```
!pip install linearmodels
```

```
Requirement already satisfied: linearmodels in /usr/local/lib/python3.12/dist-packages (7.0)
Requirement already satisfied: numpy<3,>=1.22.3 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (2
Requirement already satisfied: pandas>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (2.2.
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (1.16.
Requirement already satisfied: statsmodels>=0.13.0 in /usr/local/lib/python3.12/dist-packages (from linearmodels)
Requirement already satisfied: mypy_extensions>=0.4 in /usr/local/lib/python3.12/dist-packages (from linearmodels
Requirement already satisfied: pyhdfe>=0.1 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (0.2.0)
Requirement already satisfied: formulaic>=1.2.1 in /usr/local/lib/python3.12/dist-packages (from linearmodels) (1
Requirement already satisfied: interface-meta>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from formulaic>=
Requirement already satisfied: narwhals>=1.17 in /usr/local/lib/python3.12/dist-packages (from formulaic>=1.2.1->
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.12/dist-packages (from formulai
Requirement already satisfied: wrapt>=1.0 in /usr/local/lib/python3.12/dist-packages (from formulaic>=1.2.1->line
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.4.0->linea
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.4.0->lin
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-packages (from statsmodels>=0.13.0-
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.12/dist-packages (from statsmodels>=0.13
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->
```

```
from google.colab import files
import pandas as pd
import numpy as np
import io

uploaded = files.upload()
df = pd.read_csv(io.BytesIO(list(uploaded.values())[0]))
df.head()
```

Choose files macro_panel_updated.csv

macro_panel_updated.csv(text/csv) - 151209 bytes, last modified: 19/11/2025 - 100% done

Saving macro_panel_updated.csv to macro_panel_updated (3).csv

	country	year	quarter	npl_ratio	euribor_3m	gdp_growth	hicp_yoy	unemployment	inflation_yoy
0	AT	2000	1	NaN	3.542300	4.600445	0.0	3.8	1.733333
1	AT	2000	2	NaN	4.263000	4.600445	0.0	3.8	1.933333
2	AT	2000	3	NaN	4.737600	4.600445	0.0	3.8	2.033333
3	AT	2000	4	NaN	5.024167	4.600445	0.0	3.8	2.100000
4	BE	2000	1	NaN	3.542300	5.806174	0.0	7.1	1.633333

Next steps: Generate code with df | New interactive sheet

```
# Standardize
df.columns = df.columns.str.strip().str.lower()

# DROP the wrong inflation column
if "hicp_yoy" in df.columns:
    df = df.drop(columns=["hicp_yoy"])

# Keep only the correct inflation column
print(df.columns.tolist())
```

```
['country', 'year', 'quarter', 'npl_ratio', 'euribor_3m', 'gdp_growth', 'unemployment', 'inflation_yoy']
```

```
df["time_id"] = df["year"] * 4 + df["quarter"]

num_cols = ["year","quarter","time_id","npl_ratio",
            "gdp_growth","unemployment","inflation_yoy","euribor_3m"]

for c in num_cols:
    df[c] = pd.to_numeric(df[c], errors="coerce")
```

```
df = df.dropna(subset=["npl_ratio"])
df = df.sort_values(["country","time_id"]).reset_index(drop=True)

df.head()
```

	country	year	quarter	npl_ratio	euribor_3m	gdp_growth	unemployment	inflation_yoy	time_id
0	AT	2006	1	2.739429	2.611567	5.380802	5.7	1.433333	8025
1	AT	2006	2	2.739429	2.889500	5.380802	5.7	2.033333	8026
2	AT	2006	3	2.739429	3.221367	5.380802	5.7	1.800000	8027
3	AT	2006	4	2.739429	3.594467	5.380802	5.7	1.433333	8028
4	AT	2007	1	2.242467	3.820333	6.119790	5.3	1.766667	8029

Next steps: Generate code with df | New interactive sheet

```
df_panel = df.set_index(["country","time_id"]).sort_index()
df_panel.head()
```

		year	quarter	npl_ratio	euribor_3m	gdp_growth	unemployment	inflation_yoy
country	**time_id**							
AT	**8025**	2006	1	2.739429	2.611567	5.380802	5.7	1.433333
	8026	2006	2	2.739429	2.889500	5.380802	5.7	2.033333
	8027	2006	3	2.739429	3.221367	5.380802	5.7	1.800000
	8028	2006	4	2.739429	3.594467	5.380802	5.7	1.433333
	8029	2007	1	2.242467	3.820333	6.119790	5.3	1.766667

Next steps: Generate code with df_panel | New interactive sheet

```
vars_analysis = ["npl_ratio","gdp_growth","unemployment","inflation_yoy","euribor_3m"]
print(df_panel[vars_analysis].describe())
```

```
         npl_ratio   gdp_growth  unemployment  inflation_yoy   euribor_3m
count  1288.000000  1288.000000   1288.000000    1288.000000  1288.000000
mean      6.262905     4.332987      8.822360       2.477666     0.964973
std       8.042270     5.759178      4.539123       3.191348     1.636366
min       0.145588   -21.188764      3.100000      -3.866667    -0.566400
25%       1.992508     1.906078      5.900000       0.633333    -0.319500
50%       3.351983     4.012819      7.350000       1.816667     0.223500
75%       6.375134     7.305004     10.300000       3.166667     1.772133
max      47.747846    35.737719     27.800000      24.166667     4.981800
```

```
print(df_panel[vars_analysis].corr())
```

```
               npl_ratio  gdp_growth  unemployment  inflation_yoy  euribor_3m
npl_ratio       1.000000   -0.235386      0.624301      -0.261648   -0.199448
gdp_growth     -0.235386    1.000000     -0.340156       0.384545    0.099805
unemployment    0.624301   -0.340156      1.000000      -0.265070   -0.192088
inflation_yoy  -0.261648    0.384545     -0.265070       1.000000    0.292140
euribor_3m     -0.199448    0.099805     -0.192088       0.292140    1.000000
```

```
from linearmodels.panel import PanelOLS
import statsmodels.api as sm

y = df_panel["npl_ratio"]
X = df_panel[["gdp_growth","unemployment","inflation_yoy","euribor_3m"]]

fe = PanelOLS(y, X, entity_effects=True, check_rank=False)
fe_res = fe.fit(cov_type="clustered", cluster_entity=True)
print(fe_res)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:              npl_ratio   R-squared:                        0.5060
Estimator:                   PanelOLS   R-squared (Between):              0.1981
No. Observations:                1288   R-squared (Within):               0.5060
Date:                Tue, Nov 25 2025   R-squared (Overall):              0.3497
Time:                        22:12:57   Log-likelihood                   -3682.2
Cov. Estimator:             Clustered
                                        F-statistic:                      323.96
Entities:                          19   P-value                           0.0000
Avg Obs:                       67.789   Distribution:                  F(4,1265)
Min Obs:                       12.000
Max Obs:                       80.000   F-statistic (robust):             8.2919
                                        P-value                           0.0000
Time periods:                      80   Distribution:                  F(4,1265)
Avg Obs:                       16.100
Min Obs:                       4.0000
Max Obs:                       19.000
```

```
                             Parameter Estimates
==============================================================================
               Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
gdp_growth        0.0231     0.0713     0.3236     0.7463     -0.1168      0.1630
unemployment      1.3405     0.2703     4.9592     0.0000      0.8102      1.8707
inflation_yoy    -0.0331     0.0793    -0.4174     0.6765     -0.1888      0.1225
euribor_3m       -0.3762     0.1661    -2.2651     0.0237     -0.7020     -0.0504
==============================================================================

F-test for Poolability: 81.939
P-value: 0.0000
Distribution: F(18,1265)

Included effects: Entity
```

```
from linearmodels.panel import RandomEffects

X_re = sm.add_constant(X)
re = RandomEffects(y, X_re)
re_res = re.fit()
print(re_res)
```

```
                        RandomEffects Estimation Summary
================================================================================
Dep. Variable:              npl_ratio   R-squared:                        0.5028
Estimator:             RandomEffects   R-squared (Between):              0.1520
No. Observations:               1288   R-squared (Within):               0.5060
Date:                Tue, Nov 25 2025   R-squared (Overall):              0.3780
Time:                        22:13:17   Log-likelihood                   -3691.9
Cov. Estimator:            Unadjusted
                                        F-statistic:                      324.35
Entities:                          19   P-value                           0.0000
Avg Obs:                       67.789   Distribution:                  F(4,1283)
Min Obs:                       12.000
Max Obs:                       80.000   F-statistic (robust):             324.54
                                        P-value                           0.0000
Time periods:                      80   Distribution:                  F(4,1283)
Avg Obs:                       16.100
Min Obs:                       4.0000
Max Obs:                       19.000

                             Parameter Estimates
==============================================================================
               Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const            -5.1863     1.2296    -4.2179     0.0000     -7.5986     -2.7741
gdp_growth        0.0221     0.0247     0.8935     0.3718     -0.0264      0.0706
unemployment      1.3303     0.0431     30.855     0.0000      1.2457      1.4149
inflation_yoy    -0.0363     0.0447    -0.8118     0.4170     -0.1239      0.0514
euribor_3m       -0.3759     0.0778    -4.8301     0.0000     -0.5286     -0.2232
==============================================================================
```

```
from scipy import stats

beta = ["gdp_growth","unemployment","inflation_yoy","euribor_3m"]

b_fe = fe_res.params[beta]
b_re = re_res.params[beta]

V_fe = fe_res.cov.loc[beta, beta]
V_re = re_res.cov.loc[beta, beta]

diff = b_fe - b_re
Vdiff = V_fe - V_re

H = float(diff.T @ np.linalg.inv(Vdiff) @ diff)
p = 1 - stats.chi2.cdf(H, len(diff))

print("Hausman chi2:", H)
print("p-value:", p)
```

```
Hausman chi2: -0.03990132049242252
p-value: 1.0
```

```
df_panel.reset_index().to_csv("macro_clean_final.csv", index=False)
from google.colab import files
files.download("macro_clean_final.csv")
```