FINANCIAL TRADING IN R

Introduction to Indicators

Why Use Indicators?

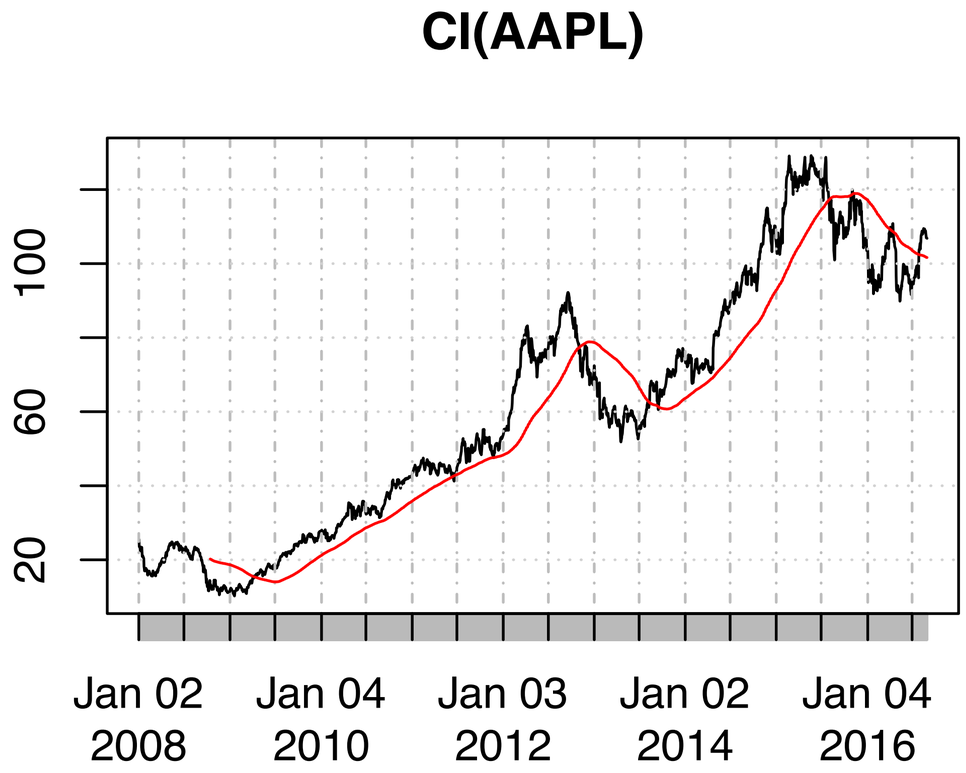
- Market data are exceptionally noisy
- In order to gain insights from market data, you need to transform the data through indicators

What Are Indicators?

- Indicators are transformations of market data
- Indicators gain smoothness and incur a lag penalty compared to raw market data
- Indicators can range from short term to very long term

Indicator examples

- *Trend* indicators: eg 200-day moving average



Indicator examples

- *Oscillation* indicators
  - Generate a signal of when it may be a good time to enter in short term position
  - often, scale of 0 to 100, -2 to 2,...
  - wait until price has pulled back with eye on future profit

In this class

- Combination of
  - basic moving average crossover
  - oscillation indicator

FINANCIAL TRADING IN R

Let’s practice!

FINANCIAL TRADING IN R

Indicator Mechanics

Five Steps to Calling Indicators

1. Write the add.indicator() function
2. Supply the strategy name (ex. strategy.st)
3. Name the function for calculating the indicator (ex. “SMA”)
4. Supply the inputs for the function as a list
5. Provide a label to your indicator (ex. “SMA200”)

Using add.indicator()

```
> # Call add.indicator() with strategy, name, arguments, and label
> add.indicator(strategy = strategy.st,
        name = 'SMA',
        arguments = list(x = quote(Cl(mktdata)), n = 200),
        label = 'SMA200'))
```

Another Way to Think About Indicators

- Applying an indicator is similar to using the apply() command in R
- You pass in the name of a function along with arguments
- The key difference is the addition of a label for your indicators

FINANCIAL TRADING IN R

Let’s practice!

FINANCIAL TRADING IN R

Indicator Structure Review

Review: Using add.indicator()

```
> add.indicator(strategy = strategy.st,
        name = 'SMA',
        arguments = list(x = quote(Cl(mktdata)), n = 200),
        label = 'SMA200'))
```

Naming Indicators

- Provide indicators with descriptive names
    - Ex. Name your 200 day simple moving average "SMA200", not just "SMA"
- Keep indicator names simple

applyIndicators

- creates intermediate data set containing market data and indicators

```
test <- applyIndicators(strategy = strategy,st, mktdata = OHLC(LQD))
head(test, n = 3)
           LQD.Open LQD.High  LQD.Low LQD.Close SMA.SMA200 SMA.SMA50 DVO.DVO_2_126
2003-01-02 58.37216 58.37216 57.32224  57.49366         NA         NA             NA
2003-01-03 57.63829 57.82042 57.45616  57.82042         NA         NA             NA
2003-01-06 57.71864 57.79363 57.39724  57.79363         NA         NA             NA

tail(test, n = 3)
           LQD.Open LQD.High  LQD.Low LQD.Close SMA.SMA200 SMA.SMA50 DVO.DVO_2_126
2015-12-23 113.9586 114.1979 113.8888   114.178   115.1378  115.0177     65.873016
2015-12-24 114.3400 114.5500 114.2000   114.550   115.1258  114.9885     92.857143
2015-12-28 114.3600 114.5600 114.2100   114.410   115.1147  114.9575     80.952381
```

- In quantstrat, indicator labels take the form of the original name, a dot and your label

Further Indicator Mechanics

- HLC() returns the high, low, and close as a time series (xts) object

```
> head(HLC(LQD))
           LQD.High  LQD.Low LQD.Close
2002-07-30 52.35639 51.97142  52.03302
2002-07-31 52.48472 52.12541  52.35126
2002-08-01 52.92102 52.51038  52.86456
2002-08-02 53.02368 52.58738  52.97235
2002-08-05 53.20334 52.61818  52.84402
2002-08-06 52.69004 52.40772  52.66437
```

Further Indicator Mechanics

- Use object[date/date] with HLC() to subset time series (xts) objects

```
> HLC(LQD["2012-01-01/2012-01-07"])
           LQD.High  LQD.Low LQD.Close
2012-01-03 97.05994 96.63424  96.77897
2012-01-04 97.01737 96.58316  96.85560
2012-01-05 96.85560 96.37881  96.43841
2012-01-06 96.90669 96.54058  96.81303
```

FINANCIAL TRADING IN R

Let’s practice!