Wolfram Mathematica Case Study

by Stefano Paggi

(Disclaimer: Entire PDF is a "screenshot" of the Mathematica notebook)

Benefits of Mathematica

- high level language
- 6000+ built-in functions!
- documentation with editable (!) examples
- super flexible with data types
- seamless flow from output to input
- no need for loading libraries
- no need for initialization
- UX similar to Jupyter Notebook

Conclusion: Mathematica is like an empty sheet of paper, your imagination (*AND* the area of the sheet) are your limits.
Goto for any "toy model" problem.

Factotum: Many areas Mathematica tackles

;

Source: Wolfram Mathematica Documentation Center

As an example: Trailing Stop with crossing average indicator

Case Study: We want to test historical stock market prices for automated trading for three cases:
1) Long entry on bullish moving average crossover with trailing stop, and time limit of 100 business days.
2) Short entry on bearish moving average crossover with trailing stop, and time limit of 100 business days.
3) Random entry on bearish moving average crossover with trailing stop, and time limit of 100 business days.

Definition of some variables

In[ ]:=
```
stopDistance = 10;
maxDays = 100;
```

Loading Financial data

We do not need to go online for stock data, Mathematica has a built-in function loading up financial data.

Load up Alphabet Inc. (Google) historical NASDAQ data from 1/1/2015 to 11/3/2023.

In[ ]:=
```
data = FinancialData["NASDAQ:GOOGL", "OHLC", {{2015, 1, 1}, {2023, 3, 11}}]
```

Out[ ]=
```
TimeSeries[ ⟨ Time: 02 Jan 2015 to 10 Mar 2023
              Data points: 2061 ⟩ ]
```

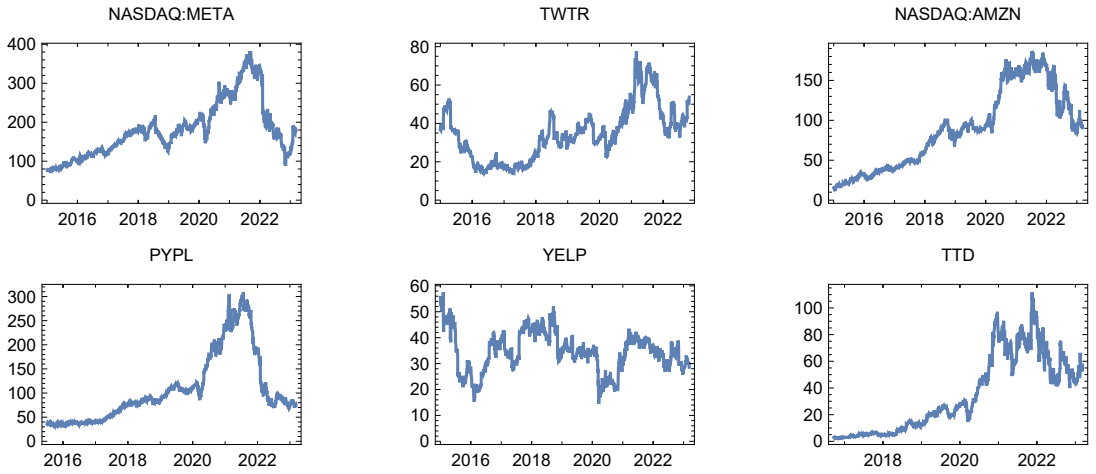
Plot Candlestick chart of stock price.

In[ ]:=
```
CandlestickChart[data]
```

Out[ ]=

In[ ]:=
```
GraphicsGrid[{DateListPlot[FinancialData[#, {{2015, 1, 1}, {2023, 3, 11}}],
      FrameLabel → {{None, None}, {None, #}}] & /@
    {"NASDAQ:META", "TWTR", "NASDAQ:AMZN"},
   DateListPlot[FinancialData[#, {{2015, 1, 1}, {2023, 3, 11}}],
      FrameLabel → {{None, None}, {None, #}}] & /@
    {"PYPL", "YELP", "TTD"}}]
```

Out[ ]=



The Google stock displays a similar behaviour to some big internet tech firms:
A slow climb over the years 2016 to 2018, followed by a bubble burst.
Going long (1) on the years prior to the burst should yield good money. Nevertheless,
we can also test if going short e.g. expecting a bearish market yields positive outcome (2) (3).

In[ ]:=

Moving Average (MA)

Calculate moving average for 100 days, 15 days.

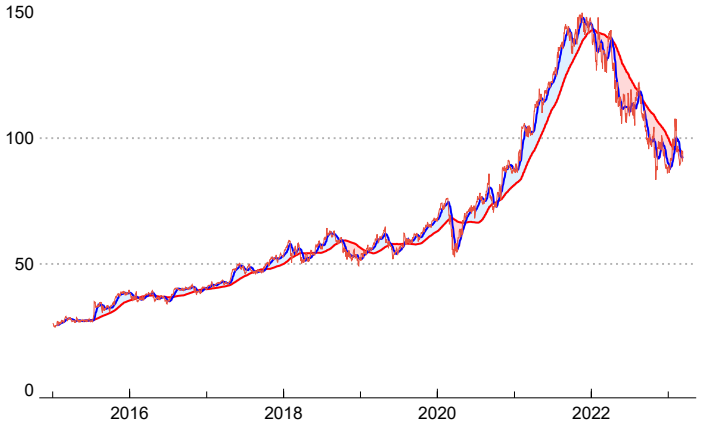
In[ ]:=
```
revData = Table[{d[[1]], QuantityMagnitude@# & /@ d[[2]]}, {d, data["Path"]}];
```

In[ ]:=
```
MovAv[sampleData_, tframe_] := Module[{movavList},
  movavList = MovingAverage[sampleData[[All, 2, 4]], tframe];
  Table[{sampleData[[Length[sampleData] - ma, 1]], movavList[[Length[movavList] - ma]]},
   {ma, 0, -1 + Length@movavList}]
 ]
```

Draw moving averages on historical.

In[ ]:=
```
DateListPlot[{MovAv[revData, 100], MovAv[revData, 15],
  {{revData[[All, 1]]} ~Join~ {revData[[All, 2, 4]]}}[[1]]^T},
 PlotTheme → "Business", PlotStyle → {
   Directive[Thickness@.003, Red],
   Directive[Thickness@.003, Blue],
   Directive[Thickness@.001]},
 Filling → {1 → {{2}, {LightBlue, LightRed}}}]
```

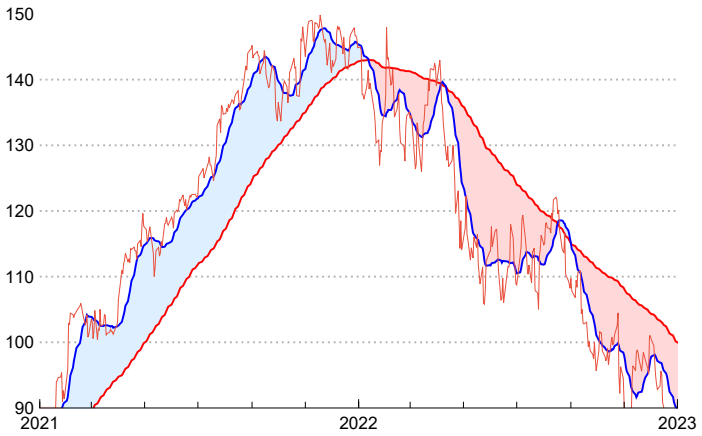
Out[ ]=



... a zoom-in into the “tech bubble” maximum.

In[ ]:=
```
DateListPlot[{MovAv[revData, 100], MovAv[revData, 15],
  {{revData[[All, 1]]} ~Join~ {revData[[All, 2, 4]]}}[[1]]^T},
 PlotTheme → "Business", PlotStyle → {
   Directive[Thickness@.003, Red],
   Directive[Thickness@.003, Blue],
   Directive[Thickness@.001]},
 Filling → {1 → {{2}, {LightBlue, LightRed}}},
 PlotRange → {{{2021, 1, 1}, {2023, 1, 1}}, {90, 150}}]
```

Out[ ]=



The very thin line represents the underlying stock price.
The blue line is the 15 day MA, the red line the 100 day MA.
A bearish crossover can be seen at 2022.

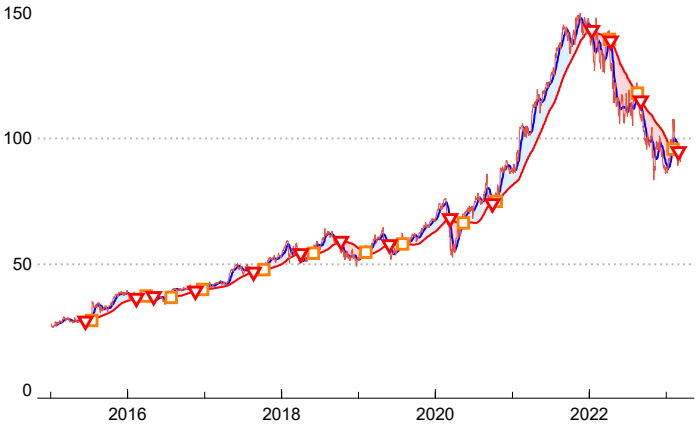
Determine Crossover

Following function determines automatically if there is a crossover, and

```
In[•]:= DetermineCrossOver[sampleData_, tframe1_, tframe2_] := Module[
          {tframe, avg1, avg2, avgboth, cross, ii, now, old, dates},
          avg1 = MovingAverage[sampleData[[All, 2, 4]], tframe1];
          avg2 = MovingAverage[sampleData[[All, 2, 4]], tframe2];
          If[tframe1 > tframe2,
           avgboth =
            Transpose[{avg1} ~Join~ {avg2[[Max[1, 1 + Length[avg2] - Length[avg1]] ;;]]}],
           avgboth = Transpose[{avg2} ~Join~ {avg1[[Max[1, 1 + Length[avg1] - Length[avg2]] ;;]]}]
          ];
          dates = Reverse@
            Table[sampleData[[Length[sampleData] - ma, 1]], {ma, 0, -1 + Length@avgboth}];
          cross = {};
          For[ii = 2, ii ≤ Length[avgboth], ii++,
           now = avgboth[[ii]];
           old = avgboth[[ii - 1]];
           If[(now[[2]] ≤ now[[1]]) && (old[[2]] ≥ old[[1]]),
            AppendTo[cross, {{dates[[ii]], (-now[[2]] × old[[1]] + now[[1]] × old[[2]])/
                (now[[1]] - now[[2]] - old[[1]] + old[[2]])}, -1}]
           ];
           If[(now[[1]] ≤ now[[2]]) && (old[[1]] ≥ old[[2]]),
            AppendTo[cross, {{dates[[ii]], (-now[[2]] × old[[1]] + now[[1]] × old[[2]])/
                (now[[1]] - now[[2]] - old[[1]] + old[[2]])}, 1}]
           ];
          ];
          cross
         ]
```

In[ ]:=
```
bullCrossOver = Cases[DetermineCrossOver[revData, 100, 15], {__, 1}][[All, 1]];
bearCrossOver = Cases[DetermineCrossOver[revData, 100, 15], {__, -1}][[All, 1]];
DateListPlot[{MovAv[revData, 100], MovAv[revData, 15],
  {{revData[[All, 1]]} ~Join~ {revData[[All, 2, 4]]}}[[1]]^T,
  bullCrossOver,
  bearCrossOver},
 Joined → {True, True, True, False, False},
 PlotTheme → "Business", PlotStyle → {
   Directive[Thickness@.003, Red],
   Directive[Thickness@.003, Blue],
   Directive[Thickness@.001],
   Directive[Orange]},
 PlotMarkers → {None, None, None, "OpenMarkers", "OpenMarkers"},
 Filling → {1 → {{2}, {LightBlue, LightRed}}}]
```

Out[ ]=



The red, downward triangles signify bearish crossovers,
the orange rectangles bullish crossovers.

Going Long

The function sets a long position at a position in time, with
a traling stop at a distance below the entry position.
If the trailing stop is not hit, the position is exited after a
given timelimit.
Returns the exit point, whether it hit a stop or not, the last position of the
stop, the first position of the stop.

```
In[•]:= TrailingStopTradingBull[historical_, timelimit_, enter_, distance_] :=
 Module[
  {lossGap, etime, eprice, position, out, now, old},
  etime = enter[[1]];
  eprice = enter[[2]];
  lossGap = eprice - distance;
  out = {};
  For[position = 1 + Position[historical, etime][[1, 1]],
   position ≤ Min[Position[historical, etime][[1, 1]] + timelimit, Length[historical]],
   position++,
   now = historical[[position]];
   old = historical[[position - 1]];
   If[now[[2, 4]] ≤ lossGap,
    out = {{now[[1]], now[[2, 4]]}, {1, {now[[1]], lossGap}, {etime, eprice - distance}}};
    Break[],
    If[now[[2, 4]] ≥ old[[2, 4]],
     If[now[[2, 4]] - lossGap ≥ distance,
      lossGap = now[[2, 4]] - distance
     ]
    ]
   ];
   out = {{now[[1]], now[[2, 4]]}, {0, {now[[1]], lossGap}, {etime, eprice - distance}}}
  ];
  out
 ]
```

Code

```
In[•]:= vrevData = revData;
```

Get all bullish crossover entry points.

```
In[•]:= entriesLong = Cases[DetermineCrossOver[vrevData, 100, 15], {__, 1}][[All, 1]]
```

```
Out[•]= {{3 645 907 200, 27.6664}, {3 668 198 400, 37.3909},
 {3 678 480 000, 36.8388}, {3 691 353 600, 39.9963},
 {3 716 496 000, 47.897}, {3 736 800 000, 54.3856}, {3 758 313 600, 54.8059},
 {3 773 433 600, 58.0809}, {3 798 316 800, 66.4373}, {3 811 795 200, 74.9848},
 {3 858 192 000, 139.395}, {3 869 683 200, 118.042}, {3 884 284 800, 95.7569}}
```

Trade long on those entry points with a trailing stop.

```
In[•]:= exitsLong = TrailingStopTradingBull[vrevData, maxDays, #, stopDistance] & /@
          (Cases[DetermineCrossOver[vrevData, 100, 15], {__, 1}][[All, 1]])
```

```
Out[•]= {{{3 658 176 000, 38.9605}, {0, {3 658 176 000, 29.1895}, {3 645 907 200, 17.6664}}},
         {{3 680 467 200, 40.1375}, {0, {3 680 467 200, 30.4245}, {3 668 198 400, 27.3909}}},
         {{3 690 748 800, 40.7825}, {0, {3 690 748 800, 31.787}, {3 678 480 000, 26.8388}}},
         {{3 704 054 400, 47.525}, {0, {3 704 054 400, 38.2305}, {3 691 353 600, 29.9963}}},
         {{3 729 196 800, 54.738}, {0, {3 729 196 800, 49.378}, {3 716 496 000, 37.897}}},
         {{3 749 241 600, 55.7455}, {0, {3 749 241 600, 54.275}, {3 736 800 000, 44.3856}}},
         {{3 768 508 800, 51.937}, {1, {3 768 508 800, 54.81}, {3 758 313 600, 44.8059}}},
         {{3 785 702 400, 67.822}, {0, {3 785 702 400, 58.035}, {3 773 433 600, 48.0809}}},
         {{3 808 771 200, 75.788}, {1, {3 808 771 200, 75.8695}, {3 798 316 800, 56.4373}}},
         {{3 824 496 000, 102.5}, {0, {3 824 496 000, 95.931}, {3 811 795 200, 64.9848}}},
         {{3 858 624 000, 128.824}, {1, {3 858 624 000, 129.395}, {3 858 192 000, 129.395}}},
         {{3 870 720 000, 109.42}, {1, {3 870 720 000, 110.17}, {3 869 683 200, 108.042}}},
         {{3 884 889 600, 95.01}, {1, {3 884 889 600, 97.64}, {3 884 284 800, 85.7569}}}}
```

```
In[•]:= startStops = Cases[exitsLong, {__, {1, __}}][[All, 2, 2]]
```

```
Out[•]= {{3 768 508 800, 54.81}, {3 808 771 200, 75.8695},
         {3 858 624 000, 129.395}, {3 870 720 000, 110.17}, {3 884 889 600, 97.64}}
```

Calculate gain for an entry bet of $10 000.

```
In[•]:= net = Table[exitsLong[[di, 1, 2]] - entriesLong[[di, 2]], {di, 1, Length@exitsLong}]
        dollars = 10 000;
        gain = Table[dollars / entriesLong[[di, 2]] * net[[di]], {di, 1, Length@exitsLong}]
        percentage = (# / dollars * 100) & /@ gain
```

```
Out[•]= {11.2941, 2.74655, 3.94369, 7.52868, 6.84097, 1.35989,
         -2.86891, 9.74111, 9.35068, 27.5152, -10.5707, -8.62192, -0.746884}
```

```
Out[•]= {4082.22, 734.551, 1070.53, 1882.34, 1428.27, 250.046,
         -523.468, 1677.16, 1407.44, 3669.43, -758.33, -730.412, -77.9979}
```

```
Out[•]= {40.8222, 7.34551, 10.7053, 18.8234, 14.2827, 2.50046,
         -5.23468, 16.7716, 14.0744, 36.6943, -7.5833, -7.30412, -0.779979}
```

Total gain.

```
In[•]:= gainLong = gain;
```

```
In[•]:= Plus @@ gain
```

```
Out[•]= 14 111.8
```

```
In[•]:= Cases[gain, _?Negative]
```

```
Out[•]= {-523.468, -758.33, -730.412, -77.9979}
```

Function creating table displaying results.

```
In[•]:= (Round[DateDifference[FromAbsoluteTime[entriesShort[[#, 1]]],
              FromAbsoluteTime[exitsShort[[#, 1, 1]]] //
            QuantityMagnitude, .01]) & /@ Range[1, Length@entriesLong]
```

```
Out[•]= {136., 146., 145., 150., 144., 106., 148., 59., 34., 37., 19., 51., 69.}
```

```
In[ ]:= NiceTable[exitsShort_, entriesShort_, gain_] :=
 Column[{Style["Cash per bet: " <> ToString[10000], Bold],
    TableForm[Table[{Style[Round[gain[[ii]], .01], Bold],
       Round[entriesShort[[ii, 2]], .01], Round[exitsShort[[ii, 1, 2]], .01],
       Round[exitsShort[[ii, 2, 2, 2]], .01], Round[exitsShort[[ii, 2, 3, 2]], .01],
       Switch[exitsShort[[ii, 2, 1]], 1, "Yes", 0, ""],
       Round[DayCount[FromAbsoluteTime[entriesShort[[ii, 1]]], FromAbsoluteTime[
          exitsShort[[ii, 1, 1]]], "BusinessDay"], .01]}, {ii, 1, Length@entriesShort}],
     TableHeadings → {None, {Style["Net", Bold], "Entry", "Exit", "Last Stop",
        "Initial Stop", "Hit stop?", "Time lapsed in days"}}],
    Style[
     "TOTAL GAIN: " <> ToString[Round[Plus @@ Cases[gain, _?Positive], .01]], Bold],
    Style[
     "TOTAL LOSS: " <> ToString[Round[Plus @@ Cases[gain, _?Negative], .01]], Bold],
    Style["TOTAL: " <> ToString[Round[Plus @@ gain, .01]], Bold]}] // TraditionalForm
```

Function plotting MAs, crossovers, and trailing stops.

```
In[ ]:= NicePlot[vrevData_, exitsLong_, options___] :=
 DateListPlot[{MovAv[vrevData, 100], MovAv[vrevData, 15],
   {{vrevData[[All, 1]]} ~ Join ~ {vrevData[[All, 2, 4]]}}[[1]]^T,
   Cases[DetermineCrossOver[vrevData, 100, 15], {__, 1}][[All, 1]],
   Cases[DetermineCrossOver[vrevData, 100, 15], {__, -1}][[All, 1]],
   Cases[exitsLong, {__, {1, __}}][[All, 2, 3]],
   Cases[exitsLong, {__, {1, __}}][[All, 2, 2]]
  },
  Joined → {True, True, True, False, False, False, False},
  PlotTheme → "Business", PlotStyle → {
    Directive[Thickness@.003, Red],
    Directive[Thickness@.003, Blue],
    Directive[Thickness@.001],
    Directive[Orange],
    Directive[Red],
    Directive[Gray],
    Directive[Black]},
  PlotMarkers → {None, None, None, "OpenMarkers",
    "OpenMarkers", {"-", Scaled[.035]}, {"-", Scaled[.07]}},
  Filling → {1 → {{2}, {LightBlue, LightRed}}}, options]
```

Plot

```
In[ ]:=
```

In[ ]:=
```
NiceTable[exitsLong, entriesLong, gain]
NicePlot[vrevData, exitsLong]
NicePlot[vrevData, exitsLong, PlotRange → {{{2019, 1, 1}, {2021, 1, 1}}, {30, 100}}]
```

Out[ ]//TraditionalForm=

Cash per bet: 10000

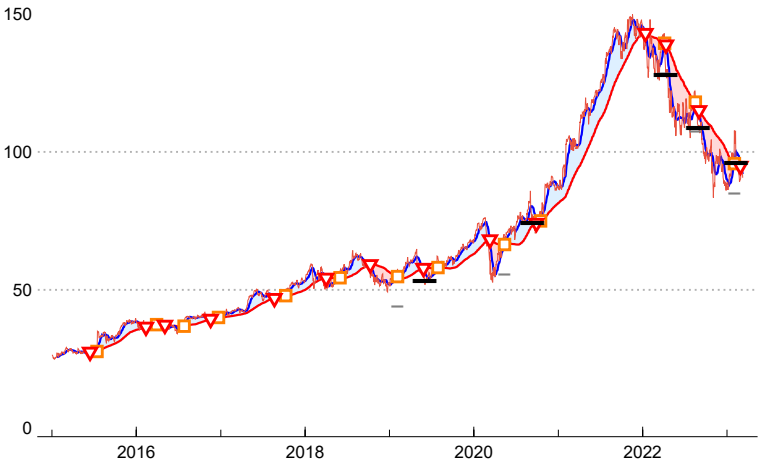
Net	Entry	Exit	Last Stop	Initial Stop	Hit stop?	Time lapsed in days
4082.22	27.67	38.96	29.19	17.67		98.
734.55	37.39	40.14	30.42	27.39		100.
1070.53	36.84	40.78	31.79	26.84		98.
1882.34	40.	47.53	38.23	30.		101.
1428.27	47.9	54.74	49.38	37.9		99.
250.05	54.39	55.75	54.28	44.39		99.
−523.47	54.81	51.94	54.81	44.81	Yes	82.
1677.16	58.08	67.82	58.04	48.08		98.
1407.44	66.44	75.79	75.87	56.44	Yes	84.
3669.43	74.98	102.5	95.93	64.98		99.
−758.33	139.39	128.82	129.39	129.39	Yes	3.
−730.41	118.04	109.42	110.17	108.04	Yes	8.
−78.	95.76	95.01	97.64	85.76	Yes	5.

TOTAL GAIN: 16202.

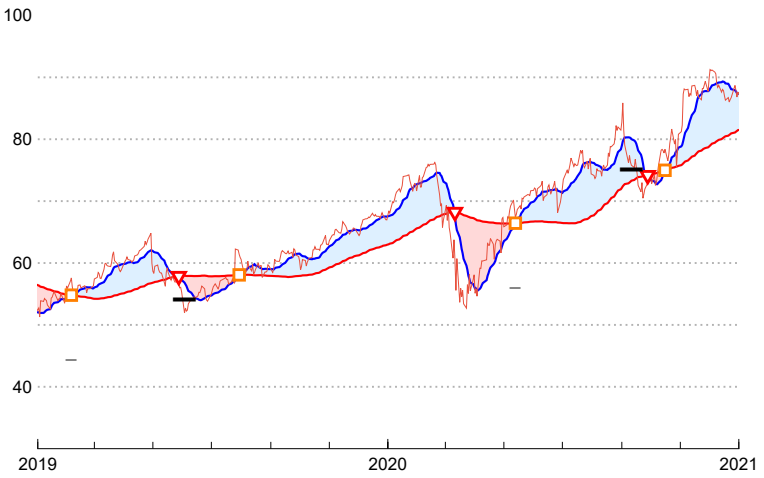
TOTAL LOSS: −2090.21

TOTAL: 14111.8

Out[ ]=



Out[ ]=



The thin gray bar signifies the stop at the entry of the position.

The thick black bar is the last position of the stop. Here, the price rapidly declined, and the stop got triggered.

Selling

The Mathematica code is analogous to above, just that at each bear crossover we go short with a reversed trailing stop.

Code

Plot

In[ ]:=
```
NiceTable[exitsShort, entriesShort, gainShort]
NicePlot[vrevData, exitsShort]
NicePlot[vrevData, exitsShort, PlotRange → {{{2019, 1, 1}, {2021, 1, 1}}, {30, 100}}]
```

Out[ ]//TraditionalForm=

Cash per bet: 10000

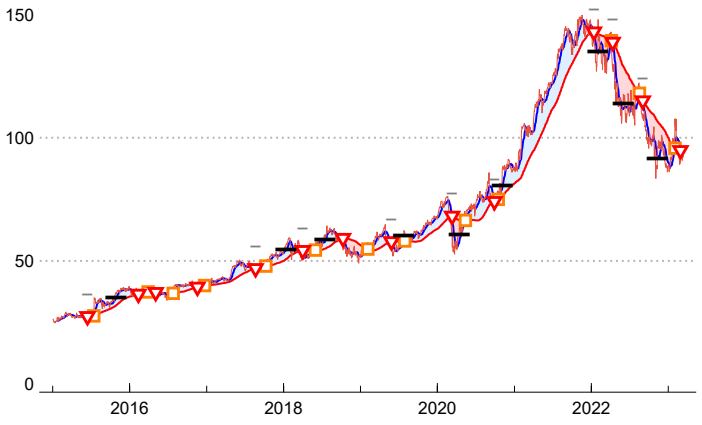
Net	Entry	Exit	Last Stop	Initial Stop	Hit stop?	Time lapsed in days
−3554.04	27.48	37.24	37.	37.48	Yes	95.
325.07	36.55	35.36	44.06	26.55		101.
−773.43	37.25	40.13	44.06	27.25		100.
−848.68	39.41	42.76	48.22	29.41		101.
−2041.83	46.95	56.53	56.41	56.95	Yes	98.
−1170.11	54.3	60.65	60.5	64.3	Yes	74.
282.02	59.21	57.54	59.23	49.21		100.
−758.94	57.87	62.26	61.94	67.87	Yes	42.
727.06	68.22	63.26	62.71	78.22	Yes	24.
−1768.01	74.18	87.29	82.55	84.18	Yes	26.
375.77	143.02	137.64	136.93	153.02	Yes	12.
1528.33	138.84	117.62	115.81	148.84	Yes	36.
1834.07	115.04	93.94	93.43	125.04	Yes	47.
445.65	94.86	90.63	100.63	84.86		8.

TOTAL GAIN: 5517.98

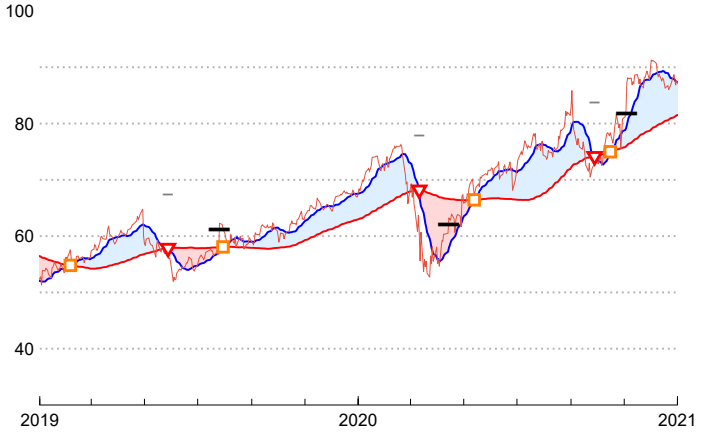
TOTAL LOSS: −10915.

TOTAL: −5397.06

Out[ ]=



Out[ ]=



Many short positions triggered the trailing stop, and
the total loss is bigger than the gain.

Random Selling

The trader expects a downward, bearish trend
on bearish crossovers. However, before entering
a position on the crossover, the trader decides on his own
whether to go long or short.
To simulate this behaviour, we randomize
going short or long.

The Mathematica code is analogous to above,
just that at each bear crossover there is 50/50
chance for long/short entry

Code

Plot

In[ ]:=
```
NiceTable[exitsRand, entriesShort, gainRandom]
NicePlot[vrevData, exitsRand]
```

Out[ ]//TraditionalForm=

Cash per bet: 10000

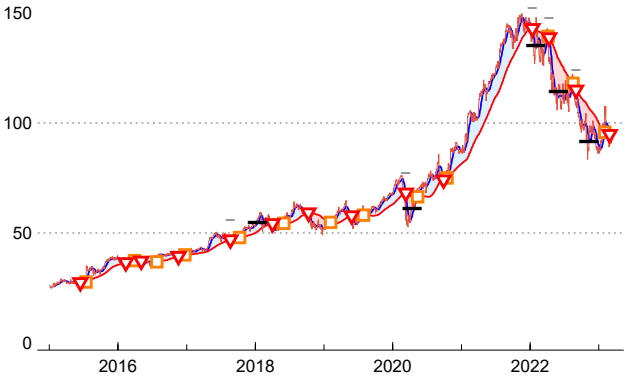
Net	Entry	Exit	Last Stop	Initial Stop	Hit stop?	Time lapsed in days
3744.38	27.48	37.77	27.77	17.48		99.
325.07	36.55	35.36	29.38	26.55		101.
773.43	37.25	40.13	30.8	27.25		100.
848.68	39.41	42.76	33.62	29.41		101.
−2041.83	46.95	56.53	56.41	56.95	Yes	98.
1249.94	54.3	61.09	54.28	44.3		100.
282.02	59.21	57.54	49.21	49.21		100.
824.43	57.87	62.64	52.64	47.87		99.
−727.06	68.22	63.26	62.71	78.22	Yes	24.
3846.86	74.18	102.71	95.93	64.18		98.
−375.77	143.02	137.64	136.93	153.02	Yes	12.
−1528.33	138.84	117.62	115.81	148.84	Yes	36.
−1834.07	115.04	93.94	93.43	125.04	Yes	47.
−445.65	94.86	90.63	100.63	84.86		8.

TOTAL GAIN: 11894.8

TOTAL LOSS: −6952.72

TOTAL: 4942.1

Out[ ]=



DifferentRuns

To check if the random behaviour has a tendency to positive gains,
we need to run the simulation over different choices for going long or short.
Mathematica creates a selection of pseudorandom choices of “long entry”,
or “short entry”. We do not check on unique selection, because there are

In[ ]:=
```
2^(Length@exitsShort)
```

Out[ ]= 16 384

permutations. With 100 choices the probability of choosing twice the same
choices is

```
In[•]:= 100 * (1 / (2^(Length@exitsShort)))^2 // N
```

```
Out[•]= 3.72529 × 10^-7
```

```
In[•]:= gain = Table[dollars / entriesShort[[di, 2]] * net[[di]], {di, 1, Length@exitsShort}];
        dollars = 10000;
        net = Table[-exitsShort[[di, 1, 2]] + entriesShort[[di, 2]], {di, 1, Length@exitsShort}];
```

```
In[•]:=
```

Do the simulation 100 times over.

```
In[•]:= randRunGains = {};
        Do[
          choice = RandomChoice[{-1, 1}, Length@shortSignals];
          exitsRand = Table[Switch[choice[[d]], 1,
              TrailingStopTradingBull[vrevData, maxDays, shortSignals[[d]], stopDistance], -1,
              TrailingStopTradingBear[vrevData, maxDays, shortSignals[[d]], stopDistance]],
            {d, 1, Length@shortSignals}];
          netRandom = Table[choice[[di]] * Abs[exitsRand[[di, 1, 2]] - entriesShort[[di, 2]]],
            {di, 1, Length@exitsShort}];
          gainRandom =
           Table[dollars / entriesShort[[di, 2]] * netRandom[[di]], {di, 1, Length@exitsShort}];
          percentageRandom = (# / dollars * 100) & /@ gainRandom;
          AppendTo[randRunGains, Plus @@ gainRandom],
          100];
        Chop[randRunGains]
```

```
Out[•]= {-7578.18, 13248., -4001.36, -8549.75, -3045.07, 4913.15, -5949.91, -3963.34,
         4020.54, -1838.86, 5993.8, 7642.96, 760.275, 9005.74, -9885.65, 7732.16, 13360.9,
         401.019, -6262.6, 6942.98, -10157., -3806.8, 2345.14, -7405.25, -3837.38,
         1479.94, 9772.11, -8122.3, 6605.07, -1621.14, 5105.01, -5041.38, 2339.83,
         -4779.57, 8006.37, -3507.11, -7386.52, 674.898, 1074.28, 5155.66, -1643.62,
         9616.3, -544.559, 8874.94, 5853.03, 2750.41, 12811.1, 4667.89, -3926.23, -3485.98,
         8565.71, 4716.99, 4312.59, 2926.3, 6063.61, 4615.79, 9735.6, 8802.92, -3855.96,
         10074.2, -7372.14, -2835.84, 3359.8, -7131.77, 1213.89, -3465.83, -1392.43,
         8238.89, 4414.01, 986.43, -3859.7, -4883.95, 523.353, 1984.88, -745.229,
         4962.06, -2674.3, 1419.29, 2797.44, -9322.77, 9857.76, -2899.86, 9728.43,
         5122.96, -1498.79, 7357.48, 413.419, 2277.69, -5803.15, -1066.9, -1484.47,
         3433.98, 5279.38, 5563.59, -4983.57, 1219.2, 7860.24, 9965.03, 10544.4, 5351.28}
```

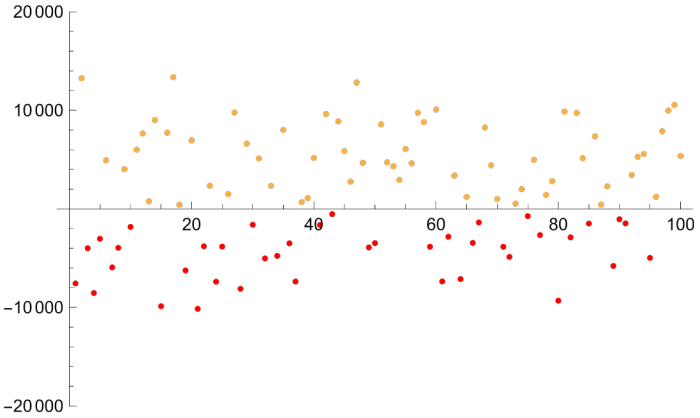
```
In[•]:= 1 / 100 * Plus @@ randRunGains
```

```
Out[•]= 1532.26
```

Plot net results over iteration number.

```
In[•]:= ListPlot[Table[{d, randRunGains[[d]]}, {d, 1, Length@randRunGains}],
         ColorFunction → Function[{x, y}, If[y > 0, , Red]],
         ColorFunctionScaling → False, PlotRange → {-20000, 20000}]
```

Out[•]=

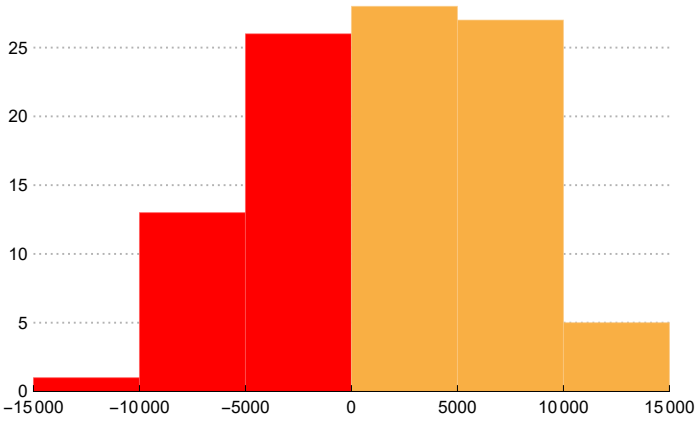


The blue dots above the zero dollars seem to be in the majority.

```
In[•]:= cedf2[from_, to_, color_ : Red, t_ : Small, rr_ : 0][
          {{xmin_, xmax_}, {ymin_, ymax_}}, ___] := {If[from < xmax ≤ to, color, Sequence[]],
          Dynamic@EdgeForm[Directive[Thickness[t], Lighter@CurrentValue["Color"]]],
          Rectangle[{xmin, ymin}, {xmax, ymax}, RoundingRadius → rr]}
```

```
In[•]:= Histogram[randRunGains, "Scott", AxesOrigin → {0, -1},
         ChartElementFunction → cedf2[-20000, 0, Red, Small, 3], PlotTheme → "Business"]
```

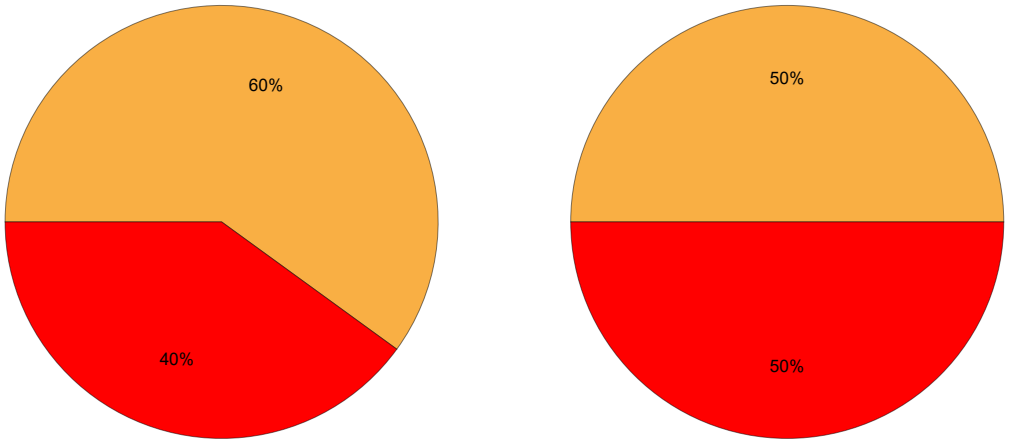
Out[•]=



The histogram shows that there is indeed a trend for positive outcomes.

In[ ]:=
```
percLoss = Length[Cases[randRunGains, _?Negative]];
percGain = Length[Cases[randRunGains, _?Positive]];
percLossS = Length[Cases[gainShort, _?Negative]] / Length[gainShort] * 100 // N;
percGainS = Length[Cases[gainShort, _?Positive]] / Length[gainShort] * 100 // N;
GraphicsRow[{PieChart[{percGain, percLoss},
    ChartLabels → {ToString[percGain] <> "%", ToString[percLoss] <> "%"},
    ChartStyle → {, Red}],
  PieChart[{percGainS, percLossS},
    ChartLabels →
     {ToString[Round[percGainS]] <> "%", ToString[Round@percLossS] <> "%"},
    ChartStyle → {, Red}]
 }]
```

Out[ ]=



The left pie chart shows the percentage for positive and negative (red) outcome on 100 tries, and the right chart for going short on every bearish crossover.
Introducing a “human” element can turn going short on a bull market fruitious.

Conclusion

In[ ]:=
```
TableForm[{Round[Plus @@ (#) / Length[#]] & /@ {Cases[gain, _?Positive],
      Cases[gainShort, _?Positive], Cases[randRunGains, _?Positive]},
    Round[Plus @@ (#) / Length[#]] & /@ {Cases[gain, _?Negative],
      Cases[gainShort, _?Negative], Cases[randRunGains, _?Negative]},
    Round[Plus @@ (#) / Length[#]] & /@ {gain, gainShort, randRunGains}},
  TableHeadings → {{"Average gain", "Average loss", "Average net"},
    {"Long", "Short", "Short + Random"}}] // TraditionalForm
```

Out[ ]//TraditionalForm=

	Long	Short	Short + Random
Average gain	1800	788	5581
Average loss	−523	−1559	−4540
Average net	1086	−386	1532

Statistics on $10 000 bet.

In this case study we took a look at three different trading systems in a highly simplified toy model. The long entry behaviour proved to yield the best results, due to the historical data's climbing prices. The short entry behaviour on a single MA crossover indicator yielded poor results, but extending the indicator by a "human element" deciding whether to go short or long on the indicator, yielded a positive outcome over 100 trials.

The primary goal of this case study was to present the elegance, and versatility of Wolfram Mathematica. Of course the study on the stock market is a first approximation, painting an over-simplified picture e.g. by neglecting the trading volumes, the regulatory influence, intraday volatility, composit bets, balance, or reinforcing indicators by other indicators.
Furthermore, determining a reasonable stop position is also worth investigating.
In conclusion, it is a ripe ground for further implementations.

Stefano Paggi, 2023