In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import warnings
import pandas as pd
from yahoofinancials import YahooFinancials
```

In [2]:
```
#plt.style.use('seaborn')
#plt.rcParams['figure.figsize']=[16,9]
#plt.rcParams['figure.dpi']=300
#warnings.simplefilter(action='ignore',category=FutureWarning)
#cf.set_config_file(world_readable=True,theme='pearl',offline=True)
```

In [3]:
```
from datetime import datetime
import backtrader as bt

class MyBuySell(bt.observers.BuySell):
    plotlines = dict(
        buy=dict(marker='^', markersize=8.0, color='blue', fillstyle='full'),
        sell=dict(marker='v', markersize=8.0, color='red', fillstyle='full')
    )
```

In [4]:
```
# create the strategy using a signal
class SmaSignal(bt.Signal):
    params = (('period', 20), )

    def __init__(self):
        self.lines.signal = self.data - bt.ind.SMA(period=self.p.period)
```

In [5]:
```
# download data
data = bt.feeds.YahooFinanceData(
    dataname='AAPL',
    fromdate = datetime(2018,1,1),
    todate = datetime(201,12,31),
    buffered= True
    )
```

In [6]:
```
print(data)
```

```
<backtrader.feeds.yahoo.YahooFinanceData object at 0x0000028A4A6D6988>
```

In [7]:
```
# create a Cerebro entity
cerebro = bt.Cerebro(stdstats = False)
```

In [8]:
```
# set up the backtest
cerebro.adddata(data)
cerebro.broker.setcash(1000.0)
cerebro.add_signal(bt.SIGNAL_LONG, SmaSignal)
cerebro.addobserver(MyBuySell)
cerebro.addobserver(bt.observers.Value)
```

In [9]:
```
# run backtest
print(f'Starting Portfolio Value: {cerebro.broker.getvalue():.2f}')
x = cerebro.run()
print(f'Final Portfolio Value: {cerebro.broker.getvalue():.2f}')
```

```
Starting Portfolio Value: 1000.00

---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-9-ae86016c083f> in <module>
      1 # run backtest
      2 print(f'Starting Portfolio Value: {cerebro.broker.getvalue():.2f}')
----> 3 x = cerebro.run()
      4 print(f'Final Portfolio Value: {cerebro.broker.getvalue():.2f}')

C:\Python_3.7\lib\site-packages\backtrader\cerebro.py in run(self, **kwargs)
   1125             # let's skip process "spawning"
   1126             for iterstrat in iterstrats:
-> 1127                 runstrat = self.runstrategies(iterstrat)
   1128                 self.runstrats.append(runstrat)
   1129                 if self._dooptimize:

C:\Python_3.7\lib\site-packages\backtrader\cerebro.py in runstrategies(self,
 iterstrat, predata)
   1208                 if self._exactbars < 1:  # datas can be full length
   1209                     data.extend(size=self.params.lookahead)
-> 1210                 data._start()
   1211                 if self._dopreload:
   1212                     data.preload()

C:\Python_3.7\lib\site-packages\backtrader\feed.py in _start(self)
    201
    202     def _start(self):
--> 203         self.start()
    204
    205         if not self._started:

C:\Python_3.7\lib\site-packages\backtrader\feeds\yahoo.py in start(self)
    350
    351         # Prepared a "path" file -  CSV Parser can take over
--> 352         super(YahooFinanceData, self).start()
    353
    354

C:\Python_3.7\lib\site-packages\backtrader\feeds\yahoo.py in start(self)
     92
     93     def start(self):
---> 94         super(YahooFinanceCSVData, self).start()
     95
     96         if not self.params.reverse:

C:\Python_3.7\lib\site-packages\backtrader\feed.py in start(self)
    672             else:
    673                 # Let an exception propagate to let the caller know
--> 674                 self.f = io.open(self.p.dataname, 'r')
    675
    676         if self.p.headers:

FileNotFoundError: [Errno 2] No such file or directory: 'AAPL'
```

In [ ]:
```
# plot results
cerebro.plot(iplot=True, volume=False)
```

In [ ]:
```
```