.05913166571
```