ENGLE- GRANGER APPROACH

Technique discussion, Drawbacks and future perspective.

Code Language: Python

IDE Plateform:
Jupyter Notebook

Tech Stacks and tools:

1) Numpy, Pandas.
2) Statsmodels.
3) Itertools, Matplotlib.

Encountered Problems:

1) Memory usage error in low(1 min) tick interval on higher sample.
2) Implemented functions are taking huge time especially in 1 min and 5 min tick interval.

Procedure:

Step 1: Long term co-integration

1. Check for price and change in price stationarity.
2. Estimate long term relationship: $y_t = \delta_0 + \delta_1 x_t + u_t$
3. Filter the pairs from above equation by ADF Test
4. Check for spurious regression: High t-statistic value, R-squared>Durbin Watson statistic.

Procedure:

Step 2: Estimate the Error Correction Model

1. Analyse the equation:

$$\Delta y_t = \phi_0 + \phi_1 \Delta y_{t-1} + \theta_1 \Delta x_{t-1} + \alpha(y_{t-1} - \hat{\delta}_0 - \hat{\delta}_1 x_{t-1}) + \varepsilon_t$$

2. Running the equation for every possible pair filtered in step 1

3. Select the best pair having most -ve alpha (-1<alpha<1).
As -ve alpha would tend to bring short term change to long term change.

4. Last term is difference in two consecutive price of independent

Trading steps:

1. Calculating the hedge ratio of the selected pair.
2. Calculating spread including hedge ratio and normalising the score.
3. Trading Algo: Sell at 1 close at 0, SL: 1.25+n_series[i]
   Buy at -1 close at 0, SL: n_series[i]-1.25
4. Calculated profits summary, Stop loss, Open and Closed position and visualising on line charts.

Drawbacks:

1. Independent and dependent coin selection is to be done by twice regression.
2. Stationarity is the main concern in all steps
3. Hedge Ratio is considered to be constant over the time(estimated by OLS).

Future Changes:

1. Hedge Ratio would be calculate by rolling regression technique.
2. Currently working by creating sample of 1 week (5 min interval) over 6 months.