In [1]:
```
import pandas as pd
import numpy as np
import datetime

# import data
data = pd.read_csv("Bank data.csv", index_col = ["bank", "year"],
                   parse_dates = True)
```

In [2]:
```
data
```

Out[2]:
```
                     Bank ID     ROE     ROA  Solvency Ratio  Liquidity Ratio
bank       year
      2015-01-01           1  0.0934  0.0092            8.50             1.02
      2016-01-01           1  0.0903  0.0092            8.80             1.01
 JPM  2017-01-01           1  0.0878  0.0088            8.91             1.02
      2018-01-01           1  0.1194  0.0118            9.22             1.00
      2019-01-01           1  0.1326  0.0127            9.28             0.92
      2015-01-01           2  0.0934  0.0092            8.50             1.02
      2016-01-01           2  0.0903  0.0092            8.80             1.01
 BAC  2017-01-01           2  0.0878  0.0088            8.91             1.02
      2018-01-01           2  0.1194  0.0118            9.22             1.00
      2019-01-01           2  0.1321  0.0127            9.28             0.92
      2015-01-01           3  0.1117  0.0123            8.22             0.93
      2016-01-01           3  0.1012  0.0107            8.63             0.92
 WFC  2017-01-01           3  0.0998  0.0106            8.38             0.88
      2018-01-01           3  0.1023  0.0109            8.62             0.88
      2019-01-01           3  0.0919  0.0093            9.25             0.87
      2015-01-01           4  0.1227  0.0135            8.01             0.83
      2016-01-01           4  0.1166  0.0127            8.30             0.83
 USB  2017-01-01           4  0.1204  0.0129            8.30             0.82
      2018-01-01           4  0.1339  0.0146            8.05             0.85
      2019-01-01           4  0.1237  0.0136            8.44             0.83
      2015-01-01           5  0.0741  0.0097            6.68             0.88
      2016-01-01           5  0.0766  0.0103            6.33             0.88
 TFC  2017-01-01           5  0.0740  0.0100            6.46             0.87
      2018-01-01           5  0.1024  0.0137            6.48             0.92
      2019-01-01           5  0.0750  0.0104            6.11             0.94
```

In [3]:

```
data.index.get_level_values("year")
```

Out[3]:

```
DatetimeIndex(['2015-01-01', '2016-01-01', '2017-01-01', '2018-01-01',
               '2019-01-01', '2015-01-01', '2016-01-01', '2017-01-01',
               '2018-01-01', '2019-01-01', '2015-01-01', '2016-01-01',
               '2017-01-01', '2018-01-01', '2019-01-01', '2015-01-01',
               '2016-01-01', '2017-01-01', '2018-01-01', '2019-01-01',
               '2015-01-01', '2016-01-01', '2017-01-01', '2018-01-01',
               '2019-01-01'],
              dtype='datetime64[ns]', name='year', freq=None)
```

In [4]:

```
diff_index = data.groupby(level=0).diff(-1).dropna().index
```

In [5]:

```
diff_index
```

Out[5]:

```
MultiIndex([('JPM', '2015-01-01'),
            ('JPM', '2016-01-01'),
            ('JPM', '2017-01-01'),
            ('JPM', '2018-01-01'),
            ('BAC', '2015-01-01'),
            ('BAC', '2016-01-01'),
            ('BAC', '2017-01-01'),
            ('BAC', '2018-01-01'),
            ('WFC', '2015-01-01'),
            ('WFC', '2016-01-01'),
            ('WFC', '2017-01-01'),
            ('WFC', '2018-01-01'),
            ('USB', '2015-01-01'),
            ('USB', '2016-01-01'),
            ('USB', '2017-01-01'),
            ('USB', '2018-01-01'),
            ('TFC', '2015-01-01'),
            ('TFC', '2016-01-01'),
            ('TFC', '2017-01-01'),
            ('TFC', '2018-01-01')],
           names=['bank', 'year'])
```

In [6]:

```
data_dict = {}
data_dict["Data"] = data
data_dict["Diff Data"] = data.copy().loc[diff_index]
data_dict["Diff Data"] = data.groupby(level=0).diff(-1)
```

In [7]:

```
data_dict["Diff Data"]
```

Out[7]:

| bank | year | Bank ID | ROE | ROA | Solvency Ratio | Liquidity Ratio |
|---|---|---|---|---|---|---|
| JPM | **2015-01-01** | 0.0 | 0.0031 | 0.0000 | -0.30 | 0.01 |
| | **2016-01-01** | 0.0 | 0.0025 | 0.0004 | -0.11 | -0.01 |
| | **2017-01-01** | 0.0 | -0.0316 | -0.0030 | -0.31 | 0.02 |
| | **2018-01-01** | 0.0 | -0.0132 | -0.0009 | -0.06 | 0.08 |
| | **2019-01-01** | NaN | NaN | NaN | NaN | NaN |
| BAC | **2015-01-01** | 0.0 | 0.0031 | 0.0000 | -0.30 | 0.01 |
| | **2016-01-01** | 0.0 | 0.0025 | 0.0004 | -0.11 | -0.01 |
| | **2017-01-01** | 0.0 | -0.0316 | -0.0030 | -0.31 | 0.02 |
| | **2018-01-01** | 0.0 | -0.0127 | -0.0009 | -0.06 | 0.08 |
| | **2019-01-01** | NaN | NaN | NaN | NaN | NaN |
| WFC | **2015-01-01** | 0.0 | 0.0105 | 0.0016 | -0.41 | 0.01 |
| | **2016-01-01** | 0.0 | 0.0014 | 0.0001 | 0.25 | 0.04 |
| | **2017-01-01** | 0.0 | -0.0025 | -0.0003 | -0.24 | 0.00 |
| | **2018-01-01** | 0.0 | 0.0104 | 0.0016 | -0.63 | 0.01 |
| | **2019-01-01** | NaN | NaN | NaN | NaN | NaN |
| USB | **2015-01-01** | 0.0 | 0.0061 | 0.0008 | -0.29 | 0.00 |
| | **2016-01-01** | 0.0 | -0.0038 | -0.0002 | 0.00 | 0.01 |
| | **2017-01-01** | 0.0 | -0.0135 | -0.0017 | 0.25 | -0.03 |
| | **2018-01-01** | 0.0 | 0.0102 | 0.0010 | -0.39 | 0.02 |
| | **2019-01-01** | NaN | NaN | NaN | NaN | NaN |
| TFC | **2015-01-01** | 0.0 | -0.0025 | -0.0006 | 0.35 | 0.00 |
| | **2016-01-01** | 0.0 | 0.0026 | 0.0003 | -0.13 | 0.01 |
| | **2017-01-01** | 0.0 | -0.0284 | -0.0037 | -0.02 | -0.05 |
| | **2018-01-01** | 0.0 | 0.0274 | 0.0033 | 0.37 | -0.02 |
| | **2019-01-01** | NaN | NaN | NaN | NaN | NaN |

In [8]:
```
data_dict["Diff Data"] = data_dict["Diff Data"].dropna()
data_dict["Diff Data"]
```

Out[8]:

| | | Bank ID | ROE | ROA | Solvency Ratio | Liquidity Ratio |
|---|---|---|---|---|---|---|
| **bank** | **year** | | | | | |
| **JPM** | **2015-01-01** | 0.0 | 0.0031 | 0.0000 | -0.30 | 0.01 |
| | **2016-01-01** | 0.0 | 0.0025 | 0.0004 | -0.11 | -0.01 |
| | **2017-01-01** | 0.0 | -0.0316 | -0.0030 | -0.31 | 0.02 |
| | **2018-01-01** | 0.0 | -0.0132 | -0.0009 | -0.06 | 0.08 |
| **BAC** | **2015-01-01** | 0.0 | 0.0031 | 0.0000 | -0.30 | 0.01 |
| | **2016-01-01** | 0.0 | 0.0025 | 0.0004 | -0.11 | -0.01 |
| | **2017-01-01** | 0.0 | -0.0316 | -0.0030 | -0.31 | 0.02 |
| | **2018-01-01** | 0.0 | -0.0127 | -0.0009 | -0.06 | 0.08 |
| **WFC** | **2015-01-01** | 0.0 | 0.0105 | 0.0016 | -0.41 | 0.01 |
| | **2016-01-01** | 0.0 | 0.0014 | 0.0001 | 0.25 | 0.04 |
| | **2017-01-01** | 0.0 | -0.0025 | -0.0003 | -0.24 | 0.00 |
| | **2018-01-01** | 0.0 | 0.0104 | 0.0016 | -0.63 | 0.01 |
| **USB** | **2015-01-01** | 0.0 | 0.0061 | 0.0008 | -0.29 | 0.00 |
| | **2016-01-01** | 0.0 | -0.0038 | -0.0002 | 0.00 | 0.01 |
| | **2017-01-01** | 0.0 | -0.0135 | -0.0017 | 0.25 | -0.03 |
| | **2018-01-01** | 0.0 | 0.0102 | 0.0010 | -0.39 | 0.02 |
| **TFC** | **2015-01-01** | 0.0 | -0.0025 | -0.0006 | 0.35 | 0.00 |
| | **2016-01-01** | 0.0 | 0.0026 | 0.0003 | -0.13 | 0.01 |
| | **2017-01-01** | 0.0 | -0.0284 | -0.0037 | -0.02 | -0.05 |
| | **2018-01-01** | 0.0 | 0.0274 | 0.0033 | 0.37 | -0.02 |

In [13]:
```
data_diff = data_dict["Diff Data"]
```

In [15]:

```
from statsmodels.tsa.stattools import adfuller
X = data_diff["ROE"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")


X = data_diff["ROA"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")


X = data_diff["Solvency Ratio"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")


X = data_diff["Liquidity Ratio"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -1.275110
p-value: 0.640481
Critical Values:
	1%: -4.223
	5%: -3.189
	10%: -2.730
Failed to Reject Ho - Time Series is Non-Stationary
ADF Statistic: -1.472154
p-value: 0.547239
Critical Values:
	1%: -3.964
	5%: -3.085
	10%: -2.682
Failed to Reject Ho - Time Series is Non-Stationary
ADF Statistic: -4.577596
p-value: 0.000142
Critical Values:
	1%: -3.833
	5%: -3.031
	10%: -2.656
Reject Ho - Time Series is Stationary
ADF Statistic: 0.805256
p-value: 0.991721
Critical Values:
	1%: -4.223
	5%: -3.189
	10%: -2.730
Failed to Reject Ho - Time Series is Non-Stationary
```

In [18]:
```python
data_diff = data_diff.diff().dropna()
data_diff
```

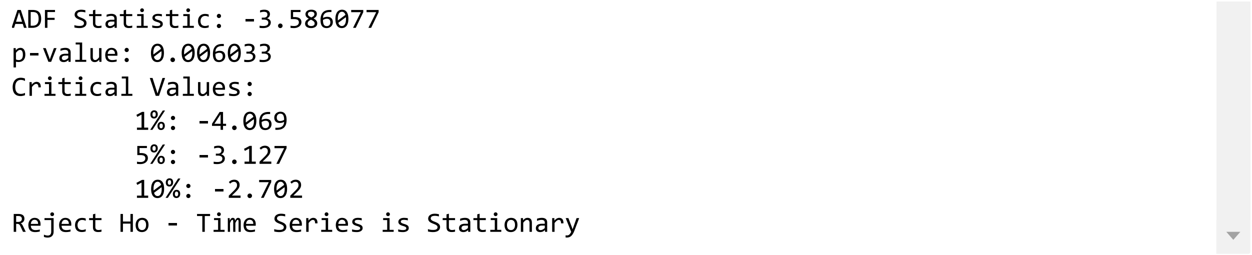
Out[18]:

|  |  | Bank ID | ROE | ROA | Solvency Ratio | Liquidity Ratio |
|---|---|---|---|---|---|---|
| **bank** | **year** |  |  |  |  |  |
| **JPM** | **2017-01-01** | 0.0 | -0.0335 | -0.0038 | -0.39 | 0.05 |
|  | **2018-01-01** | 0.0 | 0.0525 | 0.0055 | 0.45 | 0.03 |
| **BAC** | **2015-01-01** | 0.0 | -0.0021 | -0.0012 | -0.49 | -0.13 |
|  | **2016-01-01** | 0.0 | -0.0169 | -0.0005 | 0.43 | 0.05 |
|  | **2017-01-01** | 0.0 | -0.0335 | -0.0038 | -0.39 | 0.05 |
|  | **2018-01-01** | 0.0 | 0.0530 | 0.0055 | 0.45 | 0.03 |
| **WFC** | **2015-01-01** | 0.0 | 0.0043 | 0.0004 | -0.60 | -0.13 |
|  | **2016-01-01** | 0.0 | -0.0323 | -0.0040 | 1.01 | 0.10 |
|  | **2017-01-01** | 0.0 | 0.0052 | 0.0011 | -1.15 | -0.07 |
|  | **2018-01-01** | 0.0 | 0.0168 | 0.0023 | 0.10 | 0.05 |
| **USB** | **2015-01-01** | 0.0 | -0.0172 | -0.0027 | 0.73 | -0.02 |
|  | **2016-01-01** | 0.0 | -0.0056 | -0.0002 | -0.05 | 0.02 |
|  | **2017-01-01** | 0.0 | 0.0002 | -0.0005 | -0.04 | -0.05 |
|  | **2018-01-01** | 0.0 | 0.0334 | 0.0042 | -0.89 | 0.09 |
| **TFC** | **2015-01-01** | 0.0 | -0.0364 | -0.0043 | 1.38 | -0.07 |
|  | **2016-01-01** | 0.0 | 0.0178 | 0.0025 | -1.22 | 0.03 |
|  | **2017-01-01** | 0.0 | -0.0361 | -0.0049 | 0.59 | -0.07 |
|  | **2018-01-01** | 0.0 | 0.0868 | 0.0110 | 0.28 | 0.09 |

In [20]:

```python
from statsmodels.tsa.stattools import adfuller
X = data_diff["ROE"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")


X = data_diff["Solvency Ratio"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")


X = data_diff["Liquidity Ratio"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -3.296969
p-value: 0.015015
Critical Values:
        1%: -4.223
        5%: -3.189
        10%: -2.730
Reject Ho - Time Series is Stationary
ADF Statistic: -3.719737
p-value: 0.003844
Critical Values:
        1%: -4.138
        5%: -3.155
        10%: -2.714
Reject Ho - Time Series is Stationary
```



```
ADF Statistic: -3.586077
p-value: 0.006033
Critical Values:
        1%: -4.069
        5%: -3.127
        10%: -2.702
Reject Ho - Time Series is Stationary
```

In [9]:
```
from linearmodels import PanelOLS
# . . . .
y_name = ["ROE"]
X_names = ["Solvency Ratio",
          "Liquidity Ratio"]
for key, data in data_dict.items():
    for entity in [True, False]:
        for time in [True, False]:
            print(key)
            print("Entity =", entity)
            print("Time =", time)
            reg_data = data_dict[key].dropna()
            Y = reg_data[y_name]
            X = reg_data[X_names]
            X["Constant"] = 1
            # call panel_regression method
            model = PanelOLS(Y,X, entity_effects=entity, time_effects=time)
            results = model.fit(cov_type='clustered', cluster_entity=True)
            print(key, results, sep ="\n")
            reg_data["Predictor"] = results.predict()
            reg_data["Residuals"] = reg_data[y_name[0]].sub(reg_data["Predictor"]
```

```
Data
Entity = True
Time = True

C:\Users\HP\AppData\Local\Temp/ipykernel_8624/3485620510.py:15: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  X["Constant"] = 1

Data
                          PanelOLS Estimation Summary
===============================================================================
=
Dep. Variable:                    ROE   R-squared:                        0.316
8
Estimator:                   PanelOLS   R-squared (Between):             -2.859
3
No. Observations:                  25   R-squared (Within):               0.157
3
Date:                Wed, Apr 13 2022   R-squared (Overall):             -1.437
1
Time:                        20:58:17   Log-likelihood                    85.64
1
Cov. Estimator:             Clustered
                                        F-statistic:                      3.246
4
Entities:                           5   P-value                           0.069
5
Avg Obs:                       5.0000   Distribution:                    F(2,1
4)
Min Obs:                       5.0000
Max Obs:                       5.0000   F-statistic (robust):             116.9
8
                                        P-value                           0.000
0
Time periods:                       5   Distribution:                   F(2,1
4)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                             Parameter Estimates
================================================================================
====
                 Parameter  Std. Err.     T-stat    P-value    Lower CI    Uppe
r CI
--------------------------------------------------------------------------------
----
Solvency Ratio     -0.0101     0.0031    -3.2595     0.0057     -0.0168     -0.
0035
Liquidity Ratio    -0.2443     0.0950    -2.5717     0.0222     -0.4481     -0.
0406
Constant            0.4117     0.0647     6.3653     0.0000      0.2730      0.
5505
================================================================================
====

F-test for Poolability: 3.0244
P-value: 0.0338
Distribution: F(8,14)

Included effects: Entity, Time
Data
Entity = True
Time = False
Data
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_8624/3485620510.py:15: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  X["Constant"] = 1
C:\Users\HP\AppData\Local\Temp/ipykernel_8624/3485620510.py:15: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  X["Constant"] = 1
                          PanelOLS Estimation Summary
==============================================================================
===
Dep. Variable:                    ROE   R-squared:                        0.2
248
Estimator:                   PanelOLS   R-squared (Between):              0.4
365
No. Observations:                  25   R-squared (Within):               0.2
248
Date:                Wed, Apr 13 2022   R-squared (Overall):              0.3
367
Time:                        20:58:18   Log-likelihood                    76.
424
Cov. Estimator:             Clustered
                                        F-statistic:                      2.6
105
Entities:                           5   P-value                           0.1
010
Avg Obs:                       5.0000   Distribution:                    F(2,
18)
Min Obs:                       5.0000
Max Obs:                       5.0000   F-statistic (robust):             1.1
147
                                        P-value                           0.3
496
Time periods:                       5   Distribution:                    F(2,
18)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                             Parameter Estimates
==============================================================================
======
                 Parameter  Std. Err.     T-stat    P-value    Lower CI     Up
per CI
------------------------------------------------------------------------------
------
Solvency Ratio      0.0061     0.0134     0.4532     0.6558     -0.0221
0.0342
Liquidity Ratio    -0.1708     0.1234    -1.3840     0.1833     -0.4302
0.0885
Constant            0.2106     0.1644     1.2813     0.2164     -0.1347
0.5560
==============================================================================
======

F-test for Poolability: 1.3731
P-value: 0.2825
Distribution: F(4,18)

Included effects: Entity
Data
Entity = False
Time = True
```

```
Data
                          PanelOLS Estimation Summary
=============================================================================
===
Dep. Variable:                    ROE   R-squared:                        0.5
848
Estimator:                   PanelOLS   R-squared (Between):              0.7
999
No. Observations:                  25   R-squared (Within):               0.2
118
Date:                Wed, Apr 13 2022   R-squared (Overall):              0.5
227
Time:                        20:58:18   Log-likelihood                    77.
865
Cov. Estimator:             Clustered
                                        F-statistic:                      12.
679
Entities:                           5   P-value                           0.0
004
Avg Obs:                       5.0000   Distribution:                    F(2,
18)
Min Obs:                       5.0000
Max Obs:                       5.0000   F-statistic (robust):             12.
594
                                        P-value                           0.0
004
Time periods:                       5   Distribution:                    F(2,
18)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                                Parameter Estimates
=============================================================================
======
                 Parameter  Std. Err.     T-stat    P-value    Lower CI     Up
per CI
-----------------------------------------------------------------------------
------
Solvency Ratio      0.0131     0.0029     4.5388     0.0003      0.0070
0.0191
Liquidity Ratio    -0.1421     0.0564    -2.5187     0.0215     -0.2606     -
0.0236
Constant            0.1263     0.0377     3.3469     0.0036      0.0470
0.2056
=============================================================================
======

F-test for Poolability: 2.0906
P-value: 0.1243
Distribution: F(4,18)

Included effects: Time
Data
Entity = False
Time = False
Data
```

```
                          PanelOLS Estimation Summary
=============================================================================
===
Dep. Variable:                    ROE   R-squared:                        0.5
230
Estimator:                   PanelOLS   R-squared (Between):              0.8
020
No. Observations:                  25   R-squared (Within):               0.2
102
Date:                Wed, Apr 13 2022   R-squared (Overall):              0.5
230
Time:                        20:58:18   Log-likelihood                    73.
095
Cov. Estimator:             Clustered
                                        F-statistic:                      12.
063
Entities:                           5   P-value                           0.0
003
Avg Obs:                       5.0000   Distribution:                    F(2,
22)
Min Obs:                       5.0000
Max Obs:                       5.0000   F-statistic (robust):             10.
013
                                        P-value                           0.0
008
Time periods:                       5   Distribution:                    F(2,
22)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                                Parameter Estimates
=============================================================================
======
                 Parameter  Std. Err.     T-stat    P-value    Lower CI    Up
per CI
-----------------------------------------------------------------------------
------
Solvency Ratio      0.0135     0.0032     4.2362     0.0003      0.0069
0.0200
Liquidity Ratio    -0.1422     0.0521    -2.7279     0.0123     -0.2503     -
0.0341
Constant            0.1234     0.0325     3.7969     0.0010      0.0560
0.1909
=============================================================================
======


Diff Data
Entity = True
Time = True
Diff Data
                          PanelOLS Estimation Summary
=============================================================================
===
Dep. Variable:                    ROE   R-squared:                        0.4
865
```

```
Estimator:                   PanelOLS   R-squared (Between):              0.1
578
No. Observations:                  20   R-squared (Within):              -0.2
854
Date:                Wed, Apr 13 2022   R-squared (Overall):             -0.2
159
Time:                        20:58:18   Log-likelihood                    74.
537
Cov. Estimator:             Clustered
                                        F-statistic:                      4.7
378
Entities:                           5   P-value                           0.0
357
Avg Obs:                       4.0000   Distribution:                    F(2,
10)
Min Obs:                       4.0000
Max Obs:                       4.0000   F-statistic (robust):             60 
0.64
                                        P-value                           0.0
000
Time periods:                       4   Distribution:                    F(2,
10)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                                Parameter Estimates
=============================================================================
======
                 Parameter  Std. Err.     T-stat    P-value    Lower CI    Up
per CI
-----------------------------------------------------------------------------
------
Solvency Ratio      0.0056     0.0035     1.6145     0.1375     -0.0021      
0.0133
Liquidity Ratio    -0.2754     0.0338    -8.1416     0.0000     -0.3507     -
0.2000
Constant            0.0004     0.0001     3.1958     0.0096      0.0001      
0.0007
=============================================================================
======

F-test for Poolability: 7.5536
P-value: 0.0025
Distribution: F(7,10)

Included effects: Entity, Time
Diff Data
Entity = True
Time = False
Diff Data
                          PanelOLS Estimation Summary
=============================================================================
===
Dep. Variable:                    ROE   R-squared:                        0.0
003
Estimator:                   PanelOLS   R-squared (Between):             -0.0
```

```
227
No. Observations:                  20   R-squared (Within):               0.0
003
Date:                Wed, Apr 13 2022   R-squared (Overall):             -0.0
033
Time:                        20:58:18   Log-likelihood                    57.
733
Cov. Estimator:             Clustered
                                        F-statistic:                      0.0
019
Entities:                           5   P-value                           0.9
981
Avg Obs:                       4.0000   Distribution:                    F(2,
13)
Min Obs:                       4.0000
Max Obs:                       4.0000   F-statistic (robust):             0.0
022
                                        P-value                           0.9
978
Time periods:                       4   Distribution:                    F(2,
13)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                              Parameter Estimates
=============================================================================
======
                 Parameter  Std. Err.     T-stat    P-value    Lower CI    Up
per CI
-----------------------------------------------------------------------------
------
Solvency Ratio     -0.0010     0.0162    -0.0597     0.9533     -0.0359
0.0340
Liquidity Ratio     0.0045     0.1179     0.0382     0.9701     -0.2503
0.2593
Constant           -0.0032     0.0025    -1.2743     0.2249     -0.0085
0.0022
=============================================================================
======

F-test for Poolability: 0.5570
P-value: 0.6978
Distribution: F(4,13)

Included effects: Entity
Diff Data
Entity = False
Time = True
Diff Data
                          PanelOLS Estimation Summary
=============================================================================
===
Dep. Variable:                    ROE   R-squared:                        0.4
190
Estimator:                   PanelOLS   R-squared (Between):              0.3
586
```

```
No. Observations:                  20   R-squared (Within):              -0.2
408
Date:                Wed, Apr 13 2022   R-squared (Overall):             -0.1
468
Time:                        20:58:18   Log-likelihood                    69.
164
Cov. Estimator:             Clustered
                                        F-statistic:                      5.0
480
Entities:                           5   P-value                           0.0
224
Avg Obs:                       4.0000   Distribution:                    F(2,
14)
Min Obs:                       4.0000
Max Obs:                       4.0000   F-statistic (robust):             23.
121
                                        P-value                           0.0
000
Time periods:                       4   Distribution:                    F(2,
14)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                                Parameter Estimates
==============================================================================
======
                 Parameter  Std. Err.     T-stat    P-value    Lower CI     Up
per CI
------------------------------------------------------------------------------
------
Solvency Ratio     -0.0019     0.0006    -3.3627     0.0046     -0.0031      -
0.0007
Liquidity Ratio    -0.2504     0.0608    -4.1208     0.0010     -0.3808      -
0.1201
Constant           -0.0007     0.0020    -0.3567     0.7266     -0.0051
0.0036
==============================================================================
======

F-test for Poolability: 12.478
P-value: 0.0003
Distribution: F(3,14)

Included effects: Time
Diff Data
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_8624/3485620510.py:15: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
```

```
  X["Constant"] = 1
C:\Users\HP\AppData\Local\Temp/ipykernel_8624/3485620510.py:15: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  X["Constant"] = 1
```

```
Entity = False
Time = False
Diff Data
                          PanelOLS Estimation Summary
===============================================================================
=
Dep. Variable:                    ROE   R-squared:                        0.012
5
Estimator:                   PanelOLS   R-squared (Between):              0.145
1
No. Observations:                  20   R-squared (Within):              -0.012
1
Date:                Wed, Apr 13 2022   R-squared (Overall):              0.012
5
Time:                        20:58:18   Log-likelihood                    56.15
1
Cov. Estimator:             Clustered
                                        F-statistic:                      0.107
7
Entities:                           5   P-value                           0.898
5
Avg Obs:                       4.0000   Distribution:                    F(2,1
7)
Min Obs:                       4.0000
Max Obs:                       4.0000   F-statistic (robust):             0.174
7
                                        P-value                           0.841
2
Time periods:                       4   Distribution:                    F(2,1
7)
Avg Obs:                       5.0000
Min Obs:                       5.0000
Max Obs:                       5.0000

                                Parameter Estimates
===============================================================================
====
                 Parameter  Std. Err.     T-stat    P-value    Lower CI    Uppe
r CI
-------------------------------------------------------------------------------
----
Solvency Ratio      0.0004     0.0146     0.0261     0.9794     -0.0303      0.
0311
Liquidity Ratio    -0.0539     0.0937    -0.5747     0.5730     -0.2517      0.
1439
```

```
Constant             -0.0024      0.0016     -1.4672      0.1606     -0.0059      0.
0011
================================================================================
====
```

In [10]:

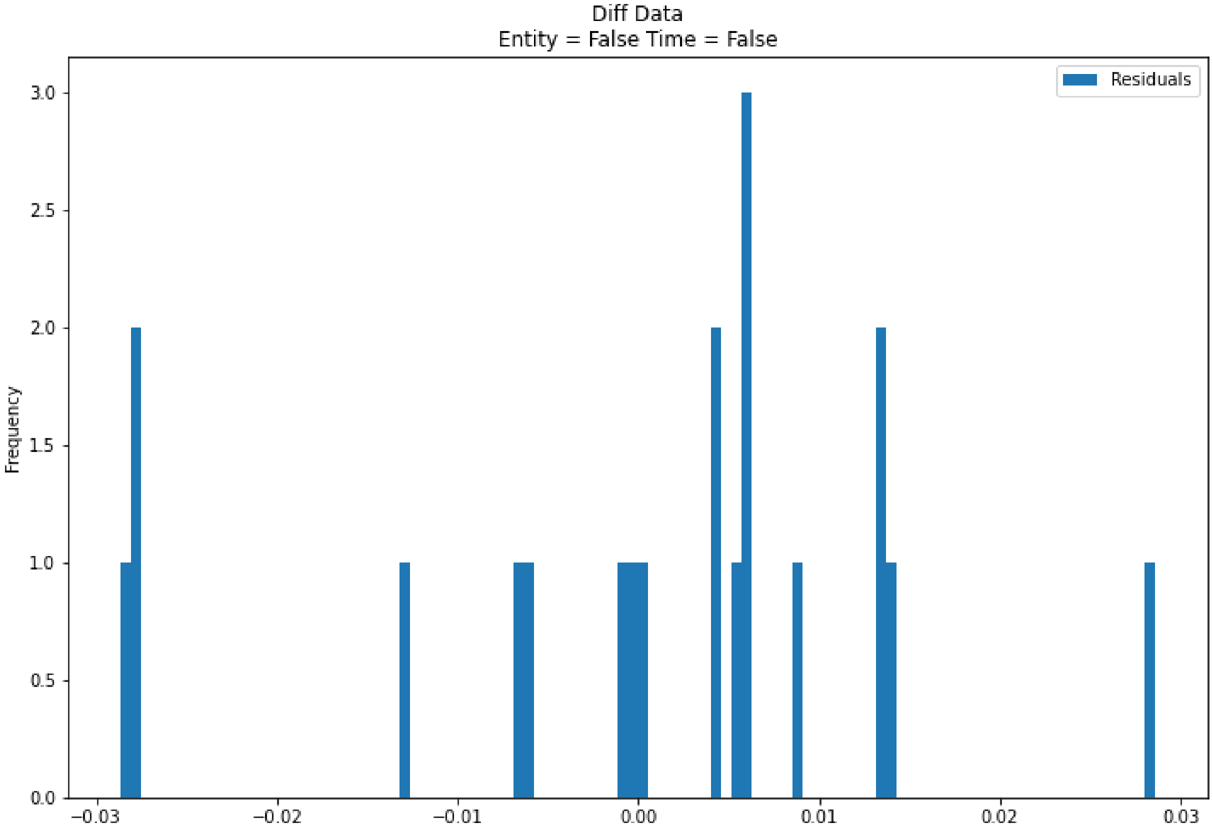
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [11]:

```
#plot distribution of residuals
fig, ax = plt.subplots(figsize = (12,8))
reg_data[["Residuals"]].plot.hist(bins = 100, ax = ax)
plt.title(key + "\nEntity = " + str(entity) + " Time = " + str(time) )
```

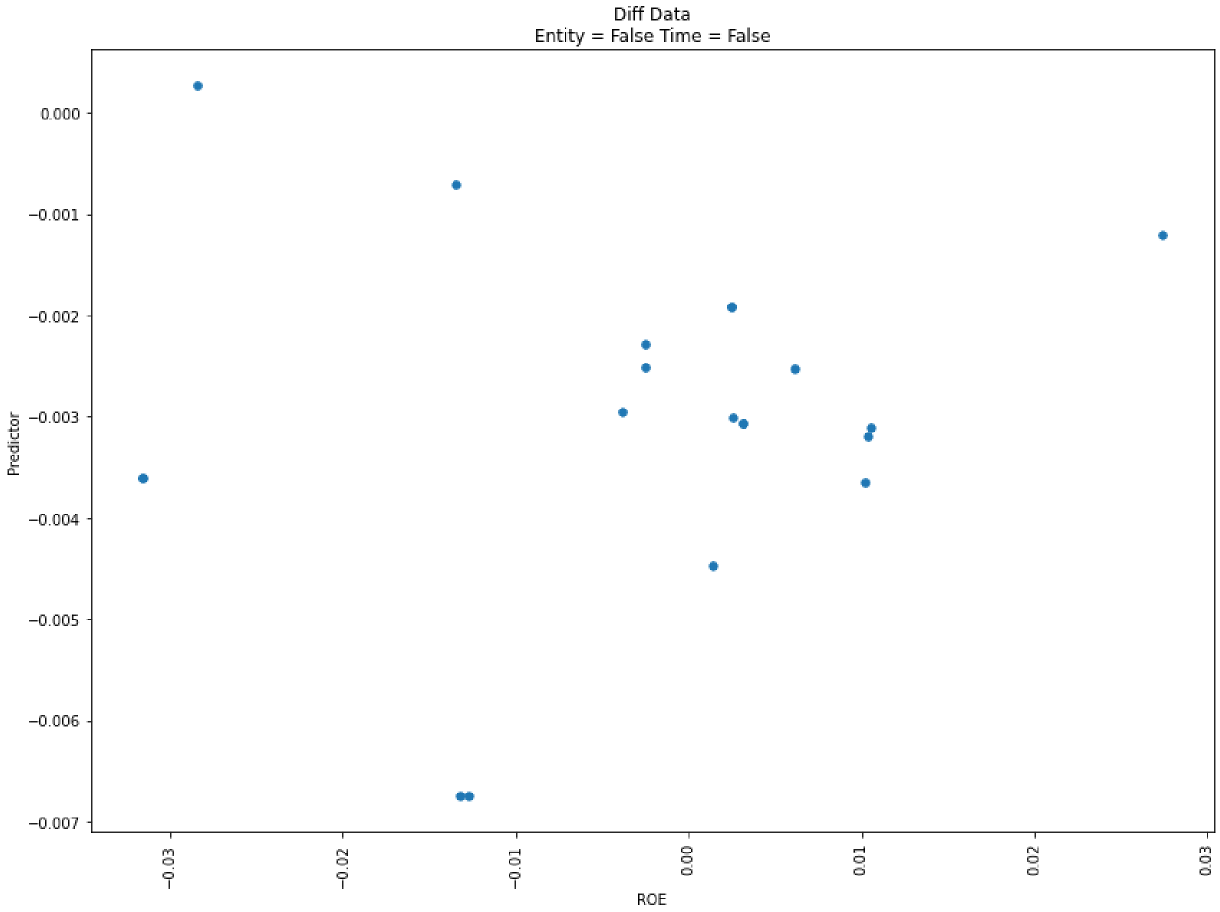
Out[11]:

```
Text(0.5, 1.0, 'Diff Data\nEntity = False Time = False')
```



In [12]:

```
 # plot observed vs. predicted values
fig, ax = plt.subplots(figsize = (14,10))
reg_data.plot.scatter(x = y_name[0],
                      y = "Predictor",
                      s = 30, ax = ax)
plt.xticks(rotation=90)
plt.title(key + "\nEntity = " + str(entity) + " Time = " + str(time) )
plt.show()
plt.close()
```


Diff Data
Entity = False Time = False
Predictor
ROE