Estimation of the Hurst Exponent for Market Indices by Use of Rescaled Range Method and CAS *Mathematica*. Case study: S&P500

Zuzanna Rzeszotko
e-mail: z.rzeszotko@gmail.com
University of Natural Sciences and Humanities in Siedlce
Siedlce, Poland

Abstract

Methodology of estimating the Hurst exponent using rescaled range method by use of Computer Algebra System "Mathematica" is presented. Depending of its value this measure of persistence indicates a random walk, "mean-reverting" series or trend reinforcing series. In this paper long−term memory of a time series of S&P500 index weekly returns is tested based on the Hurst exponent value.

1. Introduction

The "Black Monday" (October 19, 1987), when the index of American Stock Exchange Dow Jones Industrial Average diminished by 22 percent (together with many other market indices all over the world), became for economists an impulse to apply new methods in order to predict market fluctuations. Nowadays, such time series like for example changes of market index prices are analyzed using many different methods, not only the technical analysis, methods of stochastic processes theory or econometric models, but also theory of chaos [5]. Aperiodic and nonlinear nature of market often cannot be described using simple laws of mathematical statistics or economy. Therefore, the theory of finance had to be supported by methods adapted from the theory of chaos and nonlinear dynamics. In the theory of chaotic systems different methods are used [5], among others such as: correlation dimension [15]; Kolmogorov entropy, Lapunov exponent, fractal dimension [13]; Brock-Dechert-Scheinkman's test, rescaled range analysis [8,9,10,12].

In this paper I present the methodology of determining the Hurst exponent for market indices. The main goal is to implement the Computer Algebra System *Mathematica* to calculate the rescaled range (*R/S*) in order to obtain the Hurst exponent value for the chosen data set.

The inventor of the Hurst exponent – an English hydrologist H.E.Hurst – has formulated and has put into use the *R/S* method in order to examine the level of Nile overflows (he had access to data involving the level of this river overflows dating back about 847 years). H.E.Hurst deduced [3,4], that the differences of size for succeeding Nile overflows in respect to time behave as the processes with long memory, which are similar to processes of fractional Brownian motion [6,7]. They are not independent and they are not normally distributed.

H.E.Hurst observed that the difference grows faster than it could be expected in accordance with the relation $R = c\sqrt{T}$, by which A. Einstein described (in 1908) the distance R done by a particle during time interval T (c – constant). He noticed also, that the Nile overflows are not random, but they follow some "pattern". He succeeded to define a constant K, measuring the fractional Brownian motion and to determine a way to calculate this constant – the rescaled range method. (In 1968 Mandelbrot named the "pattern" a fractal and renamed the constant K in the

honour of H.E.Hurst to *H*.) It turned out, that the fractal dimension and the Hurst exponent are in close dependence, namely *D=2–H*. The Hurst exponent measures a smoothness of fractal objects and its value is between 0 and 1.

Starting from the works of E.E.Peters [8,9,10], the Hurst exponent began to be applied in finance and despite nowadays there are many methods known, which allow to determine its value (e.g. the method of aggregated variance, periodogram method, Whittle's method, Abri–Veitch's method and others), the method of rescaled range is still the most widely known one.

2. The rescaled range analysis – algorithm

The scheme given below follows descriptions from the papers listed in the bibliography [5,7,8,14]. We describe the algorithm as follows:

0. We set a series of experimental data $(x_1, x_2, ..., x_T)$.

1. We transform the above series of prices into the series of logarithmic rates of return according to the formula:

$$y_t = \log\left(\frac{x_{t+1}}{x_t}\right)$$

where *t=1,...,T-1*. The length of the resulting series will be denoted by *N=T-1*.

2. We analyze the series $(y_t)_{t=1}^{T-1}$ with respect to time intervals of different length, that is we divide this series into m subseries $(A_1, A_2, ..., A_m)$, each of length *n* $(m \cdot n = N)$. For *j=1,2,...,m* one has $A_j = (N_{k,j})_{k=1}^n$, hence the mean value of A_j is equal to:

$$e_j = \frac{\sum_{k=1}^n N_{k,j}}{n}$$

3. The next step is to sum the deviations from the mean value, to obtain the cumulated sum in each point (*k=1,2,...,n* and *j=1,2,...,m*):

$$X_{k,j} = \sum_{i=1}^{k}(N_{i,j} - e_j)$$

4. We calculate the range for each subsequence A_j:

$$R_j = \max_k(X_{k,j}) - \min_k(X_{k,j}), \qquad 1 \le k \le n$$

5. We calculate the standard deviation S_j of $N_{k,j}$ in the interval, where local mean value equals e_j

$$S_j = \sqrt{\frac{1}{n}\sum_{k=1}^n (N_{k,j} - e_j)^2}$$

6. We determine the value of rescaled range for A_j: R_j/S_j.

7. We repeat the whole procedure (steps 2–6): we determine R_j/S_j for each interval of the data set and then we calculate the mean value $(R/S)_n$:

$$(R/S)_n = \frac{\sum_{j=1}^m \frac{R_j}{S_j}}{m}$$

We repeat the steps 1-7 for the consecutive *n*, dividing the set of data into disjoint intervals and finding the mean value $(R/S)_n$ of these intervals.

8. We draw a log–log graph, i.e. we use a linear regression *Y* with respect to *X*, where $Y = \log(R/S)_n$ and $X = \log n$. The slope coefficient of the regression line determines (is equal to) a value of the Hurst exponent.

3. Monte Carlo method

For random series the formula determining the expected value of the rescaled range was given successively (with some improvements) by: W. Feller [2], Anis and Lloyd [1], Peters [9], Purczynski [11].

In this paper the most precise formula worked out by J. Purczynski was used:

$$
E(R/S)_n = \begin{cases} \dfrac{n-0.5}{n} \cdot \dfrac{\text{Gamma}(0.5(n-1))}{\sqrt{\pi}\,\text{Gamma}(0.5\,n)} \cdot \sum_{i=1}^{n-1} \sqrt{\dfrac{n-i}{i}} & \text{dla } n < 30, \\[3mm] \dfrac{(n-0.5)(n-0.2491)}{(n-1)\sqrt{n\pi/2}} \cdot \left(\dfrac{\pi}{2} + \dfrac{-2.4718 + 3.5466n - 1.4635n^2}{(n-1)^{2.5}} \right) & \text{dla } n \geq 30. \end{cases}
$$

Theoretically, for the infinite normally distributed series the Hurst exponent tends to 0.5. Nevertheless in empirical research we usually work with the finite set of data, so we can compare the exponent H_{emp} derived from experimental data with the theoretical expected value $H_{\text{teor}} = E(H)$ calculated by use of the above formula. If the difference $H_{\text{emp}} - H_{\text{teor}}$ is big and $H>0.5$ then the series is probably normally distributed. The dependence $H_{\text{emp}} > H_{\text{teor}}$ indicates an effect of long memory and long-lasting autocorrelation. The opposite dependence stands for more unstable series.

4. Analysis of price series for market index S&P500 with use of CAS *Mathematica*

Mathematica is featured with the financial database; inter alia the quotations of the main world market indices. In order to illustrate the application of the rescaled range analysis with CAS *Mathematica* I have used the data of New York Stock Exchange (NYSE) index S&P500, which consists of 500 companies of the highest capitalization. The time period being under consideration: January 6, 1950 – November 18, 2011.

In[1]:=Data=FinancialData["SP500",{ "Jan. 6,1950", "Nov. 18, 2011"}]

Out[1]=

A very large output was generated. Here is a sample of it:

```
{{{1950, 1, 6}, 16.98}, {{1950, 1, 9}, 17.08}, {{1950, 1, 10}, 17.03}, {{1950, 1, 11}, 17.09},
{{1950, 1, 12}, 16.76}, «15 560», {{2011, 11, 14}, 1251.78}, {{2011, 11, 15}, 1257.81},
{{2011, 11, 16}, 1236.91}, {{2011, 11, 17}, 1216.13}, {{2011, 11, 18}, 1215.65}}
```
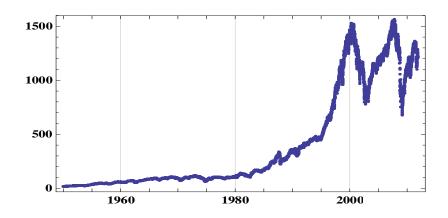
Show Less | Show More | Show Full Output | Set Size Limit…

For the visualization of data it is convenient to use the function DateListPlot, generating the plot of price fluctuations for the chosen market index, share or exchange rate with respect to time. (Unfortunately the resources of CAS *Mathematica* do not contain the data from smaller stock exchanges, such like for example the Warsaw Stock Exchange. There are accessible only the data of the biggest world stock exchanges like for example NYSE or LSE. In the *Mathematica* there is a possibility to load the external data, e.g. shares prices corresponding to the concrete time period saved in an Excel file, therefore it is not impossible to avoid this problem.)

```
In[2] := DateListPlot[FinancialData["SP500",{"Jan. 6, 1950", "Nov. 18, 2011"}],
         LabelStyle → Directive[Black, Medium, FontWeight → Bold]]
```
Out[2]=

5. χ -squared test of data distribution

Predicting of the given time series should be preceded by testing of its behaviour. If it is random, all the methods of prediction fail. In this connection we start to test our data distribution. To find out are the data normally distributed or not, one can use χ–square test [7], where the null hypothesis and the alternative hypothesis have respectively the form:

H_0: the distribution of weekly returns of the given index is a normal distribution $N(m, \sigma)$;
H_1: the distribution of weekly returns of the given index is not a normal distribution $N(m, \sigma)$;

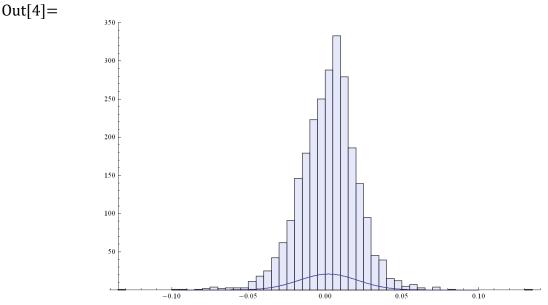To verify the null hypothesis we use the statistics:

$$\chi^2 = \sum_{i=1}^{r} \frac{(n_i - n \cdot p_i)^2}{n \cdot p_i},$$

which is consistent with χ – squared distribution with $k=(r\text{-}m\text{-}1)$ degrees of freedom, if the null hypothesis is true. Assignments: m – the number of parameters, calculated from data (in our case: mean value and standard deviation), r – number of classes, p_i – probability, that the value of characteristic X belongs to the i-th class. χ_e^2 – assigns an empirical value of the statistics; if the value is too big, we reject the null hypothesis. The relation $P(\chi^2 > \chi_\alpha^2) = \alpha$ defines a "critical set", while χ_α^2 means a critical value for the χ – squared distribution with $r\text{-}m\text{-}1$ degrees of freedom and $P = \alpha$.

     The build-in statistical functions of CAS *Mathematica* allow us to estimate (from the sample) the mean value μ and the standard deviation σ, and also to plot a histogram of our data ("WeeklyDataSP500" – assigns the set of weekly logarithmic returns of S&P500; the precise description of obtaining this set is given in the next part of this paper) together with the density function of normal distribution corresponding to the parameters μ (MeanValue) and σ (StandDev), which were calculated earlier.

```
In[3]:= MeanValue =Mean[WeeklyDataSP500];
        StandDev =StandardDeviation[WeeklyDataSP500]
```
Out[3]= 0.00206673
        0.0191886

In[4]:=Hist=Histogram[WeeklyDataSP500,PlotRange→{0,350}, ChartStyle→Opacity[0.5]];
   Norma=Plot[PDF[NormalDistribution[MeanValue, StandDev],x],{x,-0.1,0.1},
   PlotRange→{0,22}];Show[Hist,Norma]
Out[4]=

Assume the level of significance is equal to: $\alpha=0.05$.
In[5]:= α =0.05
Out[5]= 0.05

We determine the empirical number of elements in particular classes
In[6]:=NrElemClass=BinCounts[WeeklyDataSP500,{-0.1,0.08,0.03}]
Out[6]= {9,40,545,1559,345,19}

We define the right ends of intervals, dividing the set of data into classes.
In[7]:=RightEndOfIntervals=Table[-0.1+i*0.03,{i,1,6}]
Out[7]={-0.07,-0.04,-0.01,0.02,0.05,0.08}

We introduce standardized variable
In[8]:=StandardizedVariable=Table[(RightEndOfIntervals [[i]]-MeanValue)/StandDev,
{i,1,6}]
Out[8]= {-3.7557,-2.192277,-0.628848,0.934578,2.498,4.06143}

and we calculate the appropriate distribution function of normal distribution N(0,1).
In[9]:=Distribution=Table[CDF[NormalDistribution[0,1],StandardizedVariable[[i]]],
{i,1,6}]
Out[9]= {0.0000864295,0.0141799,0.264724,0.824997,0.993755,0.999976}

For each of r classes of values of random value X being under consideration, we calculate the probabilities p_i (assign. Probabilities), that the random value will belong to the i-th class, where $i=1,2,…,r$.

In[10]:= Probabilities =Prepend[Table[Distribution[[i]]- Distribution [[i-1]],{i,2,6}],
Distribution[[1]]]
Out[10]= {0.0000864295,0.0140935,0.2505445,0.560273,0.168758,0.006220375}

We define n
In[11]:=n=Total[NrElemClass]
Out[11]=2517
and the theoretical number of elements in particular classes npi, which should occur in the i-th class,
if the data had the assumed distribution.
In[12]:=npi=Table[Probabilities [[i]]*n,{i,1,6}]
Out[12]= {0.2175,35.47,630.6205,1410.206,424.76,15.65668}

We define the parameter r
In[13]:=r=Length[NrElemClass]
Out[13]=6

We determine the value "ChiSquared" of χ^2 statistics (if the null hypothesis is true, then the
statistics is compatible with chi-squared distribution of $r-1$ or of o $r-k-1$ degrees of freedom, when
there were k parameters of distribution estimated from the sample).
In[14]:=ChiSquared=Sum[(((NrElemClass[[k]]-npi[[k]])^2)/(npi[[k]]),{k,1,r}]//N
Out[14]= 398.152

The number of degrees of freedom is equal to $k=6-2-1=3$, and the critical value equals $\chi_{\alpha}^2 =$
$\chi_{0.05}^2 = 7.815$ (read from tables). We have $\chi^2 \geq \chi_{\alpha}^2$, so we reject the hypothesis about normality of
distribution. The process of weekly returns of S&P500 is not a standard (and not a fractional)
Brownian motion, because the distribution of its returns is not normally distributed (level of
significance α=0.05).

6. Preparation of the data
The first step is to calculate the logarithmic returns of the index (log(price of the given day/price of
the previous day)).

In[1] := DataSP500withGaps = Table[{Data[[i + 1,1]], Data [[i + 1,2]],
                        Log[E, Data [[i +1,2]]/ Data [[i,2]]]}, {i, 1, Length[Data] - 1}]
Out[1]=

A very large output was generated. Here is a sample of it:

{{{1950, 1, 9}, 17.08, 0.00587201}, {{1950, 1, 10}, 17.03, -0.00293169},
  {{1950, 1, 11}, 17.09, 0.003517}, ≪15 563≫, {{2011, 11, 16}, 1236.91, -0.0167558},
  {{2011, 11, 17}, 1216.13, -0.0169426}, {{2011, 11, 18}, 1215.65, -0.000394773}}

Show Less Show More Show Full Output Set Size Limit...

Because of the fact, that the data can be incomplete, one should then complete them. The below
procedure assigns the previous day price of the index to each date without quotation.

In[2]:=
CompletingData[LISTA_]:=Module[{},Lista=LISTA;NewList={};
Do[If[DateDifference[Lista[[i,1]],Lista[[i+1,1]]]==1,
NewList=Append[NewList,{DatePlus[Lista[[i,1]],0],Lista[[i,2]],Lista[[i,3]]}],
NewList=Join[NewList,Prepend[Table[{DatePlus[Lista[[i,1]],k],Lista[[i,2]],Lista[[i,3]]},
{k,1,DateDifference[Lista[[i,1]],Lista[[i+1,1]]]-1}],
{DatePlus[Lista[[i,1]],0],Lista[[i,2]],Lista[[i,3]]}]]],{i,1,Length[Lista]-1}];
NewList=Append[NewList,{DatePlus[Lista[[Length[Lista],1]],0],Lista[[Length[Lista],2]],
Lista[[Length[Lista],3]]}]]

In[3]:=
CompletedDataSP500=CompletingData[DataSP500withGaps]
Out[3]=

```
A very large output was generated. Here is a sample of it:

{{{1950, 1, 9}, 17.08, 0.00587201}, {{1950, 1, 10}, 17.03, -0.00293169},
 {{1950, 1, 11}, 17.09, 0.003517}, {{1950, 1, 12}, 16.76, -0.0194984}, {{1950, 1, 13}, 16.67, -0.0053844},
 «22 584», {{2011, 11, 14}, 1251.78, -0.00959608}, {{2011, 11, 15}, 1257.81, 0.00480558},
 {{2011, 11, 16}, 1236.91, -0.0167558}, {{2011, 11, 17}, 1216.13, -0.0169426}, {{2011, 11, 18}, 1215.65, -0.000394773}}

Show Less   Show More   Show Full Output   Set Size Limit…
```

I create now the table containing only the returns of the consecutive days, starting with the Monday, January 9, 1950, up to the Friday, November 18, 2011.
In[4]:=
DailyLogarithmicReturnsSP500=
              Table[CompletedDataSP500[[i,3]],{i,1,Length[CompletedDataSP500]}]
Out[4]=

```
A very large output was generated. Here is a sample of it:

{0.00587201, -0.00293169, 0.003517, -0.0194984, -0.0053844, -0.0053844, -0.0053844,
 0.00299491, 0.00833834, -0.000593296, «22 574», -0.0373853, 0.00858722, 0.0192932,
 0.0192932, 0.0192932, -0.00959608, 0.00480558, -0.0167558, -0.0169426, -0.000394773}

Show Less   Show More   Show Full Output   Set Size Limit…
```

In the next step one should decide which returns we want to analyze. In this paper I have studied the weekly logarithmic returns of the chosen index. (The way of determining the returns takes into account the fact, that the procedure of data completing introduced the quotations for Saturdays and Sundays, not only for the working days. Now I eliminate these "artificial" returns.)
In[5]:= WeeklyLogarithmicReturnsSP500=
Table[Sum[DailyLogarithmicReturnsSP500[[7i-k]],{k,2,6}],
{i,1,Floor[Length[DailyLogarithmicReturnsSP500]/7]}]
Out[5]=
{-0.018425484120498468,0.013702925166324985,-0.0047449673842258075,…}

In the rescaled range analysis one divides the data into equal intervals, changing their length in the consecutive steps. For calculations to be authoritative it would be good to use the same number of data by each of the divisions. For this purpose at the beginning we define the starting number of data "NumberOfData" and next we look for such number, that is smaller than "NumberOfData" and that has the greatest possible number of divisors. (I began the search of the appropriate number of returns from one third of the starting number of data – the results obtained from the very small number of data are not authoritative. [10, p. 85])

In[6]:=NumberOfData=Length[WeeklyLogarithmicReturnsSP500]
Out[6]=3227
In[7]:= ListOfDivisorsNumber=
Table[Length[Divisors[i]],{i, Floor[NumberOfData /3], NumberOfData }]
Out[7]={6,6,4,12,4,32,4,4,6,6,8,8,...}

In[8]:=Nr=Position[ListOfDivisorsNumber ,Max[ListOfDivisorsNumber]][[1,1]]
Out[8]=1446
In[9]:=RightNumberOfData=((Floor[NumberOfData/3]-1)+Nr)
Out[9]=2520
In[10]:=DIVISORS=Divisors[RightNumberOfData]
Out[10]={1,2,3,4,5,6,7,8,9,10,12,14,15,18,20,21,24,28,30,35,36,40,42,45,56,60,63,70,72,84,
90,105,120,126,140,168,180,210,252,280,315,360,420,504,630,840,1260,2520}

For further calculations I used 2520 weekly logarithmic returns.
In[11]:=WeeklyDataSP500=Take[WeeklyLogarithmicReturnsSP500, RightNumberOfData]
Out[11]={−0.018425484120498468,0.013702925166324985...}

7. Determination of the Hurst exponent by the use of CAS-*Mathematica*
The next step is to build a procedure, determining the rescaled range for the concrete lengths of intervals, according to the scheme described above.

```
In[12]:=
RescaledRange[u_,V_]:=Module[{n=u,w,LogarithmicReturnsSeries=V,
m,NewListOfReturns,RSn,NN},
NN=Length[LogarithmicReturnsSeries]-1;
m=Floor[NN/n];
w=m*n;
NewListOfReturns =Take[LogarithmicReturnsSeries,w];
A=Table[Take[NewListOfReturns,{1+(i-1)*n,n+(i-1)*n}],{i,1,m}];
e=Table[N[Sum[A[[j]][[k]],{k,n}]/n,10],{j,1,m}];
X=Table[N[Sum[A[[j]][[i]]-e[[j]],{i,k}],10],{k,1,n},{j,1,m}];
R=Table[N[Max[Table[X[[k]][[j]],{k,n}]]-Min[Table[X[[k]][[j]],{k,n}]],10],{j,1,m}];
S=Table[N[(1/n Sum[(A[[j]][[k]]-e[[j]])^2,{k,n}])^(1/2),10],{j,1,m}];
RSn=Sum[R[[j]]/S[[j]],{j,m}]/m;{Log[10,n]//N,Log[10,RSn]}]
```

Then we use the procedure to obtain the points (Log n, Log(R/S)) for n being the divisors of the "RightNumberOfData"
In[13]:=PointsSP500=Table[RescaledRange [DIVISORS[[i]],WeeklyDataSP500],
{i,7,Length[DIVISORS]-3}]
Out[13]=
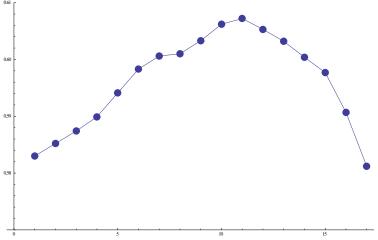{{0.8450980400142567,0.38848477370261403},{0.9030899869919433,0.4243192441315028},

{0.9542425094393248,0.4579212194613958},{1.,0.4917128445961558},...}

We check at which moment the graph consisting of the above determined points collapses. It allows us to determine the length of cycle.
In[14]:=AASP500=Table[Fit[Take[PointsSP500,k],{1,x},x],{k,23,Length[PointsSP500]}]
Out[14]={-0.092428+0.582974 x,-0.0951522+0.585208 x,-0.0978171+0.587372 x,...}

In[15]:=BBSP500=Table[Round[Coefficient[AASP500[[i]],x],0.0001],{i,1,Length[AASP500]}]
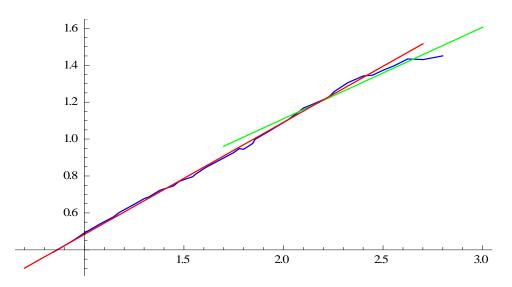Out[15]={0.583,0.5852,0.5874,...}

In[16]:=Show[ListPlot[BBSP500,PlotRange→{0.57,0.61},PlotStyle→PointSize[0.02]],
ListPlot[BBSP500,PlotRange→{0.57,0.61},Joined→True]]
Out[16]=

The highest value of the slope coefficient of the regression line was obtained for the period of 252 weeks, what gives a cycle equal to 4.84 years (assuming that one year consists of 52 weeks).
We can see, that starting from some moment the series becomes to be random (H tends to 0.5):
In[17]:=RANDOM=Fit[Take[PointsSP500,-16],{1,x},x]
Out[17]=0.118961+0.49557 x

We choose this part of the calculated points, which gives the highest value of H. Then we calculate the formula of regression line for these points and we generate the graph, which illustrates the obtained results. One can see here the change of rescaled range with respect to the chosen length of interval and one can also observe the calculated points arranging at first along the line "Hurst" and then along the random straight line.

In[18]:=Hurst= Fit[Take[PointsSP500,33],{1,x},x]
Show[ListPlot[PointsSP500, Joined→True],Plot[RANDOM,{x,1.7,3},PlotStyle→Green],
Plot[Hurst,{x,0.7,2.7}, PlotStyle→Red]]
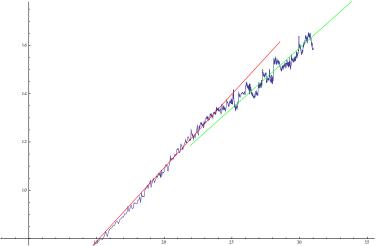Out[18]=-0.123699+0.607216 x

The slope of the line "Hurst", H_{emp}=0.607216 is just a value of the Hurst coefficient.

Note that the obtained straight lines determine quite well the path of points calculated for the consecutive n=7,8,9,…, 2520/2-1.
In[19]:=
PointsSP500WithoutDivisors=Table[RescaledRange[i, WeeklyDataSP500],
{i, 7, Floor[Length[WeeklyDataSP500]/2-1]}]
Out[19]= {{0.8450980400142567,0.38848477370261403}, … }

In[20]:= Show[ListPlot[PointsSP500WithoutDivisors,Joined→True,PlotStyle→{Blue}],
Plot[RANDOM,{x,2.3,3.5},PlotStyle→{Green}],Plot[Hurst,{x,1,2.7},PlotStyle→{Red}]]
Out[20]=

8. Results' testing of the rescaled range analysis with use of Purczynski's formula

Estimation of the expected value R/S according to J.Purczynski's formula (2000):

1) for n<30:

In[21]:=

$$P1[n\_] := ((n - 0.5)/2) * \text{Gamma}[(n - 1)/2]/(\text{Sqrt}[Pi]\,\text{Gamma}[n/2])\,\text{Sum}[\text{Sqrt}[(n - i)/i], \{i, 1, n - 1\}]//N$$

2) for n≥30:

In[22]:=

$$P2[n\_] :=$$
$$(((n - 0.5) * (n - 0.2491))/(n - 1)) * \text{Sqrt}[2/(n * Pi)] * (Pi/2 + (-2.4718 + 3.5466n - 1.4635 * n^2)/((n - 1)^2.5))//N$$

In[23]:= ListN=Table[DIVISORS[[i]],{i,7,Length[DIVISORS]-3}]
Out[23]={7,8,9,10,12,14,15,18,20,21,24,28,30,35,36,40,42,45,56,60,63,70,72,84,90,105,120, 126,140,168,180,210,252,280,315,360,420,504,630}
In[24]:=
PointsSP500TeoretPartI=Table[{Log[10,ListN[[i]]],Log[10,P1[ListN[[i]]]]},{i,1,12}]//N
Out[24]={{0.845098,0.349466},{0.90309,0.391034},{0.954243,0.426824},{1.,0.458209},...}

In[25]:= PointsSP500TeoretPartII=Table[{Log[10,ListN[[i]]],Log[10,P2[ListN[[i]]]]}, {i,12,Length[ListN]}]//N
Out[25]={{1.44716,0.74136},{1.47712,0.759219},{1.54407,0.798731},{1.5563,0.805897},...}
In[26]:= Fit[Join[PointsSP500TeoretPartI, PointsSP500TeoretPartII],{1,x},x]
Out[26]= $-0.0978632+0.573083x$

Hence the theoretical value of Hurst exponent is equal to H_{teor}=0.573083. From the other side we obtained the empirical value of the Hurst exponent to be H_{emp}=0.607216.

The result is significant, if the empirical H is larger by at least a square root of 1/N (where N assigns a number of returns being analyzed, here: N=Length[WeeklyDataSP500]) than the theoretical value. One can easily check that this is the case here:
In[27]:= Hemp-Hteor>=Sqrt[1/Length[WeeklyDataSP500]]//N
Out[27]=True

9. Conclusion

This study has shown the way to implement Computer Algebra System *Mathematica* in order to determine the value of the Hurst exponent, with use of the rescaled range analysis. The procedure "RescaledRange" allows determining the value of H efficiently and very fast, even for the big sets of starting data. For completeness, the author gave also the full description of data preparation including χ -squared test of data distribution as well as the results' significance testing.

10. References

[1] Anis, A.A. and Lloyd, E.H., *The expected value of the adjusted rescaled Hurst range of independent normal summands*, Biometrika **63,** 111−116, 1976.

[2] Feller, W., *The asymptotic distribution of the range of sums of independent random variables*, Ann. Math. Statist. Volume **22**, Number **3,** 427−432, 1951.

[3] Hurst, H.E., *Long-term storage capacity of reservoirs*, Transactions of American Society of Civil Engineers **116,** 770−799, 1951.

[4] Hurst, H.E., Black, R.P. and Simaika, Y.M., *Long-term storage: an experimental study Constable*, London, 1965.

[5] Jajuga, K. and Papla, D., *Teoria chaosu w analizie finansowych szeregow czasowych – aspekty teoretyczne i badania empiryczne*, Dynamiczne modele ekonometryczne: Materialy na VII Ogolnopolskie Seminarium Naukowe, 9-11 wrzesnia 1997, Torun. Wyd. Uniwersyte-tu im. M. Kopernika, Torun, 1997: http://www.econ1.uni.torun.pl/dme97/referaty97.htm .

[6] Mandelbrot, B.B., Van Ness, J.W., *Fractional Brownian Motion. Fractional noises and Applications*, SIAM Review, Vol. **10**, No. **4**, 422−437, October 1968.

[7] Mastalerz–Kodzis, A., *Modelowanie procesow na rynku kapitalowym za pomoca multifraktali*, Wyd. Akademii Ekonomicznej im. Karola Adamieckiego w Katowicach, Katowice, 2003.

[8] Peters, E.E., *Chaos and Order in the Capital Markets: a new view of cycles, prices and market volatility*, Wiley&Sons Inc, New York, 1991.

[9] Peters, E.E., *Fractal market analysis: applying chaos theory to investment and economy*, John Wiley & Sons, New York, 1994.

[10] Peters, E.E., *Teoria Chaosu a Rynki Kapitalowe: nowe spojrzenie na cykle, ceny i ryzyko*, WIG Press, Warszawa, 1997.

[11] Purczynski, J., *Wybrane problemy numeryczne stosowania analizy R/S*, "Przeglad Statystyczny" 2000 **R. XLVII**, Zeszyt 1−2 , 17−21, 2000.

[12] Rzeszotko, Z., *Fractal analysis of market indices: Estimating the Hurst exponent using rescaled range method*, Monografia naukowa: Computer Algebra Systems in Teaching and Research (Mathematical Modeling in Physics, Civil Engineering, Economics and Finance) 289−299, Wyd. Collegium Mazovia, Siedlce, 2011.

[13] Schuster, H.G., *Chaos deterministyczny*, PWN, Warszawa, 1995.

[14] Weron, A. and Weron, R., *Inzynieria finansowa: wycena instrumentow pochodnych, symulacje komputerowe, statystyka rynku*, WNT, Warszawa, 1998.

[15] Zawadzki, H., *Chaotyczne systemy dynamiczne*, Akademia Ekonomiczna, Katowice, 1996.

11. Supplementary Electronic Materials

[16] Rzeszotko Z., χ -squared test of data distribution – *Mathematica* notebook for **Section_4_5**.

[17] Rzeszotko Z., Preparation of the data, calculating the Hurst exponent and results testing with use of Purczynski's formula – *Mathematica* notebook for **Section_6_7_8**.