Exploiting Wall Street

An algorithmic approach to investing
Goldman Sachs Fired 99% of Traders And Replaced Them With Robots
What disadvantages do you have as an investor?
Beating the market: Everybody tries to do beat it, but few succeed.

“Yes, you may be able to beat the market, but with investment fees, taxes, and human emotion working against you, you're more likely to do so through luck than skill. If you can merely **match** the S&P 500, minus a small fee, you'll be doing better than most investors.”

-investopedia
Algorithmic vs Buy and Hold
S&P 500

- Benchmark
- Algorithm Returns

Year

Value:
- $300,000
- $200,000
- $100,000
- $0
Algorithmic vs Buy and Hold

S&P 500

Year

Benchmark

Algorithm Returns
The Algorithm
elif (signal['action'] == "sell" and positionPrice > 0.0):
    cash += todayCandle['close']*positionSize
    positionSize = 0
    positionPrice = 0.0
elif (lastTradePrice < todayCandle['close']):
    winningTrades += 1
else:
    losingTrades += 1
lastTradePrice = todayCandle['close']

elif (signal['action'] == "buy" and positionPrice == 0.0):
    positionPrice = todayCandle['close']
    positionSize = int(cash/positionPrice)
    cash -= positionSize*positionPrice
    if (lastTradePrice > todayCandle['close']):
        winningTrades += 1
    else:
        losingTrades += 1
lastTradePrice = todayCandle['close']
```python
class Event(object):
    pass

class MarketEvent(Event):
    ''' Handles the event of receiving a new market order update with corresponding bars '''
    def __init__(self):
        self.type = 'MARKET'

class SignalEvent(Event):
    def __init__(self, symbol, datetime, signalType, signalStrength):
        self.type = 'SIGNAL'
        self.symbol = symbol
        self.datetime = datetime
        self.signalType = signalType
        self.signalStrength = signalStrength

class OrderEvent(Event):
    def __init__(self, symbol, orderType, quantity, direction, stop=None, stop_type=None):
        self.type = 'ORDER'
        self.symbol = symbol
        self.orderType = orderType
        self.quantity = quantity
        self.direction = direction
        self.stop = stop
        self.stop_type = stop_type

    def printOrder(self):
        print("Order : Symbol= %s, Type= %s, Quantity= %s, Direction= %s" % (self.symbol, self.orderType, self.quantity, self.direction))

class FillEvent(Event):
    def __init__(self, timeindex, symbol, exchange, quantity, direction, fill_cost, commission=None):
        self.type = 'FILL'
        self.timeindex = timeindex
        self.symbol = symbol
        self.exchange = exchange
        self.quantity = quantity
        self.direction = direction
        self.fill_cost = fill_cost  # holding value in dollars
        self.commission = commission  # should never be anything if using alpaca
```

The Road Ahead
Questions?
Appendix
<table>
<thead>
<tr>
<th>Year</th>
<th>Benchmark</th>
<th>Algorithm Returns</th>
<th>Algorithm Returns A</th>
<th>Benchmark Returns</th>
<th>AlgoReturns</th>
<th>Algo vs Benchmark</th>
<th>Winning Trades</th>
<th>Losing Trades</th>
<th>Total Trades</th>
<th>Max DrawDown</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>$24,421.50</td>
<td>$29,691.09</td>
<td>$28,752.88</td>
<td>-2.31%</td>
<td>18.76%</td>
<td>21.08%</td>
<td>50</td>
<td>42</td>
<td>92</td>
<td>-2.55%</td>
<td>54.35%</td>
</tr>
<tr>
<td>1995</td>
<td>$33,631.91</td>
<td>$31,130.68</td>
<td>$29,904.54</td>
<td>34.53%</td>
<td>24.52%</td>
<td>-10.00%</td>
<td>49</td>
<td>57</td>
<td>106</td>
<td>-0.85%</td>
<td>46.23%</td>
</tr>
<tr>
<td>1996</td>
<td>$30,054.38</td>
<td>$33,398.58</td>
<td>$31,718.86</td>
<td>20.22%</td>
<td>33.59%</td>
<td>13.38%</td>
<td>51</td>
<td>51</td>
<td>102</td>
<td>-3.04%</td>
<td>50.00%</td>
</tr>
<tr>
<td>1997</td>
<td>$32,613.00</td>
<td>$36,533.01</td>
<td>$34,226.41</td>
<td>30.45%</td>
<td>46.13%</td>
<td>15.68%</td>
<td>52</td>
<td>52</td>
<td>104</td>
<td>-3.56%</td>
<td>50.00%</td>
</tr>
<tr>
<td>1998</td>
<td>$31,568.00</td>
<td>$32,654.69</td>
<td>$31,123.75</td>
<td>26.27%</td>
<td>30.62%</td>
<td>4.35%</td>
<td>52</td>
<td>66</td>
<td>118</td>
<td>-5.02%</td>
<td>44.07%</td>
</tr>
<tr>
<td>1999</td>
<td>$29,668.75</td>
<td>$36,521.03</td>
<td>$34,216.82</td>
<td>18.68%</td>
<td>46.08%</td>
<td>27.41%</td>
<td>55</td>
<td>48</td>
<td>103</td>
<td>-2.85%</td>
<td>53.40%</td>
</tr>
<tr>
<td>2000</td>
<td>$22,039.50</td>
<td>$33,549.30</td>
<td>$31,893.44</td>
<td>-11.84%</td>
<td>34.20%</td>
<td>46.04%</td>
<td>54</td>
<td>50</td>
<td>104</td>
<td>-3.53%</td>
<td>51.92%</td>
</tr>
<tr>
<td>2001</td>
<td>$21,602.70</td>
<td>$29,570.68</td>
<td>$28,656.55</td>
<td>-13.59%</td>
<td>18.28%</td>
<td>31.87%</td>
<td>51</td>
<td>57</td>
<td>108</td>
<td>-6.76%</td>
<td>47.22%</td>
</tr>
<tr>
<td>2002</td>
<td>$19,145.91</td>
<td>$30,834.24</td>
<td>$29,667.39</td>
<td>-23.42%</td>
<td>23.34%</td>
<td>48.75%</td>
<td>62</td>
<td>57</td>
<td>119</td>
<td>-5.04%</td>
<td>52.10%</td>
</tr>
<tr>
<td>2003</td>
<td>$31,269.68</td>
<td>$30,096.47</td>
<td>$31,477.17</td>
<td>25.08%</td>
<td>32.39%</td>
<td>7.31%</td>
<td>53</td>
<td>42</td>
<td>95</td>
<td>-2.67%</td>
<td>55.79%</td>
</tr>
<tr>
<td>2004</td>
<td>$26,954.01</td>
<td>$30,057.79</td>
<td>$29,046.23</td>
<td>7.82%</td>
<td>20.23%</td>
<td>12.42%</td>
<td>61</td>
<td>49</td>
<td>110</td>
<td>-1.19%</td>
<td>55.45%</td>
</tr>
<tr>
<td>2005</td>
<td>$25,524.55</td>
<td>$30,002.17</td>
<td>$29,001.73</td>
<td>2.10%</td>
<td>19.91%</td>
<td>5.40%</td>
<td>54</td>
<td>50</td>
<td>104</td>
<td>-1.84%</td>
<td>51.92%</td>
</tr>
<tr>
<td>2006</td>
<td>$28,182.38</td>
<td>$31,603.53</td>
<td>$30,282.82</td>
<td>12.73%</td>
<td>26.41%</td>
<td>13.68%</td>
<td>52</td>
<td>34</td>
<td>86</td>
<td>-3.01%</td>
<td>60.47%</td>
</tr>
<tr>
<td>2007</td>
<td>$25,586.75</td>
<td>$28,073.50</td>
<td>$27,458.80</td>
<td>2.35%</td>
<td>12.29%</td>
<td>9.95%</td>
<td>53</td>
<td>61</td>
<td>114</td>
<td>-3.41%</td>
<td>46.49%</td>
</tr>
<tr>
<td>2008</td>
<td>$15,340.80</td>
<td>$25,457.67</td>
<td>$25,366.13</td>
<td>-38.64%</td>
<td>1.83%</td>
<td>40.47%</td>
<td>52</td>
<td>63</td>
<td>115</td>
<td>-4.48%</td>
<td>45.22%</td>
</tr>
<tr>
<td>2009</td>
<td>$30,757.44</td>
<td>$39,559.73</td>
<td>$36,647.78</td>
<td>23.03%</td>
<td>58.24%</td>
<td>35.21%</td>
<td>50</td>
<td>56</td>
<td>106</td>
<td>-5.02%</td>
<td>47.17%</td>
</tr>
<tr>
<td>2010</td>
<td>$27,916.50</td>
<td>$30,502.17</td>
<td>$29,401.73</td>
<td>11.67%</td>
<td>22.01%</td>
<td>10.34%</td>
<td>45</td>
<td>51</td>
<td>96</td>
<td>-3.27%</td>
<td>46.88%</td>
</tr>
<tr>
<td>2011</td>
<td>$24,723.50</td>
<td>$24,176.64</td>
<td>$24,341.31</td>
<td>-1.11%</td>
<td>-3.29%</td>
<td>-2.19%</td>
<td>48</td>
<td>64</td>
<td>112</td>
<td>-7.79%</td>
<td>42.86%</td>
</tr>
<tr>
<td>2012</td>
<td>$27,769.95</td>
<td>$29,572.61</td>
<td>$28,658.09</td>
<td>-11.08%</td>
<td>18.29%</td>
<td>7.21%</td>
<td>48</td>
<td>59</td>
<td>107</td>
<td>-1.35%</td>
<td>44.86%</td>
</tr>
<tr>
<td>2013</td>
<td>$31,768.68</td>
<td>$29,402.36</td>
<td>$28,521.89</td>
<td>27.07%</td>
<td>17.61%</td>
<td>-9.46%</td>
<td>43</td>
<td>76</td>
<td>119</td>
<td>-2.11%</td>
<td>36.13%</td>
</tr>
<tr>
<td>2014</td>
<td>$27,747.90</td>
<td>$30,402.88</td>
<td>$29,322.31</td>
<td>10.99%</td>
<td>21.61%</td>
<td>10.62%</td>
<td>56</td>
<td>48</td>
<td>104</td>
<td>-2.30%</td>
<td>53.85%</td>
</tr>
<tr>
<td>2015</td>
<td>$24,668.27</td>
<td>$31,163.54</td>
<td>$29,930.83</td>
<td>-1.33%</td>
<td>24.65%</td>
<td>25.98%</td>
<td>60</td>
<td>48</td>
<td>108</td>
<td>-3.17%</td>
<td>55.56%</td>
</tr>
<tr>
<td>2016</td>
<td>$27,717.72</td>
<td>$27,415.24</td>
<td>$26,932.20</td>
<td>10.87%</td>
<td>9.66%</td>
<td>-1.21%</td>
<td>44</td>
<td>62</td>
<td>106</td>
<td>-3.65%</td>
<td>41.51%</td>
</tr>
<tr>
<td>2017</td>
<td>$29,621.46</td>
<td>$27,580.33</td>
<td>$27,064.27</td>
<td>18.49%</td>
<td>10.32%</td>
<td>-8.16%</td>
<td>37</td>
<td>63</td>
<td>100</td>
<td>-1.05%</td>
<td>37.00%</td>
</tr>
<tr>
<td>2018</td>
<td>$23,040.75</td>
<td>$28,019.26</td>
<td>$27,415.41</td>
<td>-7.84%</td>
<td>12.08%</td>
<td>19.91%</td>
<td>51</td>
<td>45</td>
<td>96</td>
<td>-4.27%</td>
<td>53.13%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>48,334.00</strong></td>
<td><strong>144,969.19</strong></td>
<td><strong>115,975.35</strong></td>
<td><strong>193.34%</strong></td>
<td><strong>579.88%</strong></td>
<td><strong>386.54%</strong></td>
<td><strong>1283</strong></td>
<td><strong>1351</strong></td>
<td><strong>2634</strong></td>
<td><strong>-</strong></td>
<td><strong>-</strong></td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>26,933.36</strong></td>
<td><strong>30,798.77</strong></td>
<td><strong>29,639.01</strong></td>
<td><strong>7.73%</strong></td>
<td><strong>23.20%</strong></td>
<td><strong>15.46%</strong></td>
<td><strong>51.32</strong></td>
<td><strong>54.04</strong></td>
<td><strong>105.36</strong></td>
<td><strong>-3.35%</strong></td>
<td><strong>48.94%</strong></td>
</tr>
<tr>
<td><strong>Std Dev</strong></td>
<td><strong>4,411.27</strong></td>
<td><strong>3,435.58</strong></td>
<td><strong>2,748.46</strong></td>
<td><strong>17.65%</strong></td>
<td><strong>13.74%</strong></td>
<td><strong>16.18%</strong></td>
<td><strong>5.55</strong></td>
<td><strong>9.16</strong></td>
<td><strong>8.35</strong></td>
<td><strong>1.69%</strong></td>
<td><strong>6.00%</strong></td>
</tr>
<tr>
<td>Year</td>
<td>Benchmark</td>
<td>Algorithm Returns</td>
<td>Returns After Tax</td>
<td>Benchmark Returns</td>
<td>AlgoReturns</td>
<td>Algo vs Benchmark</td>
<td>Winning Trades</td>
<td>Losing Trades</td>
<td>Total Trades</td>
<td>Max DrawDown</td>
<td>Accuracy</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>1994</td>
<td>$24,422</td>
<td>$29,691</td>
<td>$28,752.88</td>
<td>-2.31%</td>
<td>18.76%</td>
<td>21.08%</td>
<td>50</td>
<td>42</td>
<td>92</td>
<td>-2.55%</td>
<td>54.35%</td>
</tr>
<tr>
<td>1995</td>
<td>$32,956</td>
<td>$37,214</td>
<td>$34,771</td>
<td>34.94%</td>
<td>25.34%</td>
<td>-9.61%</td>
<td>100</td>
<td>98</td>
<td>198</td>
<td>-2.55%</td>
<td>50.51%</td>
</tr>
<tr>
<td>1996</td>
<td>$39,580</td>
<td>$49,714</td>
<td>$44,771</td>
<td>20.10%</td>
<td>33.59%</td>
<td>13.49%</td>
<td>151</td>
<td>149</td>
<td>300</td>
<td>-3.04%</td>
<td>50.33%</td>
</tr>
<tr>
<td>1997</td>
<td>$52,026</td>
<td>$73,365</td>
<td>$63,692</td>
<td>31.44%</td>
<td>47.57%</td>
<td>16.13%</td>
<td>205</td>
<td>199</td>
<td>404</td>
<td>-3.55%</td>
<td>50.74%</td>
</tr>
<tr>
<td>1998</td>
<td>$66,096</td>
<td>$95,847</td>
<td>$81,678</td>
<td>27.04%</td>
<td>30.64%</td>
<td>3.60%</td>
<td>257</td>
<td>265</td>
<td>522</td>
<td>-5.03%</td>
<td>49.23%</td>
</tr>
<tr>
<td>1999</td>
<td>$78,725</td>
<td>$140,085</td>
<td>$117,068</td>
<td>19.11%</td>
<td>46.15%</td>
<td>27.05%</td>
<td>312</td>
<td>313</td>
<td>625</td>
<td>-5.03%</td>
<td>49.92%</td>
</tr>
<tr>
<td>2000</td>
<td>$70,317</td>
<td>$186,166</td>
<td>$153,933</td>
<td>-10.68%</td>
<td>32.89%</td>
<td>43.58%</td>
<td>367</td>
<td>363</td>
<td>730</td>
<td>-5.03%</td>
<td>50.27%</td>
</tr>
<tr>
<td>2001</td>
<td>$61,265</td>
<td>$220,287</td>
<td>$181,214</td>
<td>-12.87%</td>
<td>18.32%</td>
<td>31.19%</td>
<td>419</td>
<td>419</td>
<td>836</td>
<td>-6.76%</td>
<td>50.00%</td>
</tr>
<tr>
<td>2002</td>
<td>$47,291</td>
<td>$275,841</td>
<td>$225,673</td>
<td>-22.81%</td>
<td>25.33%</td>
<td>48.04%</td>
<td>483</td>
<td>474</td>
<td>957</td>
<td>-6.76%</td>
<td>50.47%</td>
</tr>
<tr>
<td>2003</td>
<td>$59,646</td>
<td>$378,925</td>
<td>$308,140</td>
<td>26.12%</td>
<td>37.37%</td>
<td>11.25%</td>
<td>536</td>
<td>515</td>
<td>1051</td>
<td>-6.76%</td>
<td>51.00%</td>
</tr>
<tr>
<td>2004</td>
<td>$64,786</td>
<td>$455,463</td>
<td>$369,371</td>
<td>8.62%</td>
<td>20.20%</td>
<td>11.58%</td>
<td>597</td>
<td>565</td>
<td>1162</td>
<td>-6.76%</td>
<td>51.38%</td>
</tr>
<tr>
<td>2005</td>
<td>$66,737</td>
<td>$546,910</td>
<td>$442,528</td>
<td>3.01%</td>
<td>20.08%</td>
<td>17.07%</td>
<td>652</td>
<td>614</td>
<td>1266</td>
<td>-6.76%</td>
<td>51.50%</td>
</tr>
<tr>
<td>2006</td>
<td>$75,908</td>
<td>$700,103</td>
<td>$565,082</td>
<td>13.74%</td>
<td>28.01%</td>
<td>14.27%</td>
<td>704</td>
<td>648</td>
<td>1352</td>
<td>-6.76%</td>
<td>52.07%</td>
</tr>
<tr>
<td>2007</td>
<td>$78,369</td>
<td>$786,689</td>
<td>$634,351</td>
<td>3.24%</td>
<td>12.37%</td>
<td>9.13%</td>
<td>758</td>
<td>708</td>
<td>1466</td>
<td>-6.76%</td>
<td>51.71%</td>
</tr>
<tr>
<td>2008</td>
<td>$48,369</td>
<td>$800,811</td>
<td>$645,649</td>
<td>-38.28%</td>
<td>1.00%</td>
<td>40.08%</td>
<td>811</td>
<td>770</td>
<td>1581</td>
<td>-6.76%</td>
<td>51.30%</td>
</tr>
<tr>
<td>2009</td>
<td>$59,732</td>
<td>$1,324,473</td>
<td>$1,064,578</td>
<td>23.49%</td>
<td>65.39%</td>
<td>41.90%</td>
<td>862</td>
<td>824</td>
<td>1686</td>
<td>-6.76%</td>
<td>51.13%</td>
</tr>
<tr>
<td>2010</td>
<td>$67,402</td>
<td>$1,630,391</td>
<td>$1,309,313</td>
<td>12.84%</td>
<td>23.10%</td>
<td>10.26%</td>
<td>908</td>
<td>874</td>
<td>1782</td>
<td>-6.76%</td>
<td>50.95%</td>
</tr>
<tr>
<td>2011</td>
<td>$67,268</td>
<td>$1,576,460</td>
<td>$1,266,168</td>
<td>-0.20%</td>
<td>-3.31%</td>
<td>-3.11%</td>
<td>956</td>
<td>938</td>
<td>1894</td>
<td>-7.80%</td>
<td>50.48%</td>
</tr>
<tr>
<td>2012</td>
<td>$76,332</td>
<td>$1,865,715</td>
<td>$1,497,572</td>
<td>13.47%</td>
<td>18.35%</td>
<td>4.87%</td>
<td>1004</td>
<td>997</td>
<td>2001</td>
<td>-7.80%</td>
<td>50.17%</td>
</tr>
<tr>
<td>2013</td>
<td>$98,994</td>
<td>$2,261,516</td>
<td>$1,814,213</td>
<td>29.69%</td>
<td>21.21%</td>
<td>-8.47%</td>
<td>1048</td>
<td>1071</td>
<td>2119</td>
<td>-7.80%</td>
<td>49.46%</td>
</tr>
<tr>
<td>2014</td>
<td>$110,169</td>
<td>$2,742,438</td>
<td>$2,198,950</td>
<td>11.29%</td>
<td>21.27%</td>
<td>9.98%</td>
<td>1105</td>
<td>1119</td>
<td>2224</td>
<td>-7.80%</td>
<td>49.69%</td>
</tr>
<tr>
<td>2015</td>
<td>$109,274</td>
<td>$3,343,912</td>
<td>$2,680,129</td>
<td>-0.81%</td>
<td>21.93%</td>
<td>22.74%</td>
<td>1167</td>
<td>1167</td>
<td>2334</td>
<td>-7.80%</td>
<td>50.00%</td>
</tr>
<tr>
<td>2016</td>
<td>$119,812</td>
<td>$3,669,460</td>
<td>$2,940,568</td>
<td>9.64%</td>
<td>9.74%</td>
<td>0.09%</td>
<td>1212</td>
<td>1228</td>
<td>2440</td>
<td>-7.80%</td>
<td>49.67%</td>
</tr>
<tr>
<td>2017</td>
<td>$143,037</td>
<td>$4,077,434</td>
<td>$3,266,947</td>
<td>19.38%</td>
<td>11.12%</td>
<td>-8.27%</td>
<td>1250</td>
<td>1290</td>
<td>2540</td>
<td>-7.80%</td>
<td>49.21%</td>
</tr>
<tr>
<td>2018</td>
<td>$132,794</td>
<td>$4,584,300</td>
<td>$3,672,440</td>
<td>-7.16%</td>
<td>12.43%</td>
<td>19.59%</td>
<td>1301</td>
<td>1335</td>
<td>2636</td>
<td>-7.80%</td>
<td>49.36%</td>
</tr>
</tbody>
</table>

**TOTAL** | **212.06%** | **599.55%** |

**AVERAGE** | **16.31%** | **46.12%** |

**STD DEV** | **17.77%** | **14.71%** |
<table>
<thead>
<tr>
<th>open</th>
<th>168.368</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>168.694</td>
</tr>
<tr>
<td>low</td>
<td>166.611</td>
</tr>
<tr>
<td>close</td>
<td>166.956</td>
</tr>
<tr>
<td>volume</td>
<td>1.40803e+08</td>
</tr>
<tr>
<td>unadjustedClose</td>
<td>None</td>
</tr>
<tr>
<td>unadjustedVolume</td>
<td>1.40803e+08</td>
</tr>
<tr>
<td>changed</td>
<td>-1.8657</td>
</tr>
<tr>
<td>changePercent</td>
<td>-1.105</td>
</tr>
<tr>
<td>vwap</td>
<td>167.536</td>
</tr>
<tr>
<td>label</td>
<td>Apr 7, 14</td>
</tr>
<tr>
<td>changeOverTime</td>
<td>0</td>
</tr>
<tr>
<td>Name: 2014-04-07, dtype: object</td>
<td></td>
</tr>
</tbody>
</table>

0
Okay so this is where it gets tricky. The method update_timeindex handles the new holdings tracking. First it obtains the latest prices from the market data handler and creates a new dictionary of symbols to represent the current_positions these are changed when a FillEvent is given. Which is handled in a later function. The method appends this set of current_positions to the all_positions list. Next the holdings are updated in a similar manner, with the exemption that the market_value is recalculated by multiplying the current_positions count with the closing price of the last bar.

ie: self.current_positions[s] * bars[s][0][5]

then the new holdings are appended to all_holdings

def update_timeindex(self, event):
    # Adds a new record to the positions matrix/table for the current market
    # data bar. This rejects the previous bar.
    # This means all current market data is "Known"
    # This is the function that actually uses the MarketEvent from the queue
    print(\"inside Portfolio.update_timeindex\")
    bars = {}
    for sym in self.symbollist:
        bars[sym] = self.bars.get_latest_bars(sym, N=1)

    # Update positions
    index = 0
    dp = dict((x,y) for x, y in [(s, 0) for s in self.symbollist])
    dp[\'date\'] = bars[0][0][1]
    for x in self.symbollist:
        dp[x] = self.current_positions[x]

    # Append current positions
    self.all_positions.append(dp)

    # Update holdings:
    dh = dict((x,y) for x, y in [(s, 0) for s in self.symbollist])
    dh[\'date\'] = bars[0][0][1]
    dh[\'cash\'] = float(self.current_holding[\'cash\'])
    dh[\'total\'] = float(self.current_holding[\'cash\'])
    for s in self.symbollist:
        # the real value of portfolio.
        market_value = float(float(self.current_positions[s]) * float(bars[s][0][5]))
        print(market_value)
        dh[s] = market_value
        dh[\'total\'] += market_value
    self.all_holdings.append(dh)
from DataHandler import HistoricSQLDataHandler
from ExecutionDriver import SimulatedExecutionHandler
import PortfolioDriver
import Strategy
import sys
is_py2 = sys.version[0] == '2'
if is_py2:
    import Queue as queue
else:
    from multiprocessing import Queue
import time
import queue

events = queue.Queue()
symbolList = []
symbolList.append('SPY_Data')
bars = HistoricSQLDataHandler(events, 'SELECT * FROM', symbolList)
strategy = Strategy.StochasticBands(bars, 4, 2)
broker = SimulatedExecutionHandler(events)

while True:
    if bars.continueBacktest:
        bars.updateBars()
    else:
        break
    while True:
        try:
            event = events.get(False)
        except queue.Empty:
            print('The Queue Is Currently Empty')
            break
        else:
            if event is not None:
                print(event)
                if event.type == 'MARKET':
                    print('MARKET EVENT RECEIVED')
                    print(event)
                    strategy.calculate_signals(event)
                    port.update_timeIndex(event)
                elif event.type == 'SIGNAL':
                    print('SIGNAL EVENT RECEIVED')
                    input('Press Enter To Continue')
                    port.updateSignal(event)
                elif event.type == 'ORDER':
                    print('ORDER EVENT RECEIVED')
                    input('Press Enter To Continue')
                    broker.execute_order(event)
                elif event.type == 'FILL':
                    print('FILL EVENT RECEIVED')
                    input('Press Enter To Continue')
                    port.updateFill(event)
                print('Analysis Complete')