In [ ]:

```python
import numpy as np
import pandas as pd
from statsmodels.tsa.stattools import grangercausalitytests
from statsmodels.tsa.stattools import adfuller
```

Ex. 1: Stationarity of wage growth and inflation data

Checking stationarity in a loop

In [ ]:

```python
#import data
df = pd.read_csv('Mehra.csv', index_col='date', parse_dates=True)

#parameter to check if all time series stationary
z = 1

#loop to difference
for i in range(0, 100):

    #check every time series
    for j in range(0, 8):

        #calculate adfuller to check for stationarity
        ad = adfuller(df.iloc[:, j])

        #if p is too high time series is non-stationary
        if(ad[1] > 0.05):
            z = 0
            print('Time series ' + str(j) + ' not stationary after ' + str(i) + ' differentials.')
            break

        #else keep going
        else:
            z += 1

    #if we exited with break one of time series wasn't statinary, need to difference again
    if(z == 0):
        df = df.diff().dropna()

    #else all time series stationary
    else:
        print('All time series stationary after ' + str(i) + " differentials.")
        break
```

```
Time series 0 not stationary after 0 differentials.
Time series 1 not stationary after 1 differentials.
All time series stationary after 2 differentials.
```

Ex. 2: Granger Causality Matrix

In [ ]:

```python
def GrangerMatrix(df, cols, lags):
    #matrix filled with zeroes for results
    res = pd.DataFrame(np.zeros((len(cols), len(cols))), columns = cols, index = cols)

    #loop for columns
    for i in cols:

        #loop for rows
        for j in cols:

            #run the test, verbose cancels the printing of results
            test = grangercausalitytests(df[[j, i]], maxlag = lags, verbose = False)

            #set minimum value to 1 at the start
            p_min = 1

            #finding the lowest value of p
            for k in range(lags):

                #if current p is lower than prevoius minimum, change it to new minimum
                if(test[k + 1][0]['ssr_chi2test'][1] < p_min):
                    p_min = test[k + 1][0]['ssr_chi2test'][1]

            #found the minimal p, set it to the field for corresponding time series in the matrix
            res.loc[j, i] = p_min

    #return the matrix
    return res
```

In [ ]:

```python
#create the matrix with dataframe from previous task - every time series already stationary
o = GrangerMatrix(df, df.columns, 12)

#print it
print(o)
```

```
           rgnp          pgnp           ulc     gdfco           gdf     gdfim  \
rgnp   1.000000  2.140627e-02  1.720399e-07  0.062738  2.357566e-04  0.018985
pgnp   0.004453  1.000000e+00  4.532711e-01  0.194061  1.783769e-02  0.046037
ulc    0.000003  6.286447e-04  1.000000e+00  0.550075  7.317498e-07  0.017228
gdfco  0.028141  5.764715e-02  2.664852e-04  1.000000  3.504453e-03  0.000029
gdf    0.207876  1.695280e-07  7.664548e-02  0.012704  1.000000e+00  0.000863
gdfim  0.095896  9.367638e-04  1.422892e-02  0.077287  1.429787e-04  1.000000
gdfcf  0.025423  1.990574e-07  1.148651e-01  0.809962  4.977663e-02  0.012006
gdfce  0.004623  1.089179e-01  9.617873e-04  0.082542  2.363979e-03  0.000466

          gdfcf     gdfce
rgnp   0.000071  0.020485
pgnp   0.046132  0.079810
ulc    0.000827  0.001268
gdfco  0.101157  0.001004
gdf    0.056880  0.000002
gdfim  0.000292  0.010624
gdfcf  1.000000  0.003076
gdfce  0.000018  1.000000
```