```
#(a). Run an OLS regression, including at least one independent variable and a time variable (as dummies).
#Explain how you think your independent variable relates to your dependent variable.
#Did you find what you expected to find?
!pip install linearmodels
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from linearmodels.panel import PanelOLS
from linearmodels.panel import RandomEffects
from linearmodels.panel import compare

```

```
Collecting linearmodels
  Downloading linearmodels-6.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (7.9 kB)
Requirement already satisfied: numpy<3,>=1.22.3 in /usr/local/lib/python3.10/dist-packages (from linearmodels) (1.26.4)
Requirement already satisfied: pandas>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from linearmodels) (2.2.2)
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from linearmodels) (1.13.1)
Requirement already satisfied: statsmodels>=0.13.0 in /usr/local/lib/python3.10/dist-packages (from linearmodels) (0.14.4)
Collecting mypy-extensions>=0.4 (from linearmodels)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: Cython>=3.0.10 in /usr/local/lib/python3.10/dist-packages (from linearmodels) (3.0.11)
Collecting pyhdfe>=0.1 (from linearmodels)
  Downloading pyhdfe-0.2.0-py3-none-any.whl.metadata (4.0 kB)
Collecting formulaic>=1.0.0 (from linearmodels)
  Downloading formulaic-1.0.2-py3-none-any.whl.metadata (6.8 kB)
Collecting setuptools-scm<9.0.0,>=8.0.0 (from setuptools-scm[toml]<9.0.0,>=8.0.0->linearmodels)
  Downloading setuptools_scm-8.1.0-py3-none-any.whl.metadata (6.6 kB)
Collecting interface-meta>=1.2.0 (from formulaic>=1.0.0->linearmodels)
  Downloading interface_meta-1.3.0-py3-none-any.whl.metadata (6.7 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from formulaic>=1.0.0->linearmodels)
Requirement already satisfied: wrapt>=1.0 in /usr/local/lib/python3.10/dist-packages (from formulaic>=1.0.0->linearmodels) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.4.0->linearmodels) (2.8
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.4.0->linearmodels) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.4.0->linearmodels) (2024.2)
Requirement already satisfied: packaging>=20 in /usr/local/lib/python3.10/dist-packages (from setuptools-scm<9.0.0,>=8.0.0->setuptools-s
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from setuptools-scm<9.0.0,>=8.0.0->setuptools-scm[
Requirement already satisfied: tomli>=1 in /usr/local/lib/python3.10/dist-packages (from setuptools-scm<9.0.0,>=8.0.0->setuptools-scm[tc
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.13.0->linearmodels) (1.0.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.4.0->linearmc
Downloading linearmodels-6.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 12.8 MB/s eta 0:00:00
Downloading formulaic-1.0.2-py3-none-any.whl (94 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.5/94.5 kB 6.6 MB/s eta 0:00:00
Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Downloading pyhdfe-0.2.0-py3-none-any.whl (19 kB)
Downloading setuptools_scm-8.1.0-py3-none-any.whl (43 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.7/43.7 kB 2.5 MB/s eta 0:00:00
Downloading interface_meta-1.3.0-py3-none-any.whl (14 kB)
Installing collected packages: setuptools-scm, mypy-extensions, interface-meta, pyhdfe, formulaic, linearmodels
Successfully installed formulaic-1.0.2 interface-meta-1.3.0 linearmodels-6.1 mypy-extensions-1.0.0 pyhdfe-0.2.0 setuptools-scm-8.1.0
```

```
url = 'https://www.qogdata.pol.gu.se/data/qog_bas_ts_jan24.xlsx'
data = pd.read_excel(url)
data.head()
```

	ccode	cname	year	ccode_qog	cname_qog	ccodealp	ccodecow	version	cname_year	ccodealp_year	...	wdi_trade	wdi_unem
0	4	Afghanistan	1946	4	Afghanistan	AFG	700.0	QoGBasTSjan24	Afghanistan 1946	AFG46	...	NaN	
1	4	Afghanistan	1947	4	Afghanistan	AFG	700.0	QoGBasTSjan24	Afghanistan 1947	AFG47	...	NaN	
2	4	Afghanistan	1948	4	Afghanistan	AFG	700.0	QoGBasTSjan24	Afghanistan 1948	AFG48	...	NaN	
3	4	Afghanistan	1949	4	Afghanistan	AFG	700.0	QoGBasTSjan24	Afghanistan 1949	AFG49	...	NaN	
4	4	Afghanistan	1950	4	Afghanistan	AFG	700.0	QoGBasTSjan24	Afghanistan 1950	AFG50	...	NaN	

5 rows × 251 columns

```
# Prepare the data by dropping rows with missing values in relevant columns
regression_data = data[['wdi_birth', 'wdi_unempyfilo', 'year']].dropna()
```

I am going to run an OLS regression with female unemployment rate as my dependent variable and birth rate as my independent variable with year dummies. Unemployment rate is quite complex but I believe that increases in birth rate may also drive female unemployment rates up due to costs in maternity leave or liability reasons. These are just speculations, however.

```
# Run an OLS regression without country fixed effects, only using the urban population and year fixed effects
ols_model = smf.ols(formula='wdi_unempyfilo ~ wdi_birth + C(year)', data=regression_data).fit()

# Display the summary of the OLS regression
ols_model.summary()
```

OLS Regression Results

Dep. Variable:	wdi_unempyfilo	**R-squared:**	0.050
Model:	OLS	**Adj. R-squared:**	0.044
Method:	Least Squares	**F-statistic:**	9.095
Date:	Sat, 16 Nov 2024	**Prob (F-statistic):**	5.08e-41
Time:	22:28:57	**Log-Likelihood:**	-21920.
No. Observations:	5430	**AIC:**	4.390e+04
Df Residuals:	5398	**BIC:**	4.411e+04
Df Model:	31		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	23.7271	1.217	19.495	0.000	21.341	26.113
C(year)[T.1992]	-0.6397	1.532	-0.418	0.676	-3.643	2.364
C(year)[T.1993]	0.2892	1.524	0.190	0.849	-2.698	3.277
C(year)[T.1994]	0.6213	1.524	0.408	0.684	-2.367	3.609
C(year)[T.1995]	0.8799	1.525	0.577	0.564	-2.109	3.869
C(year)[T.1996]	1.2269	1.525	0.805	0.421	-1.762	4.216
C(year)[T.1997]	1.0214	1.525	0.670	0.503	-1.969	4.011
C(year)[T.1998]	0.7866	1.526	0.516	0.606	-2.204	3.777
C(year)[T.1999]	0.9716	1.526	0.637	0.524	-2.020	3.963
C(year)[T.2000]	0.7015	1.526	0.460	0.646	-2.290	3.693
C(year)[T.2001]	0.7971	1.527	0.522	0.602	-2.196	3.790
C(year)[T.2002]	0.9987	1.525	0.655	0.512	-1.990	3.988
C(year)[T.2003]	1.4179	1.525	0.930	0.353	-1.572	4.407
C(year)[T.2004]	1.2677	1.525	0.831	0.406	-1.722	4.258
C(year)[T.2005]	0.8276	1.525	0.543	0.587	-2.163	3.818
C(year)[T.2006]	0.5769	1.522	0.379	0.705	-2.406	3.560
C(year)[T.2007]	-0.2288	1.522	-0.150	0.881	-3.212	2.755
C(year)[T.2008]	-0.3235	1.522	-0.213	0.832	-3.307	2.660
C(year)[T.2009]	1.0962	1.522	0.720	0.471	-1.888	4.080
C(year)[T.2010]	1.5755	1.522	1.035	0.301	-1.409	4.560
C(year)[T.2011]	1.7286	1.521	1.137	0.256	-1.252	4.709
C(year)[T.2012]	1.9738	1.521	1.298	0.194	-1.008	4.955
C(year)[T.2013]	2.1458	1.521	1.410	0.158	-0.837	5.128
C(year)[T.2014]	1.9029	1.522	1.251	0.211	-1.080	4.886
C(year)[T.2015]	1.5734	1.522	1.034	0.301	-1.411	4.558
C(year)[T.2016]	1.2746	1.523	0.837	0.403	-1.711	4.260
C(year)[T.2017]	0.7870	1.524	0.517	0.605	-2.200	3.774
C(year)[T.2018]	0.1637	1.524	0.107	0.914	-2.824	3.152
C(year)[T.2019]	-0.2728	1.525	-0.179	0.858	-3.262	2.716
C(year)[T.2020]	2.5970	1.526	1.702	0.089	-0.394	5.588
C(year)[T.2021]	1.5426	1.530	1.008	0.313	-1.457	4.542
wdi_birth	-0.2528	0.016	-15.502	0.000	-0.285	-0.221

Omnibus:	1021.562	**Durbin-Watson:**	0.093
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1742.050
Skew:	1.237	**Prob(JB):**	0.00
Kurtosis:	4.257	**Cond. No.**	899.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

We see that most years, there is in fact a positive relationship between birth rate and unemployment rate. However, not one of the years listed is statistically significant, and thus we cannot conclude anything about this relationship at the 5% level.

```
#Then run a fixed effect model version of that OLS model.  Interpret your results.
#Did you find what you expected to find?  Why?  Why not?
from linearmodels.panel import PanelOLS
import statsmodels.formula.api as smf
```

```python

# Prepare the data by dropping rows with missing values in relevant columns
regression_data = data[['wdi_birth', 'wdi_unempyfilo', 'year', 'cname', 'gle_cgdpc']].dropna()

# Set the time and entity index
regression_data = regression_data.set_index(['cname', 'year'])

# Fit the fixed effects model
fixed_effects_model = PanelOLS.from_formula('wdi_unempyfilo ~ wdi_birth + EntityEffects + TimeEffects',
data=regression_data).fit(cov_type='clustered', cluster_entity=True)

# Display the summary of the regression
print(fixed_effects_model.summary)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:         wdi_unempyfilo   R-squared:                        0.0283
Estimator:                   PanelOLS   R-squared (Between):             -1.0001
No. Observations:                3651   R-squared (Within):               0.0467
Date:                Sat, Nov 16 2024   R-squared (Overall):             -0.9830
Time:                        22:28:57   Log-likelihood                -1.048e+04
Cov. Estimator:             Clustered
                                        F-statistic:                      100.65
Entities:                         178   P-value                           0.0000
Avg Obs:                       20.511   Distribution:                  F(1,3452)
Min Obs:                       1.0000
Max Obs:                       21.000   F-statistic (robust):             11.734
                                        P-value                           0.0006
Time periods:                      21   Distribution:                  F(1,3452)
Avg Obs:                       173.86
Min Obs:                       152.00
Max Obs:                       178.00

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
wdi_birth     -0.4301     0.1255    -3.4255     0.0006     -0.6762     -0.1839
==============================================================================

F-test for Poolability: 152.02
P-value: 0.0000
Distribution: F(197,3452)

Included effects: Entity, Time
```

The fixed effects model revealed a stronger negative relationship between birth rates and unemployment than the OLS model suggested. This is likely because it effectively accounted for unobserved, time-invariant factors that bias OLS estimates. The results are also statistically significant as opposed to the insignificant results of the OLS model.

```python
#Then include an additional predictor in your fixed effects model that you think might account
#for the initial relationship you found between your X and your Y.
#What effect does that new independent variable have in your new regression?
# Fit the fixed effects model
fixed_effects_model_with_education = PanelOLS.from_formula('wdi_unempyfilo ~ wdi_birth + gle_cgdpc + EntityEffects + TimeEffects',
data=regression_data).fit(cov_type='clustered', cluster_entity=True)

# Display the summary of the regression
print(fixed_effects_model_with_education.summary)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:         wdi_unempyfilo   R-squared:                        0.0307
Estimator:                   PanelOLS   R-squared (Between):             -0.9753
No. Observations:                3651   R-squared (Within):               0.0521
Date:                Sat, Nov 16 2024   R-squared (Overall):             -0.9580
Time:                        22:28:58   Log-likelihood                -1.048e+04
Cov. Estimator:             Clustered
                                        F-statistic:                      54.677
Entities:                         178   P-value                           0.0000
Avg Obs:                       20.511   Distribution:                  F(2,3451)
Min Obs:                       1.0000
Max Obs:                       21.000   F-statistic (robust):             6.3279
                                        P-value                           0.0018
Time periods:                      21   Distribution:                  F(2,3451)
Avg Obs:                       173.86
Min Obs:                       152.00
Max Obs:                       178.00
```

```
                          Parameter Estimates
==========================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------
wdi_birth     -0.4036     0.1248    -3.2335     0.0012     -0.6484     -0.1589
gle_cgdpc  -5.779e-05  4.669e-05    -1.2377     0.2159     -0.0001   3.376e-05
==========================================================================

F-test for Poolability: 144.95
P-value: 0.0000
Distribution: F(197,3451)

Included effects: Entity, Time
```

We see that the GDP variable doesn't seem to have much more of an impact on the relationship between the birth rate variable and the unemployment variable. This is probably because GDP is another highly complex variable that may have heterogeniety that even this model cannot account for. The R-squared of the model did improve by roughly a percentage point, however.

```
#(d) Then run a random effects model equivalent to your fixed effects model in step (b).  Interpret the results.
# For random effects, we use the Random Effects model from statsmodels
import statsmodels.api as sm
from linearmodels.panel import RandomEffects

# Run the random effects model
random_effects_model = RandomEffects.from_formula('wdi_unempyfilo ~ wdi_birth + gle_cgdpc',
data=regression_data).fit(cov_type='clustered', cluster_entity=True)

# Display the summary of the random effects model
print(random_effects_model.summary)
```

```
                        RandomEffects Estimation Summary
============================================================================
Dep. Variable:         wdi_unempyfilo   R-squared:                      0.0034
Estimator:              RandomEffects   R-squared (Between):           -0.1374
No. Observations:                3651   R-squared (Within):             0.0176
Date:                Sat, Nov 16 2024   R-squared (Overall):           -0.1345
Time:                        22:28:58   Log-likelihood                 -1.077e+04
Cov. Estimator:             Clustered
                                        F-statistic:                    6.2372
Entities:                         178   P-value                         0.0020
Avg Obs:                       20.511   Distribution:                F(2,3649)
Min Obs:                       1.0000
Max Obs:                       21.000   F-statistic (robust):           1.2605
                                        P-value                         0.2836
Time periods:                      21   Distribution:                F(2,3649)
Avg Obs:                       173.86
Min Obs:                       152.00
Max Obs:                       178.00

                          Parameter Estimates
==========================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------
wdi_birth     -0.0818     0.0531    -1.5423     0.1231     -0.1859      0.0222
gle_cgdpc   1.359e-05   3.84e-05     0.3540     0.7234   -6.17e-05   8.889e-05
==========================================================================
```

We see that the model does not explain any relationships well. It is most likely due to the fact that a random effects model cannot account for the heterogeneity that is almost certainly present in variables such as birth rate, unemployment rate, and GDP. This is evidenced by the p-values and the R-squared.

```
# Hausman test - comparing random and fixed effects
result = compare({'Random Effects': random_effects_model, 'Fixed Effects': fixed_effects_model_with_education})
print(result)
```

```
                       Model Comparison
=============================================================
                          Random Effects      Fixed Effects
-------------------------------------------------------------
Dep. Variable             wdi_unempyfilo     wdi_unempyfilo
Estimator                  RandomEffects           PanelOLS
No. Observations                    3651               3651
Cov. Est.                      Clustered          Clustered
R-squared                         0.0034             0.0307
R-Squared (Within)                0.0176             0.0521
R-Squared (Between)              -0.1374            -0.9753
R-Squared (Overall)              -0.1345            -0.9580
```

```
F-statistic                      6.2372            54.677
P-value (F-stat)                 0.0020            0.0000
====================     ================   ================
wdi_birth                       -0.0818           -0.4036
                              (-1.5423)         (-3.2335)
gle_cgdpc                     1.359e-05        -5.779e-05
                               (0.3540)         (-1.2377)
======================= ================== ==================
Effects                                              Entity
                                                       Time
-------------------------------------------------------------

T-stats reported in parentheses
```

```python
# Building our own Hausman test from scratch

import numpy as np
from scipy import stats

# Random Effects model
random_effects_model = RandomEffects.from_formula('wdi_unempyfilo ~ wdi_birth + gle_cgdpc', data=regression_data).fit()

# Fixed Effects model
fixed_effects_model = PanelOLS.from_formula('wdi_unempyfilo ~ wdi_birth + gle_cgdpc + EntityEffects', data=regression_data).fit()

# Extract the coefficients
b_fixed = fixed_effects_model.params
b_random = random_effects_model.params

# Extract the variance-covariance matrices
v_fixed = fixed_effects_model.cov
v_random = random_effects_model.cov

# Calculate the difference in coefficients
b_diff = b_fixed - b_random

# Calculate the variance of the difference
v_diff = v_fixed - v_random

# Hausman test statistic
hausman_stat = b_diff.T @ np.linalg.inv(v_diff) @ b_diff

# Degrees of freedom (number of coefficients being compared)
df = len(b_diff)

# Calculate p-value
p_value = 1 - stats.chi2.cdf(hausman_stat, df)

# Display the test statistic and p-value
print(f"Hausman test statistic: {hausman_stat}")
print(f"P-value: {p_value}")

```

```
Hausman test statistic: 438.88580823829324
P-value: 0.0
```

The test shows that the variables are only significant for the fixed effects model but not the random effects model. This lines up with what we found with the previous random effects model. The fixed effects model also produces a larger absolute coefficient for wdi_birth (-0.4301 vs. -0.2528). This suggests that, after accounting for unobserved entity-level heterogeneity, the effect of birth rates on unemployment is more pronounced than what OLS estimates indicated. Overall, as we expected, the fixed effects model accounts for heterogeniety much better than the OLS model and produces better results. Additionally, a very large Hausman test statistic indicates that there is a substantial difference between the coefficients of the fixed effects model and the random effects model, which is probably because the random effects model violates its assumptions.