Drawdown of Historical Stock Prices

Introduction

The drawdown of a stock indicates how much time it's spent "underwater" - it's essentially the percentage drop of its price from a peak to a trough, with the drawdown resetting to zero if a previous high is reached. The drawdown of a stock is a valuable risk measure and is employed by traders to gauge volatility.

This application

- downloads historical stock prices from Yahoo Finance for a chosen ticker symbol,
- defines a procedure that calculates the drawdown of the historical stock price
- and plots the drawdown against the adjusted close price of the asset

By changing the ticker and the dates, you can examine drawdown of any stock between your chosen dates.

```
> restart :
```

Ticker, Dates and Frequency

Download historical data for the S&P 500

```
> ticker := "^GSPC" :
> startDay := "1" : startMonth := "1" : startYear := "1975" :
> endDay := "1" : endMonth := "1" : endYear := "2011" :
> frequency := "m" :
```

Download Historical Stock Quotes

```
> url := cat( "http://ichart.finance.yahoo.com/table.csv?s=", ticker, "&a=", startMonth, "&b=",
      startDay, "&c=", startYear, "&d=", endMonth, "&e=", endDay, "&f=", endYear,
      "&g=",frequency,"&ignore=.csv") :
```

Strip out header row

```
> data := ImportMatrix( url) [2 ..,..]
```

$$data := \left[\begin{array}{c} \textit{433 x 7 Matrix} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{array}\right]$$

Note that the adjusted close price is the seventh column.

Reverse the matrix so it's in date ascending order.

```
> data := convert( ListTools:-Reverse( convert( data, listlist ) ), Matrix ) :
> nRows := LinearAlgebra:-RowDimension( data )
                                        nRows := 433
```

Calculate and Plot Drawdown

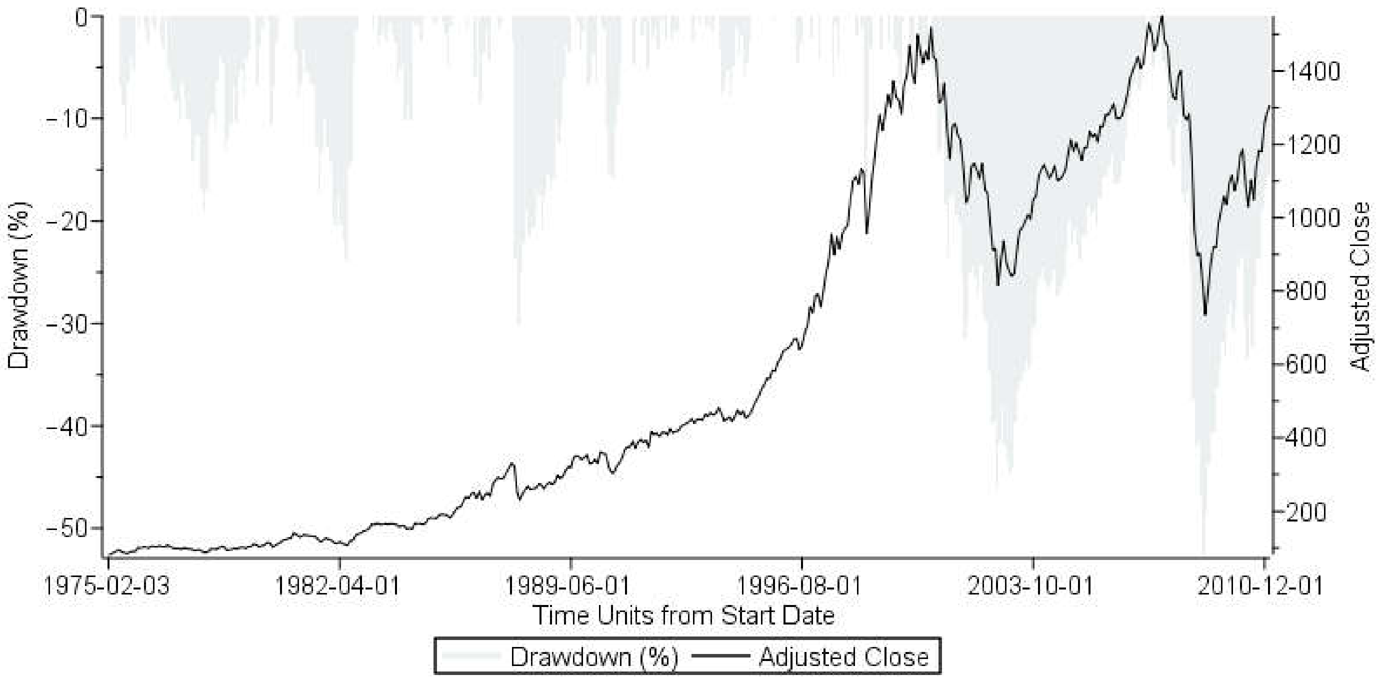
The algorithm is referenced from http://en.wikipedia.org/wiki/Drawdown_(economics)

DD is a vector that will be filled with the drawdown of the historical stock price

```
> DD := Vector( nRows, datatype = float[8] ) :
> peak := -99999 :
  for i from 1 to nRows do
    if data[i, 7] > peak then
      peak := data[i, 7] :
    end if:
    DD[i] := 100·( peak − data[i, 7] ) / peak :
  end do:
```

The maximum drawdown is

```
> max( DD )
                                   52.5558594600000006
> p1 := Statistics[ColumnGraph]( -DD, thickness = 0, color = COLOR( RGB, 236/255, 240/255, 241/255 ), distance
      = 10^(-6), width = 1, labels = ["Time Units from Start Date", "Drawdown (%)"], labeldirections
      = [horizontal, vertical], labelfont = [Arial], style = patchnogrid, legend = "Drawdown (%)",
      legendstyle = [font = [Arial]], axesfont = [Arial] ) :
> p2 := plots:-pointplot( [seq( [i, data[i, 7]], i = 1 ..nRows )], connect = true, color = black, thickness = 0,
      labels = ["Time Units from Start Date", "Adjusted Close"], labeldirections = [horizontal, vertical],
      labelfont = [Arial], legend = "Adjusted Close", legendstyle = [font = [Arial]], axesfont = [Arial] ) :
> plots:-dualaxisplot( p1, p2, size = [800, 400], tickmarks = [[seq( i = data[i, 1], i = 1 ..nRows,
      floor( nRows·0.2 ) )], decimalticks])
```


Drawdown (%)
0
−10
−20
−30
−40
−50
Adjusted Close
1400
1200
1000
800
600
400
200
1975-02-03
1982-04-01
1989-06-01
1996-08-01
2003-10-01
2010-12-01
Time Units from Start Date
Drawdown (%)
Adjusted Close