

**WYDZIAŁ  
MATEMATYKI  
I FIZYKI STOSOWANEJ**  
POLITECHNIKI RZESZOWSKIEJ

**Sylwia Szubart**

Narzędzia analizy zależności dalekosiężnych w szeregach  
czasowych

**Praca dyplomowa inżynierska**

Opiekun pracy:  
dr hab. inż. Dominik Strzałka

Rzeszów, 2023



# Spis treści

<b>1. Wstęp . . . . .</b>	<b>5</b>
<b>2. Zależności dalekosiężne . . . . .</b>	<b>7</b>
<b>3. Estymacja indeksu Hursta . . . . .</b>	<b>10</b>
3.1. Metodyka uzyskania danych . . . . .	10
3.2. Sposób obliczania indeksu H . . . . .	11
3.2.1. Metoda Beztrendowej Analizy Fluktuacyjnej . . . . .	11
3.2.2. Metoda Przeskalowanego Zasięgu . . . . .	12
3.3. Zastosowane metody rozwiązania . . . . .	13
3.3.1. Metoda Beztrendowej Analizy Fluktuacyjnej . . . . .	13
3.3.2. Metoda Przeskalowanego zasięgu . . . . .	17
3.4. Uzyskane wyniki . . . . .	21
3.5. Analiza uzyskanych wyników . . . . .	29
<b>4. Podsumowanie i wnioski końcowe . . . . .</b>	<b>31</b>
<b>Literatura . . . . .</b>	<b>33</b>



# 1. Wstęp

Odkrycie istnienia mikroświata zmieniło postrzeganie rzeczywistości wielu ludzi. Fakt możliwości jego obserwowania za pomocą mikroskopu fascynował nie tylko naukowców. Na początku XIX wieku, zainteresował się tym brytyjski botanik Robert Brown. Przyglądał się on zachowaniu pyłku kwiatowego w kropli wody. Początkowo myślał, że zaobserwowane drgania pyłku można traktować jako przejaw życia. Tezę tą obalił, gdy zauważył analogiczne zachowanie cząstek w przypadku materii nieorganicznej [1]. Wyjaśnieniem przyczyny takich chaotycznych ruchów jako pierwszy, w 1900 roku zajął się francuski matematyk Louis Bachelier [2]. Podczas analizy wyceny opcji na rynku spekulacyjnym zastanawiał się, czy przyrosty i spadki cen mają charakter losowy. Zauważył, że wykresy cen akcji wyglądają w przybliżeniu jak zaobserwowane przez Browna ruchy materii, w zależności od czasu. Bachelier stwierdził, że ceny na rynkach kapitałowych opisuje proces błędzenia losowego (inaczej standardowy ruch Browna), a wartość oczekiwana zmiany cen równa się zero. Pierwszy model matematyczny opisujący na czym polega ułamkowy ruch Browna, zaproponował Albert Einstein w pracy z 1905 roku [3]. Do takich samych wniosków, rok później doszedł Marian Smoluchowski, który prowadził niezależne badania eksperymentalnie od strony fizyki. Okazało się, że jest to najpowszechniejszy ruch występujący w przyrodzie. Z biegiem czasu, stwierdzono, że przyrosty w ruchu Browna opisywane przez rozkład normalny są niezależne i całkowicie losowe. W latach 50' Harold Edwin Hurst odkrył, że przyrosty poziomu wody w rzece Nil nie są niezależne, a zależności sięgają o wiele dłuższych przedziałów czasu, niż w przypadku zależności krótkoterminowych. Jego zdaniem był to przejaw istnienia długookresowej pamięci danych. Aby zbadać siłę zależności między nimi, opracował metodę nazwaną Rescaled Range (R/S), a samą siłę tych zależności określono jako wykładnik Hursta ( $H$ ). Z czasem zrozumiano, że ruch Browna jest szczególnym przypadkiem procesu Ułamkowego ruchu Browna i występuje, gdy  $H = 0.5$  [4].

Zasadniczym celem pracy było skupienie się na metodach estymacji współczynnika Hursta, w uproszczeniu wyznaczającego prawdopodobieństwo persystencji. Jako iż do dnia dzisiejszego nie istnieje w 100% dokładny sposób jego wyznaczania, skupiono się na pierwszej z opracowanych metod: "Przeskalowanego Zasięgu" (R/S) oraz zaproponowanej przez Peng et al. w 1994 roku metodzie "Beztrendowej Analizy Fluktuacyjnej" (DFA). Jako iż wszystkie aplikacje i programy wyznaczające indeks  $H$  metodą R/S

estymują wartości w niewielkiej liczby punktów, w tym przypadku, głównym celem było uzyskanie jak największej dokładności. Projekt został zrealizowany w programie Origin, który dzięki bardzo dużej różnorodności dostępnych wykresów, ułatwia analizę oraz wizualizację danych. Jest on produkowany przez OriginLab Corporation i działa w systemie Microsoft Windows. Ponadto Origin jest kompatybilny z językami programowania takimi jak: Python, R, LabView czy MATLAB i umożliwia tworzenie aplikacji z ich pomocą [5].

W ramach projektu należało przygotować serię rozwiązań w postaci zaimplementowanych funkcji pozwalających na analizę szeregów czasowych pod kątem występowania zależności dalekosiężnych znanych także jako samopodobieństwo statystyczne. Metody miały przetwarzać dane generując odpowiednie zbiory wyjściowe z charakterystykami oraz szacowaniem wartości indeksu samopodobieństwa  $H$ . Dodatkowym elementem było opracowanie rozwiązań w programie "Origin". Na potrzeby projektu wykorzystano generator ułamkowych ruchów Browna.

Praca składa się z rozdziału teoretycznego i praktycznego. Pierwszy rozdział zawiera opis tematyki, omówienie aktualnego stanu zagadnienia oraz grupy podziału szeregów w zależności od indeksu Hursta. Przedstawione i opisane zostały w nim również wybrane metody rozwiązania. Rozdział praktyczny został podzielony na podrozdziały. Pierwszy z nich zawiera metodę generacji danych. Podrozdział drugi i trzeci składa się z opisu zastosowanych metod oraz ich implementacji w programie "Origin". Część praktyczną kończy przedstawienie i analiza otrzymanych wyników.

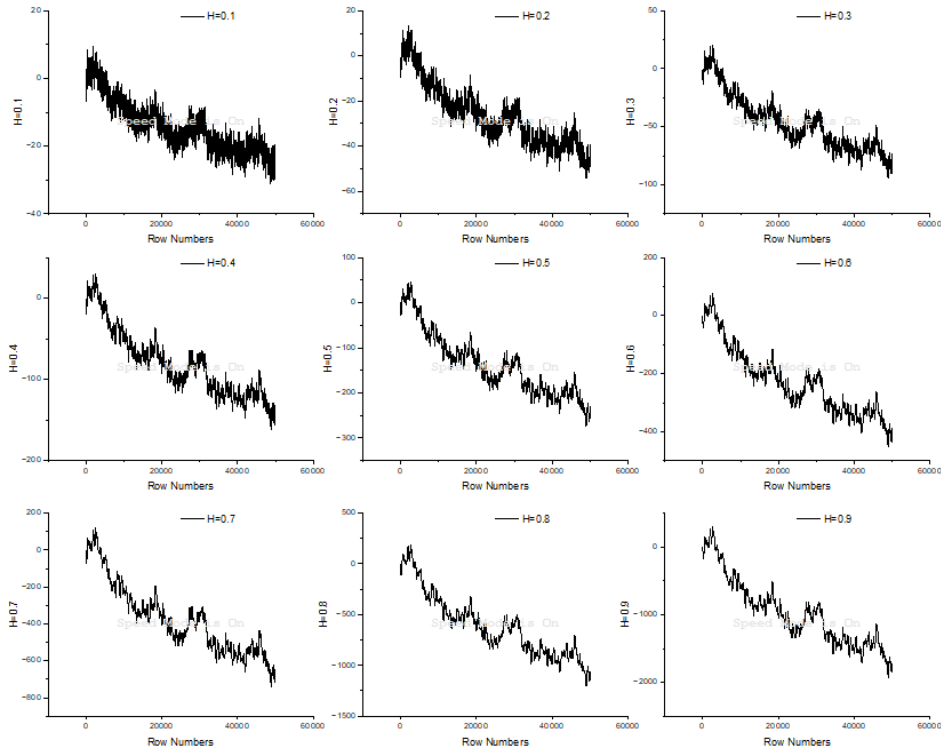
## 2. Zależności dalekosiężne

Zależności dalekosiężne odnoszą się do wzorców lub zależności, istniejących w dłuższym okresie czasu. Jedną z miar służących do ich wykrywania i analizy jest współczynnik Hursta. Określa on prawdopodobieństwo warunkowe tego, że kolejna zmiana procesu będzie zorientowana zgodnie z poprzednią [6]. Służy również do określania miary intensywności takich zależności. Na podstawie estymowanej wartości indeksu  $H$  można określić w jaki sposób wydarzenia z przeszłości korelują z przyszłością. Aktualnie jest wykorzystywany m.in. do przewidywania niestabilności generacji wiatrowych [8], opisu zachowań ludzkich czy też analizy szeregów stóp zwrotu na rynkach kapitałowych [7].

Istnieje podział na trzy grupy procesów w zależności od otrzymanej wartości  $H$  [9]:

- $H \in [0, 0.5)$  - szereg antypersystentny, charakteryzuje się dużą zmiennością. Jest określany jako "powracający do średniej" [10]. Występują ujemne korelacje między kolejnymi zmianami szeregu. Takie procesy niosą za sobą największe ryzyko.
- $H = 0.5$  - błądzenie losowe, brak korelacji. Proces jest całkowicie nieprzewidywalny. Inaczej mówiąc, żadne wydarzenia z przeszłości nie mają wpływu na przyszłość.
- $H \in (0.5, 1]$  - szereg persystentny, posiada pamięć długookresową (ang. long-term memory). Oznacza to, że występuje pozytywna korelacja między kolejnymi wyrazami szeregu, które wskazują na istnienie trendu. Dzięki temu możliwe są krótkoterminowe prognozy. Szeregom o takich własnościach towarzyszy najmniejsze ryzyko.

Wraz ze wzrostem wartości współczynnika Hursta, zwiększa się intensywność długozasięgowej zależności (ang. long-range dependence), a prawdopodobieństwo zmiany kierunku spada. Jak można zauważyć na rys. 2.1 im wyższe  $H$ , tym wykresy są bardziej "wygładzone".



Rysunek 2.1: Wykresy z różnymi wartościami indeksu  $H$

Hurst zajął się badaniem zależności dalekosiężnych zjawisk naturalnych takich jak wylewy rzek, temperatury, opady czy nawet plamy słoneczne. W każdym przypadku wykładnik  $H$  był większy od 0.5. Oznacza to, że ruch Browna nie sprawdza się w przyrodzie, a nawet bardzo odległe w czasie wydarzenia mogą pozostawiać piętno na długie lata. Ponadto, możliwe jest wyznaczenie długości cykli, czyli moment zaniku pamięci układu. Takie cykliczne procesy rzadko zmieniają kierunek trendu, można zatem powiedzieć, że są persystentne, więc dla  $n$  krótszych od średniej długości cyklu,  $H$  będzie większe niż 0.5. Natomiast spadek wartości wykładnika Hursta dla wartości  $N^*$  do co najmniej 0.5 będzie oznaczać zanik pamięci długookresowej. Przeciętną długość cyklu wyznacza wartość  $N^*$  dla której występuje ta zmiana [11].

Istnieje wiele metod pozwalających na wyznaczenie indeksu  $H$ , takich jak: Whittle estimator, Periodogram, Variance, Variance of Residuals (znana również pod nazwą Detrended Fluctuation Analysis) czy Rescaled Range. Żadna z tych metod nie jest jednak dokładna i jedynie przybliża wartość współczynnika  $H$ .



Najczęściej używaną i jednocześnie najstarszą zaproponowaną metodą estymacji  $H$  jest metoda Przeskalowanego Zasięgu (ang. Rescaled Range) oznaczana w skrócie R/S. Opracował ją sam Harold Edwin Hurst. Wyznaczenie estymowanej wartości indeksu, sprowadza się do:

- 1) Podzielenia wyjściowego szeregu na podszeregi z  $n$  obserwacjami.
- 2) Zcentrowania podszeregu poprzez odjęcie od każdej jego wartości średniej arytmetycznej.
- 3) Wyznaczenia skumulowanych szeregów czasowych.
- 4) Obliczenia zakresów skumulowanych szeregów przez odjęcie minimalnej wartości od maksymalnej.
- 5) Po podzieleniu zakresu przez odchylenie standardowe szeregu otrzymana wartość będzie przeskalowanym zasięgiem.

Wartość współczynnika Hursta określa kąt nachylenia prostej regresji otrzymanej po zlogarytmowaniu przeskalowanego zasięgu i liczby obserwacji.

Drugą równie popularną metodą estymacji  $H$  jest metoda Beztrendowej Analizy Fluktuacyjnej (DFA). Została ona opracowana pod koniec XX wieku. Największymi zwolennikami tej metody są fizycy. Sprowadza się ona do:

- 1) Zcentrowania podszeregu poprzez odjęcie od każdej jego wartości średniej arytmetycznej.
- 2) Wyznaczenia skumulowanego szeregu czasowego.
- 3) Zdefiniowania liczby elementów w przedziałach.
- 4) Podzieleniu sygnału na segmenty, usunięciu linii trendu i obliczeniu średniego błędu kwadratowego.
- 5) Powtórzeniu powyższego kroku dla każdego przedziału.

Wartość  $H$  będzie połową kąta nachylenia prostej aproksymującej logarytm z rozmiaru segmentu do logarytmu z średniego błędu kwadratowego.

Obie wyżej przedstawione metody będą szerzej omawiane w kolejnym rozdziale.

### 3. Estymacja indeksu Hursta

#### 3.1. Metodyka uzyskania danych

Zbiory danych, użyte w kolejnych rozdziałach do porównania obliczonych różnymi sposobami wartości współczynnika Hursta, zostały uzyskane poprzez generator ułamkowych ruchów Browna - tsfBm. Oprogramowanie jest dostępne na stronie autora - Johna Conovera [13].

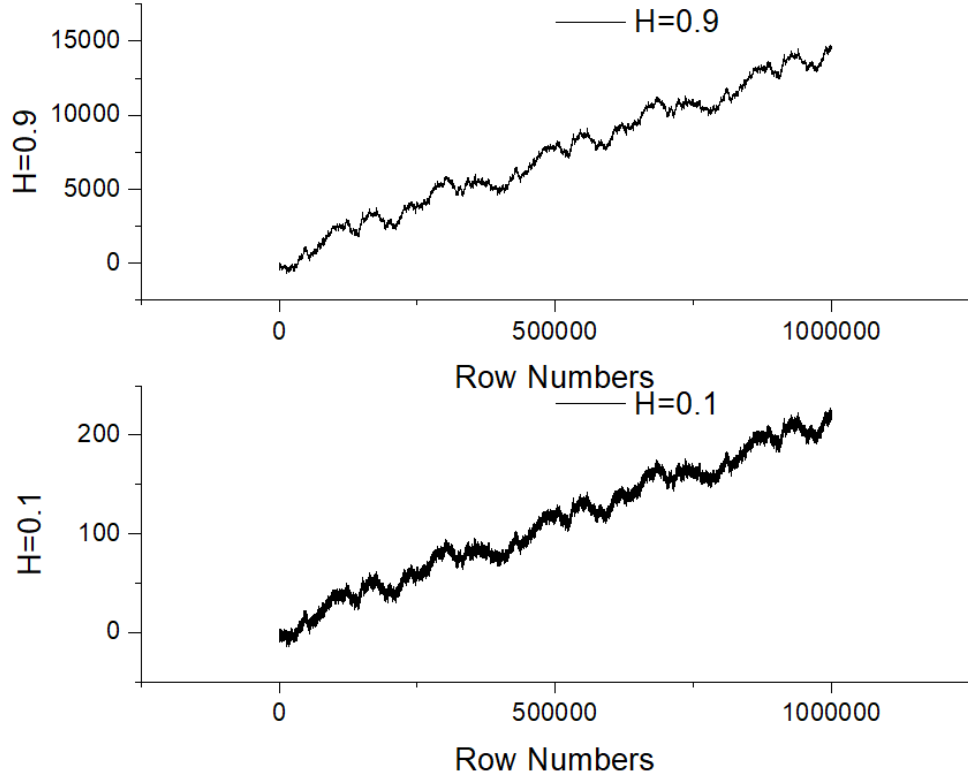
Opracowana przez niego metoda, pozwala na tworzenie zbiorów danych o dowolnej długości, z wybraną wartością współczynnika Hursta. Przykładowe wywołania procedury z poziomu wiersza poleceń pokazano poniżej.

```
tsfBm -h 0.1 -i 2 -m 100 1000000 > h0.1_02.txt
```

```
tsfBm -h 0.9 -i 2 -m 100 1000000 > h0.9_02.txt
```

W tym przypadku wygenerowany szereg składający się z 1 000 000 elementów, będzie miał indeks  $H$  równy odpowiednio 0.1 i 0.9, a głębokość zależności długoterminowych, za które odpowiada parametr  $m$  w obu przypadkach będzie sięgać 100 elementów. Dodatkowo, zmiana parametru  $i$  gwarantuje zmianę punktu startowego zbioru. Na rys. 3.2 przedstawiono wykresy otrzymane z wygenerowanych danych.

W tej pracy, z powodu wysokiej złożoności obliczeniowej zaimplementowanych metod, do testów wygenerowano szeregi liczące 50 000 elementów.



Rysunek 3.2: Wykresy dla wygenerowanych danych z  $H$  równym 0.1 i 0.9

## 3.2. Sposób obliczania indeksu $H$

Do obliczenia współczynnika Hursta wykorzystano dwie metody: Beztrendowej Analizy Fluktuacyjnej (DFA) i Przeskalowanego Zasięgu (R/S), które są uznawane za najdokładniejsze [14].

### 3.2.1. Metoda Beztrendowej Analizy Fluktuacyjnej

Wynaleziona w 1994 roku przez Peng et al. w celu udowodnienia istnienia pamięci długoterminowej w sekwencjach kwasu deoksyrybonukleinowego (DNA). Kroki obliczania [15]:

- 1) Obliczenie średniej (3.1)

$$m = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.1)$$

- 2) Obliczenie sumy bieżącej  $y_k$  (3.2)

$$y_k = \sum_{i=1}^k (x_i - m) \quad (3.2)$$

- 3) Podzielenie  $y_k$ , dla  $k = 1, 2, \dots, n$  na przedziały  $X_r$  dla  $r = 1, 2, \dots, n/v$ .

$$X_1 = (y_1, y_2, \dots, y_v), X_2 = (y_{v+1}, y_{v+2}, \dots, y_{2v}) \dots, X_{n/v} = (y_{n-v+1}, y_{n-v+2}, \dots, y_n)$$

- 4) Wyznaczenie prostej metodą najmniejszych kwadratów, która najlepiej przybliża wartości każdego  $X_r$ . Niech  $y_v(t)$  będzie rzędną dla czasu  $t$ . Będzie to lokalny trend  $X_r$ ,  $r = t/v + 1$ .
- 5) Po usunięciu trendu z sumy bieżącej można obliczyć średni błąd kwadratowy (ang. root mean square, RMS) (3.3)

$$F(v) = \sqrt{\frac{1}{n} \sum_{t=1}^n [y_t - y_v(t)]^2} \quad (3.3)$$

- 6) Po stworzeniu wykresu podwójnie logarytmicznego zależności  $F(v)$  od rozmiaru segmentu  $n$  ( $\log F(n) - \log n$ ), wykładnik Hursta jest połową wartości nachylenia prostej  $F(v)$ .

### 3.2.2. Metoda Przeskalowanego Zasięgu

Metoda R/S (ang. Recaled Range Analysis) została zaproponowana przez Harolda E. Hurst w latach '50, a później zmodyfikowana przez E. Petersa. Procedurę pozwalającą na wyznaczenie współczynnika Hursta można podsumować w kilku krokach [16]:

- 1) Podzielenie zbioru  $R$  na  $d$  przedziałów o długości  $n$ .
- 2) Obliczenie średniej ( $E_m$ ) i odchylenia standardowego ( $S_m$ ) dla każdego przedziału  $m = 1, 2, \dots, d$ .
- 3) Znormalizowanie danych dla  $i = 1, 2, \dots, n$  (3.4).

$$Z_{i,m} = R_{i,m} - E_m \quad (3.4)$$

- 4) Stworzenie szeregu czasowego z sum bieżących  $Y_{i,m}$  dla  $i = 1, 2, \dots, n$  (3.5).

$$Y_{i,m} = \sum_{j=1}^i Z_{j,m} \quad (3.5)$$

- 5) Obliczenie różnicy między największym a najmniejszym elementem szeregu  $Y_{i,m}$  (3.7).

$$R_m = \max[Y_{1,m}, Y_{2,m}, \dots, Y_{i,m}] - \min[Y_{1,m}, Y_{2,m}, \dots, Y_{i,m}] \quad (3.6)$$

- 6) Przeskalowanie zakresu przez podzielenie  $R_m$  przez  $S_m$ . Po uśrednieniu całego zbioru przedziałów  $d$ , średnia wartość przeskalowanego zakresu dla przedziałów o długości  $n$  jest obliczana według wzoru (3.7).

$$\left(\frac{R}{S}\right)_n = \frac{1}{d} \sum_{m=1}^d \frac{R_m}{S_m} \quad (3.7)$$

- 7) Indeks Hursta jest przybliżany przez kąt nachylenia prostej estymującej wykres  $\log \frac{R}{S} - \log n$ .

### 3.3. Zastosowane metody rozwiązania

Jako iż OriginPro jest kompatybilny z językiem programowania Python, to za jego pomocą zostały stworzone funkcje umożliwiające obliczanie wykładnika Hursta zarówno metodą Beztrendowej Analizy Fluktuacyjnej (ang. Detrended Fluctuation analysis, DFA), jak i Przeskalowanego zasięgu (ang. Rescaled Range, R/S). Funkcje wykorzystują pakiety znajdujące się na Listingu 1.

```
1 import numpy as np
2 import math as math
3 import originpro as op
4
```

Listing 1: Wykorzystane pakiety

#### 3.3.1. Metoda Beztrendowej Analizy Fluktuacyjnej

Metoda wymagała jednej funkcji pomocniczej obliczającej średni błąd kwadratowy RMS (średni błąd kwadratowy, ang. Root Mean Square) - Listing 2.

Funkcja za argumenty przyjmuje trzy parametry:

- tv: ilość danych w przedziale,
- numv: liczba przedziałów,
- yk: szereg.

Dla każdego przedziału wyznaczono metodą najmniejszych kwadratów wzór funkcji liniowej. Następnie wyliczono wartości tej funkcji w punktach  $x$ , które odpowiadają za indeksy danych zawartych w konkretnym przedziale (dodając 1) i odjęto ją od  $xr$ , czyli wartości danego szeregu o indeksach  $x - 1$ . Różnica ta została podniesiona do kwadratu, a następnie zsumowana i podzielona przez ilość danych w zbiorze. Na koniec z tej liczby wyciągnięto pierwiastek drugiego stopnia.

```

1 def RMS(tv,numv,yk):
2     nn=np.size(yk)
3     st=[]
4     #dla kazdego przedzialu
5     for nv in range(numv):
6         #wyznaczamy dane
7         xr= yk[nv*tv:((nv+1)*tv)]
8         x=np.arange(nv*tv+1,(nv+1)*tv+1)
9         #metoda najmniejszych kwadratow (linia zbiegajaca do
10        przedzialu)
11        pv=np.polyfit(x,xr,1)
12        #wzor funkcji liniowej
13        fv=np.poly1d(pv)
14        #wyliczenie wartosci funkcji dla x
15        ord=fv(x)
16        st=np.append(st, (ord - xr)**2)
17    wartosc=np.cumsum(st)
18    rms=(wartosc[np.size(wartosc)-1]*1/nn)** .5
19    return rms

```

Listing 2: Definicja funkcji RMS

Funkcja "DFA" za parametr przyjmuje wektor (zbiór danych). Sprowadza się ona do:

- 1) Obliczenia sumy bieżącej.
- 2) Wyznaczenia liczby danych w przedziałach za pomocą wzoru (3.8).

$$4 * (2^{1/8})^s + 1/2 \quad (3.8)$$

Obliczenia są wykonywane do momentu, gdy długość zbioru wejściowego podzielona przez 4 jest większa lub równa wartości funkcji (3.8). Za  $s$  przyjmowane są wartości od 11 do  $\frac{1}{4}$  długości zbioru.

- 3) Stworzenia listy przyjmującej wartości funkcji "RMS". Przekazywanymi argumentami są kolejno: wyznaczone ilości danych w przedziałach, długość wektora wejściowego podzielona przez wyznaczone ilości danych w przedziałach oraz wektor sumy bieżącej.
- 4) Wygenerowania z użyciem metody najmniejszych kwadratów prostej aproksymującej log z wyznaczonych ilości danych w przedziałach do log z listy przyjmującej wartości funkcji "RMS".

Wykładnik Hursta jest połową wartości nachylenia prostej aproksymującej.

W dalszej części funkcji zostały usunięte z aktywnego arkusza wartości z kolumn [1,6] oraz dodane cztery kolumny. Na koniec uzupełniono kolumny danymi oraz dodano wykresy. Definicja funkcji znajduje się na Listingu 3.

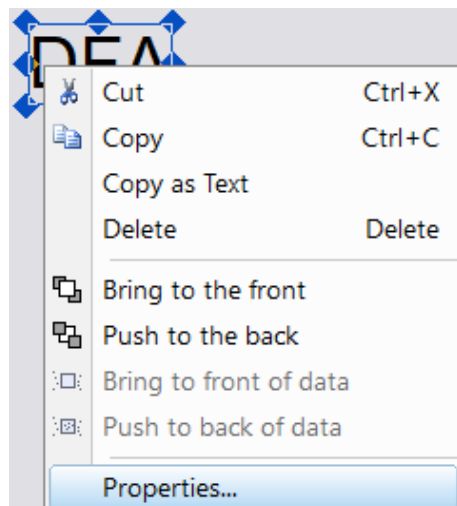
```

1  def DFA(xs):
2      n=len(xs)
3      m=np.mean(xs)          #srednia
4      #suma biezaca
5      xsm=xs-m
6      sums=xsm[0]
7      yk=[]
8      for i in range(n):
9          yk.append(sums)
10         sums+=xsm[i]
11     sugp=[]
12     for s in range(11,int(n/4)):
13         if int(4*(2**(1/8))**s+0.5)<=int(n/4):
14             sugp=np.append(sugp,int(4*(2**(1/8))**s+0.5))
15         else:
16             break
17     #wartosci Fn
18     Fn=[]
19     for tv in sugp:
20         Fn=np.append(Fn,RMS(int(tv),int(n/tv),yk))
21
22     prlog=np.polyfit(np.log10(sugp),np.log10(Fn),1)
23     #wzor funkcji liniowej
24     flog=np.poly1d(prlog)
25     st_flog=str(object=flog)
26     logx=np.log10(sugp)
27     logy=flog(np.log10(sugp))
28     logF=np.log10(Fn)
29     hurst=[]
30     hurst.append(round(flog[1]/2,3))
31     wks = op.find_sheet('w')
32     wks.del_col(1,6)
33     a=wks._check_add_cols(4,2)
34     #uzupelnianie kolumn danymi
35     h=wks.from_list(1,'Hurst','Hurst exponent')
36     hu=wks.from_list(2,hurst)
37     l=wks.from_list(3,logx,'Data scale (log)')
38     ly=wks.from_list(4,logy,'Fit')
39     f=wks.from_list(5,logF,'RMS (log)')
40     g1 = op.new_graph('Linear fit')[0]
41     dp = g1.add_plot(wks,'F','D',type='s')
42     g2 = op.new_graph('Data')[0]
43     d2 = g2.add_plot(wks,'A','#')
44     d=g1.add_plot(wks,'E','D',type='l')
45     d.color='#FA3C3C'
46     g1.rescale()
47     g2.rescale()
48     return logx
49

```

Listing 3: Funkcja "DFA"

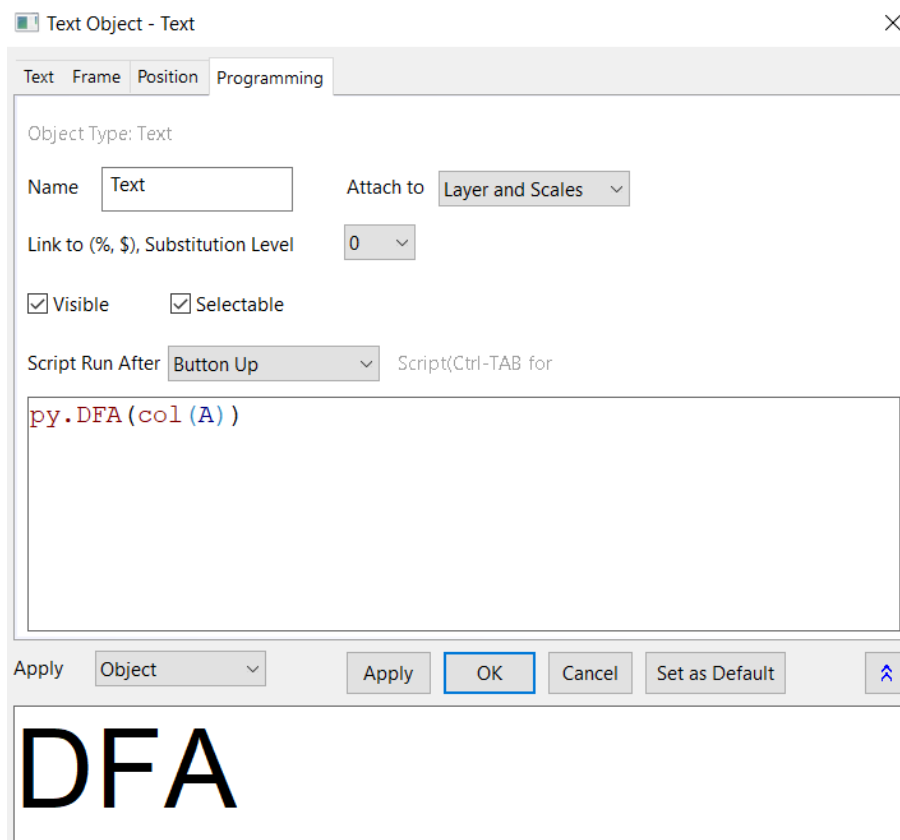
Obie funkcje razem z bibliotekami zostały wprowadzone do pliku "labtalk.py". W przypadku pominięcia tego kroku, dalsza część programu będzie zwracać błędy, ponieważ Origin nie będzie rozpoznawał tych funkcji. Po dodaniu pola tekstowego, nazwaniu go analogicznie do wykorzystanej metody (DFA) i kliknięciu prawego przycisku myszy wybrano opcję "Properties..." jak na rys. 3.3.



Rysunek 3.3: Właściwości pola tekstowego

We właściwościach wybrano zakładkę "Programming". Z tego poziomu można dodać opcję uruchamiania skryptu po konkretnej akcji. W tym przypadku po kliknięciu przycisku uruchomi się funkcja "DFA", która za dane wejściowe (szereg czasowy) przyjmie pierwszą kolumnę (A) arkusza, tak jak na rys. 3.4.





Rysunek 3.4: Programowanie przycisku

### 3.3.2. Metoda Przeskalowanego zasięgu

Funkcję obliczającą indeks Hursta metodą R/S zaczerpnięto z programu SELFIS [17], który został napisany w języku programowania JAVA. Przekonwertowano ją na język Python, a następnie zmodyfikowano, tak aby estymowała wartość dla każdego punktu oraz działała dla dowolnie długiego szeregu. Do obliczenia wartości  $H$ , w tej metodzie wykorzystano funkcje pomocnicze, które za parametry wejściowe przyjmują wektor:

- 1) Avg (średnia) - Listing 4, liczona za pomocą wzoru  $m = \frac{1}{n} \sum_{i=1}^n x_i$ , gdzie  $n$  to długość wektora wejściowego, a  $x$  - element.

```

1 def avg(paramvector):
2     d=0.0
3     b=0
4     while(b<len(paramvector)):
5         double_=paramvector[b]
6         d+=double_
7         b+=1
8     return d/len(paramvector)
9

```

Listing 4: Definicja funkcji "avg"

- 2) RS - odpowiada za normalizację danych, stworzenie szeregu z sum bieżących i obliczenie różnicy między najmniejszym, a największym elementem zbioru (jak na listingu 5).

```

1 def RS(paramvector):
2     d1=avg(paramvector)
3     double_=paramvector[0]
4     d2=float(double_)
5     d3=d2
6     d4=0.0
7     b=0
8     while (b<len(paramvector)):
9         double_=paramvector[b]
10        d4+=float(double_)-d1
11        if (d4> d2):
12            d2=d4
13        elif (d4<d3):
14            d3=d4
15        b+=1
16    return d2-d3

```

Listing 5: Definicja funkcji "RS"

- 3) Var - wyznaczenie wariancji danego podzbioru (Listing 6).

```

1 def var (paramvector):
2     d1=avg(paramvector)
3     d2=0.0
4     d3=0.0
5     b=0
6     while (b<len(paramvector)):
7         double_=paramvector[b]
8         d2+=math.pow(float(double_-d1),2.0)
9         b+=1
10    if (len(paramvector)>1):
11        d3=d2/(len(paramvector)-1)
12    return d3

```

Listing 6: Definicja funkcji "var"

Funkcja główna "RS Plot", której definicja jest zawarta na Listingu 7, w skrócie polega na wyznaczeniu ilości danych w przedziałach (w tym przypadku podzbiory będą liczyć od 4 elementów do podłogi długości wektora wejściowego podzielonego przez dwa). Następnie obliczana jest liczba przedziałów i wyznaczane są ich elementy. Dla tych fragmentów danych policzono kolejno, za pomocą funkcji "RS" różnicę między najmniejszym, a największym elementem podzbioru oraz za pomocą funkcji "var" pierwiastek z wariancji czyli odchylenie standardowe. Następnie stworzono wektory do wykresów logarytmicznych oraz prostą estymującą i wyznaczono wartość wykładnika Hursta ( $H$ ). Na koniec dodano kolumny i usunięto z nich ewentualne dane (jeżeli istniały), wygenerowano wykresy i wyświetlono dane w arkuszu.

```

1 def RS_Plot (paramvector):
2     i=int(math.floor(len(paramvector)/2))
3     j=4
4     vector=[]
5     b=0
6     while (j<=i):
7         vector.append(j)
8         j+=1
9     vector1=[]
10
11     b=0
12     while ( b< len(vector)):
13         k=int(vector[b])
14         m=len(paramvector)/k
15         b1=0
16         n=0
17         vector2=[]
18         while(b1<math.floor(m)):
19             n=b1*k
20             vector3=[]
21             while ((n<(b1+1)*k) and n<=len(paramvector)):
22                 vector3.append(paramvector[n])
23                 n+=1
24             d1=RS(vector3)
25             d2=var(vector3)
26             d3=0.0
27             if (d2>0.0):
28                 d3=d1/math.sqrt(d2)
29             else:
30                 d3=((len(vector3)-1)/math.sqrt(len(vector3)))
31             vector2.append(d3)
32             b1+=1
33         vector1.append(avg(vector2))
34         b+=1
35     b=0
36     logY=[]
37     logX=[]
38     logF=[]
39     while (b<len(vector1)):
40         d1=float(vector1[b])

```

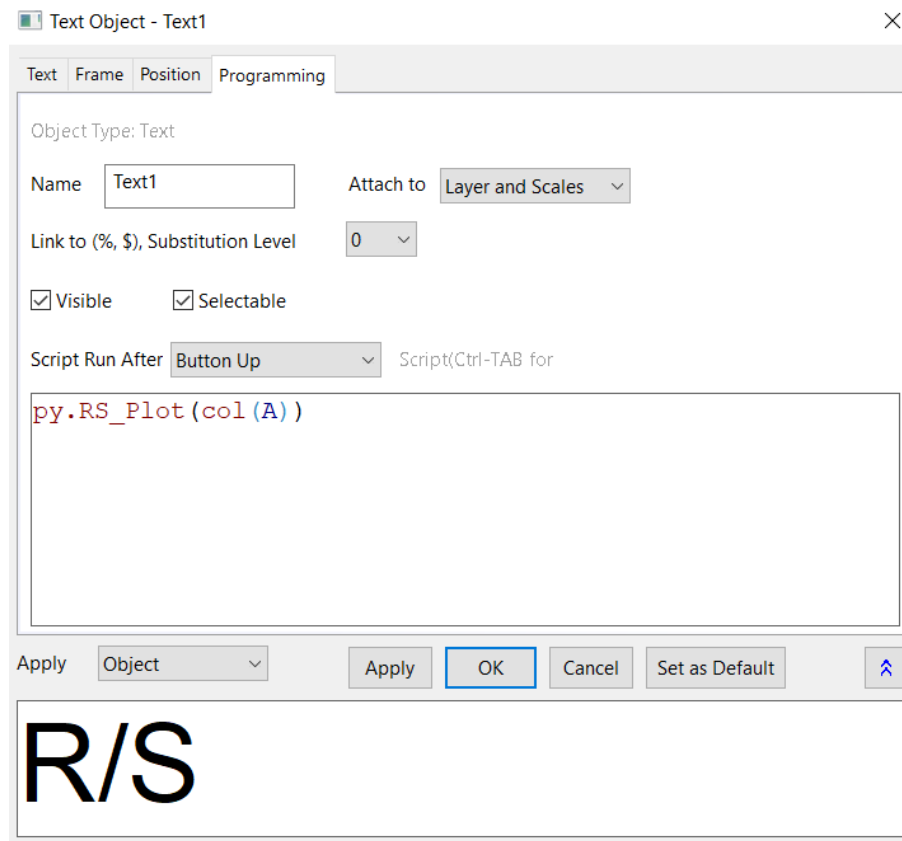
```

41     k=(vector[b])
42     d2=math.log(d1)
43     logY.append(d2)
44     d3=math.log(k)
45     logX.append(d3)
46     b+=1
47 prlog=np.polyfit(logX,logY,1)
48 #wzor funkcji liniowej
49 flog=np.poly1d(prlog)
50 logF=flog(logX)
51 d1=0.0
52 d2=0.0
53 d3=0.0
54 d4=0.0
55 b=0
56 while (b<len(logX)):
57     double_1=logX[b]
58     double_2=logY[b]
59     d9=double_1
60     d10=double_2
61     d1+=d9
62     d2+=d10
63     d3+=d9*d9
64     d4+=d9*d10
65     b+=1
66 d6=(d4-d1*d2/len(logX))/(d3-d1*d1/len(logX))
67 h=[]
68 h.append(d6)
69
70 wks = op.find_sheet('w')
71 wks.del_col(1,6)
72 a=wks._check_add_cols(4,2)
73 hh=wks.from_list(1,'Hurst','Hurst exponent')
74 hu=wks.from_list(2,h)
75 l=wks.from_list(3,logX,'Log(x)')
76 ly=wks.from_list(4,logY,'Log(y)')
77 l2=wks.from_list(5,logF,'Fit')
78 gl = op.new_graph('Linear fit')[0]
79 dp = gl.add_plot(wks,'E','D',type='s')
80 dd = gl.add_plot(wks,'F','D',type='l')
81 dd.color='#FA3C3C'
82 dp.symbol_size=3
83 g2 = op.new_graph('Data')[0]
84 d2 = g2.add_plot(wks,'A','#')
85 gl.rescale()
86 g2.rescale()
87 return h

```

Listing 7: Definicja funkcji "RS\_Plot"

Tak jak poprzednio, funkcje zostały dodane do pliku "labtalk.py". Aby możliwa była aktywacja funkcji przez przycisk, do pola tekstowego dodano we właściwościach uruchomienie funkcji "RS Plot" po kliknięciu (pokazane na rys. 3.5).

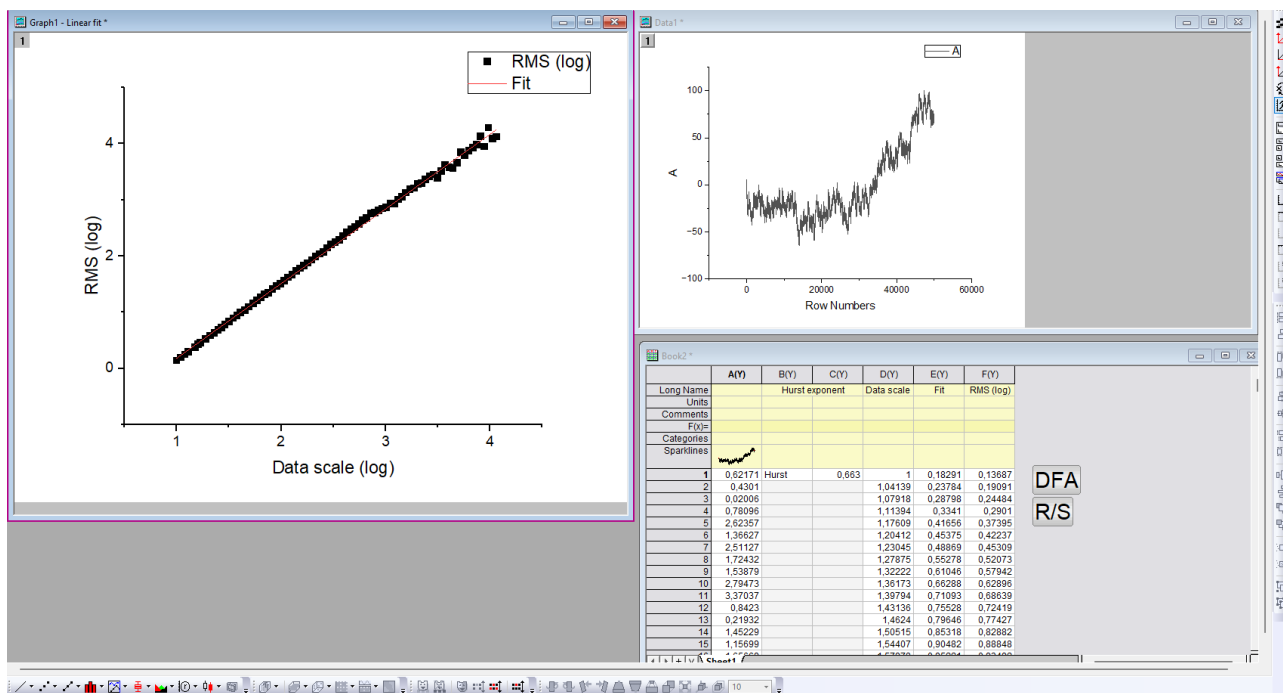


Rysunek 3.5: Programowanie przycisku R/S

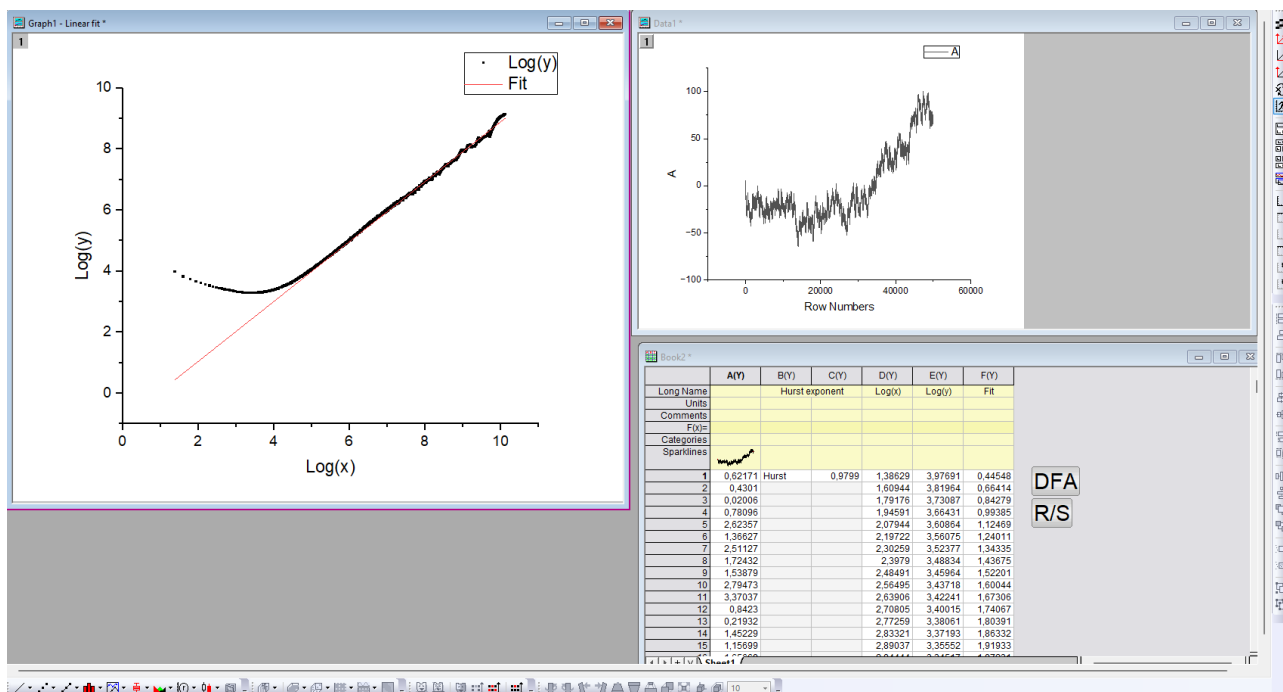
### 3.4. Uzyskane wyniki

Po wprowadzeniu danych z różnymi wartościami współczynnika Hursta, wygenerowanych przez program "tsfBm" do pierwszej kolumny, obliczano indeks  $H$  za pomocą metody DFA i R/S. W przypadku metody DFA program zwraca również wartości  $\log n$  oznaczanego jako "Data Scale",  $\log F(n)$  nazwany w programie "Origin" "RMS (log)", a także dopasowanie liniowe "Fit". Dla metody R/S, oprócz wartości współczynnika Hursta, zwracane do kolumn są wartości  $\log \frac{R}{S}$  ("log(x)"),  $\log n$  nazwane "log(y)", oraz dopasowanie liniowe "Fit". Wykresy tworzone są z wartości w powyżej opisanych kolumnach, z czego czerwony wykres jest estymowaną prostą. Dla metody R/S uzyskano wyniki zarówno w przypadku estymacji wartości w każdym punkcie, jak i dla elementów  $2^n$ . Poniżej (na rys. 3.6 - 3.17) przedstawiono przykładowe wywołania funkcji, dla danych o współczynniku Hursta równym:

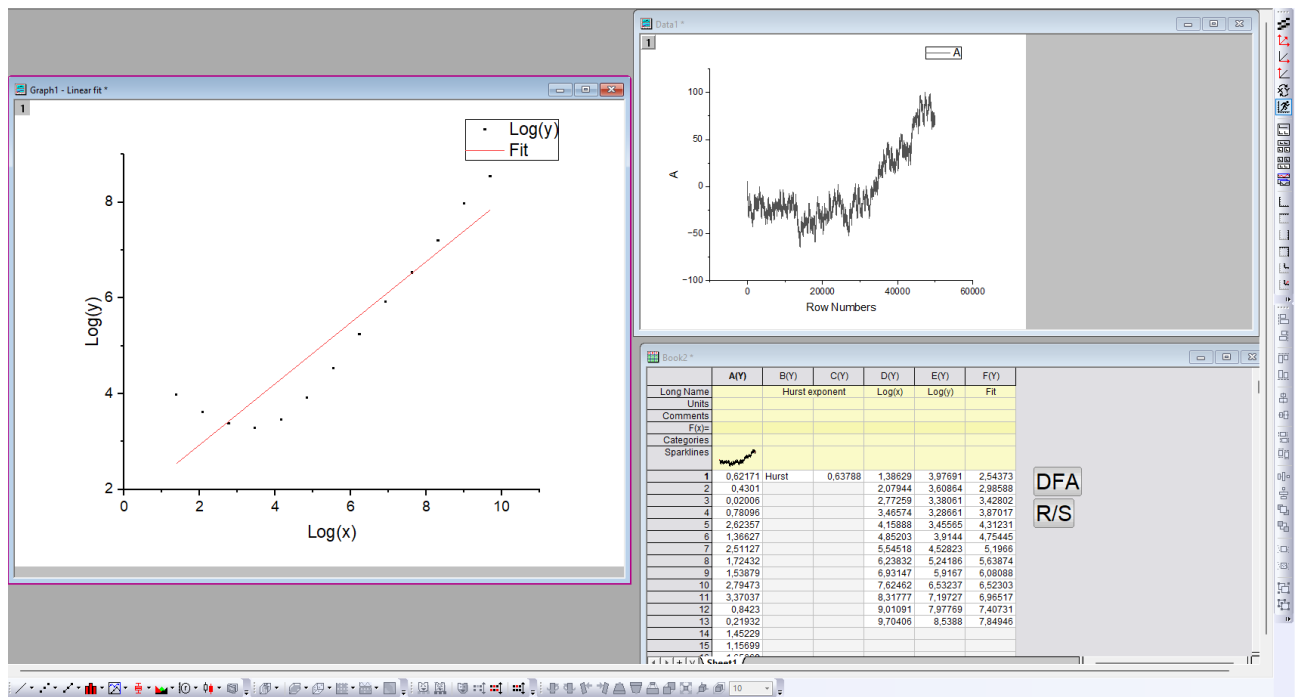
- 1)  $H = 0.4$  - rys. 3.6- 3.8.



Rysunek 3.6: Wyniki uzyskane metodą DFA

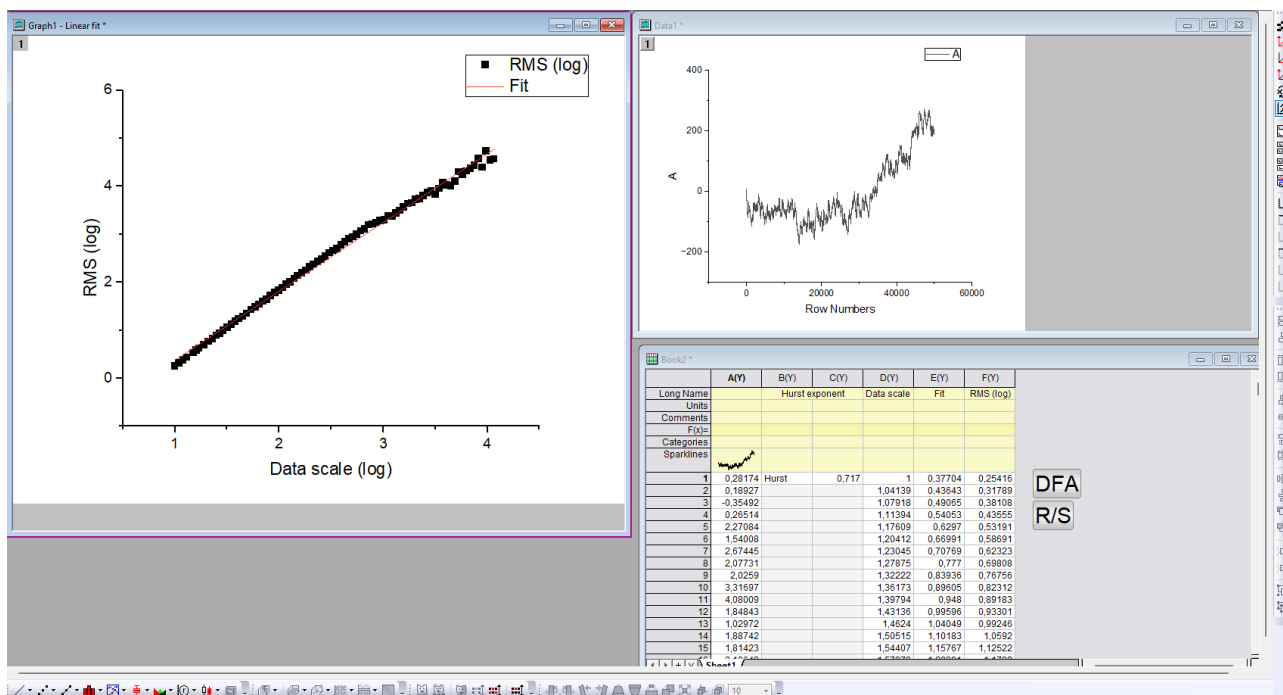


Rysunek 3.7: Wyniki uzyskane metodą R/S z estymacją wartości w każdym punkcie

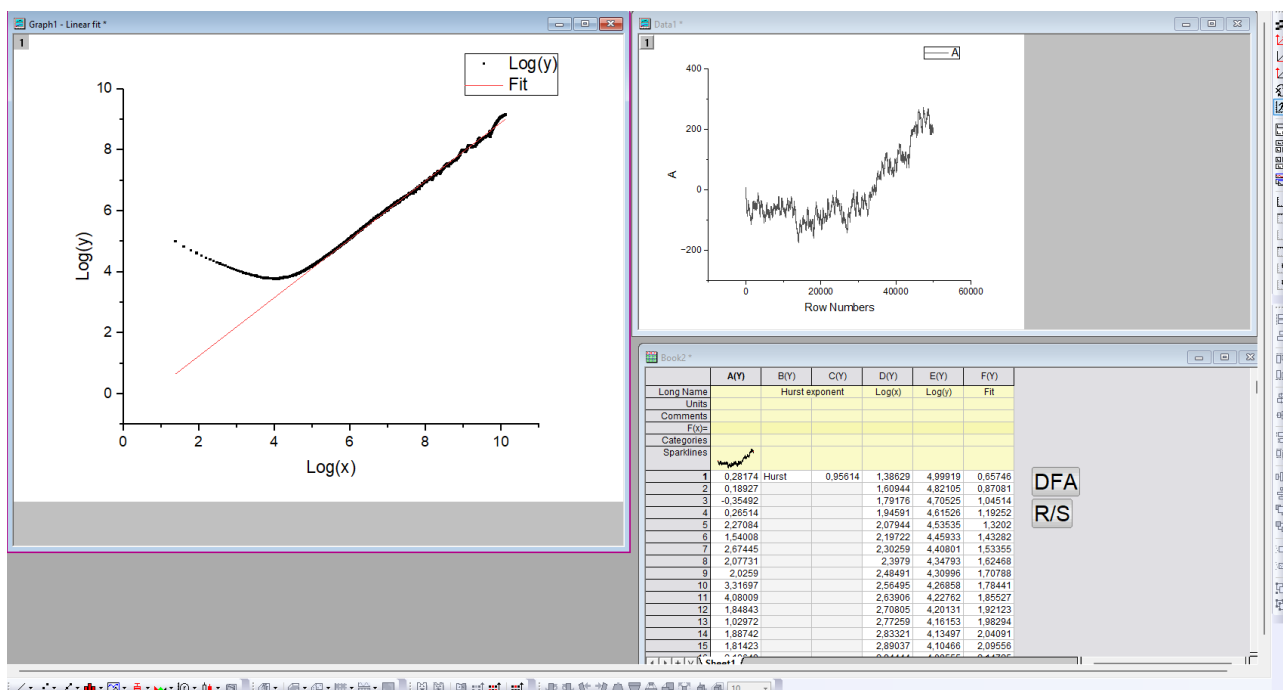


Rysunek 3.8: Wyniki uzyskane metodą R/S z estymacją wartości w punktach  $2^n$

2)  $H = 0.6$  - rys. 3.9- 3.11.

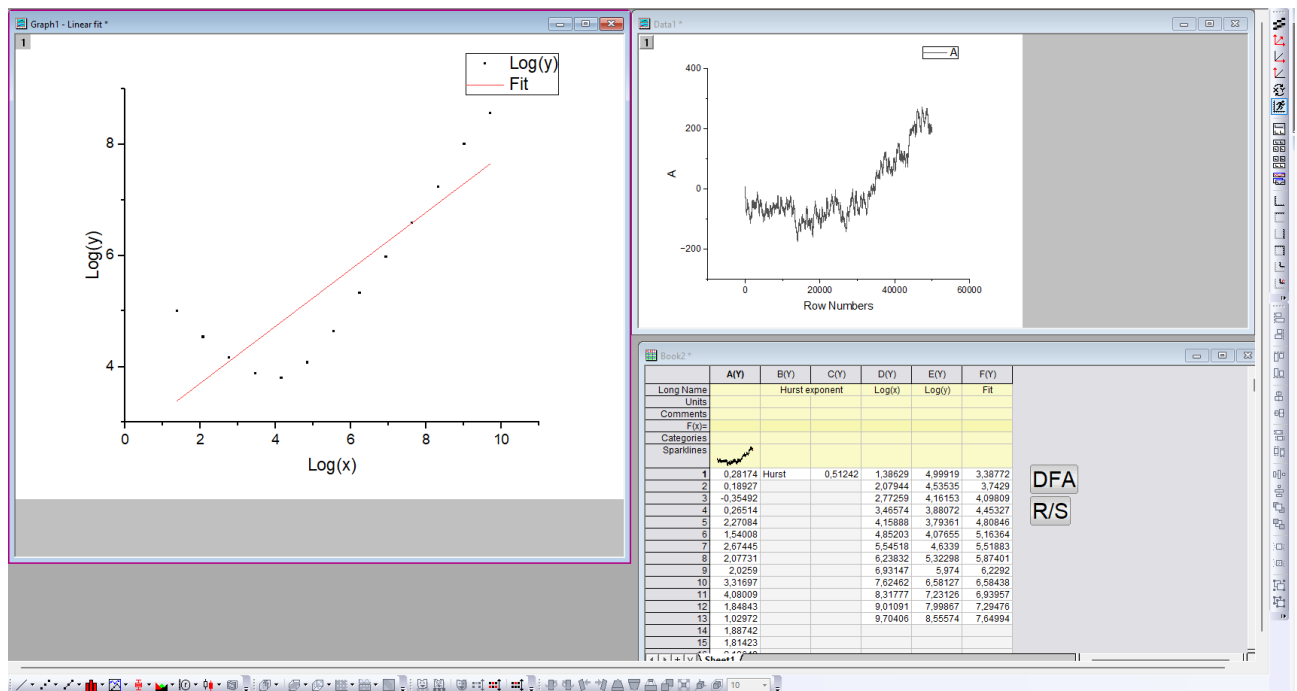


Rysunek 3.9: Wyniki uzyskane metodą DFA



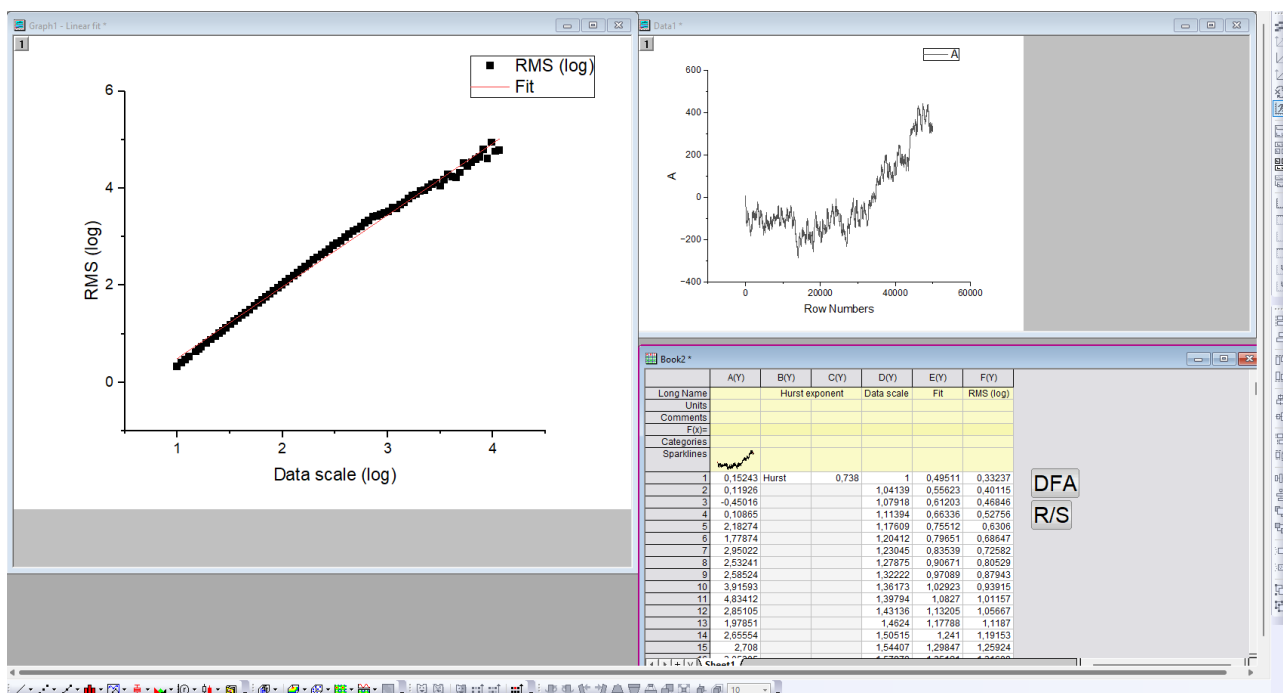
Rysunek 3.10: Wyniki uzyskane metodą R/S z estymacją wartości w każdym punkcie



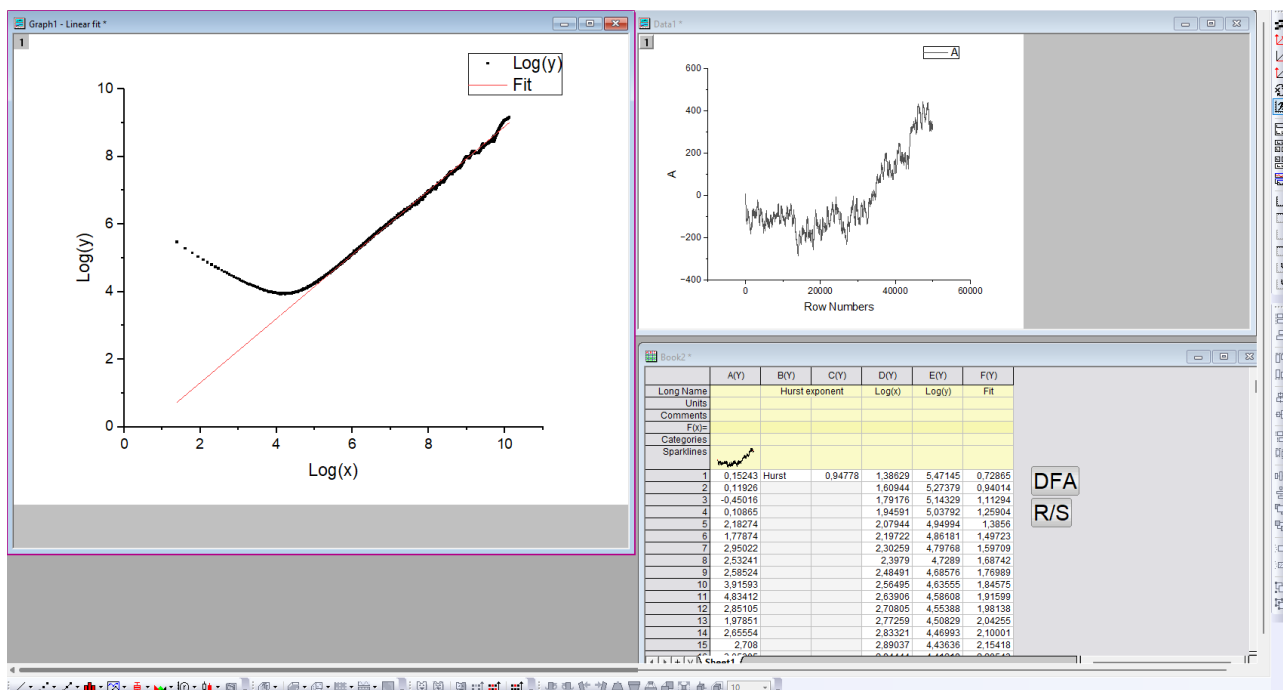


Rysunek 3.11: Wyniki uzyskane metodą R/S z estymacją wartości w punktach  $2^n$

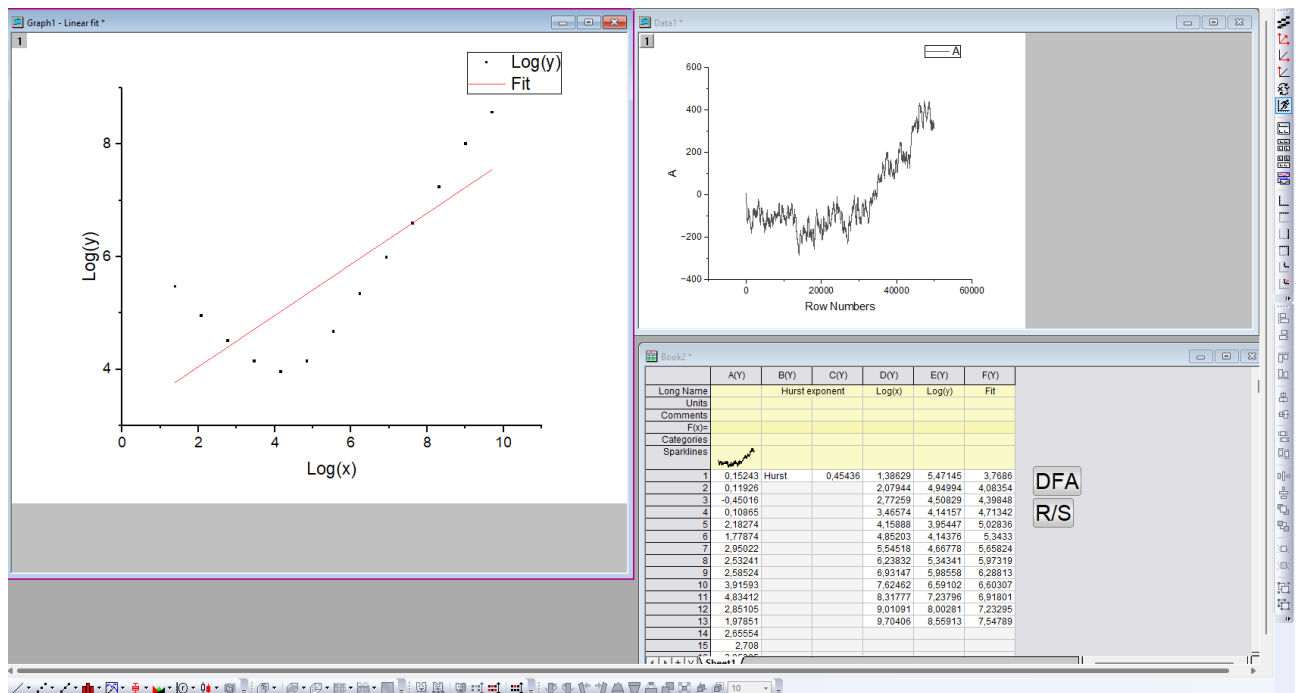
3)  $H = 0.7$  - rys. 3.12- 3.14.



Rysunek 3.12: Wyniki uzyskane metodą DFA

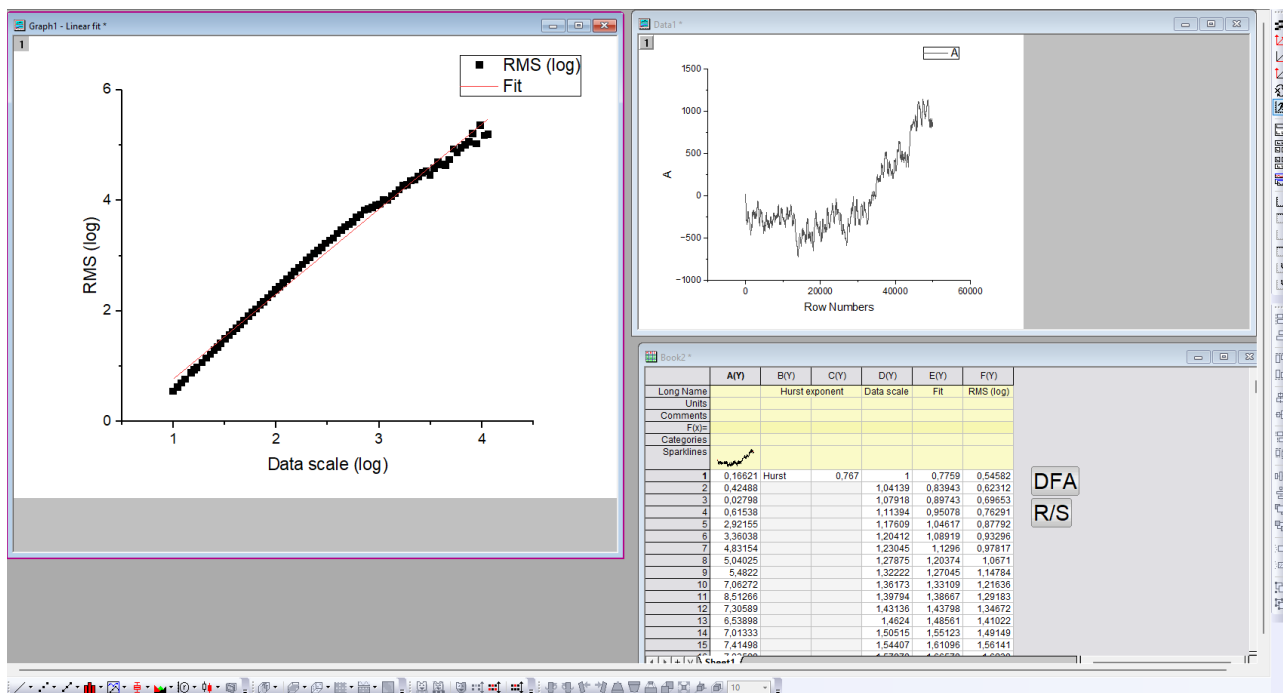


Rysunek 3.13: Wyniki uzyskane metodą R/S z estymacją wartości w każdym punkcie

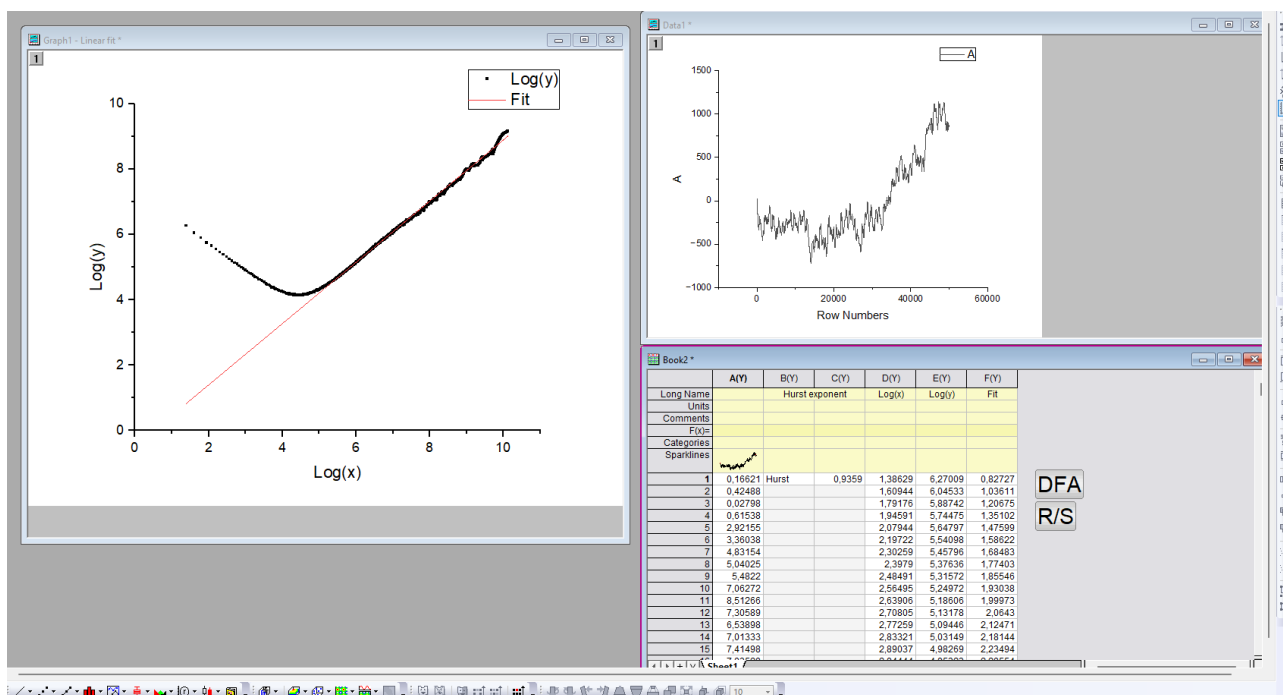


Rysunek 3.14: Wyniki uzyskane metodą R/S z estymacją wartości w punktach  $2^n$

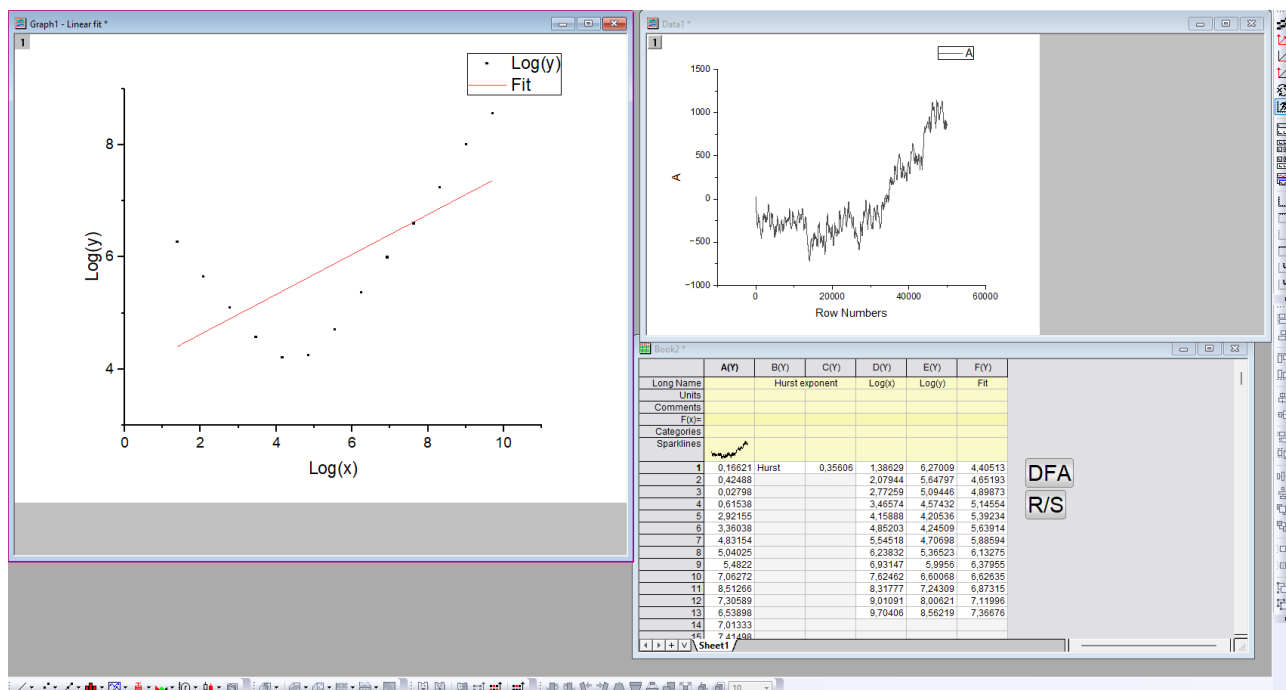
4)  $H = 0.9$  - rys. 3.15- 3.17.



Rysunek 3.15: Wyniki uzyskane metodą DFA



Rysunek 3.16: Wyniki uzyskane metodą R/S z estymacją wartości w każdym punkcie



Rysunek 3.17: Wyniki uzyskane metodą R/S z estymacją wartości w punktach  $2^n$

### 3.5. Analiza uzyskanych wyników

Analizie zostały poddane zbiory z indeksem  $H = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ . Uzyskane wyniki przedstawiono w tabelach 3.1 - 3.3. Jako *Hurst* oznaczono teoretyczną wartość współczynnika Hursta (wartość podawana podczas generacji danych, jak na rys. 3.2). W drugim wierszu znajduje się nazwa metody oraz obliczone za jej pomocą wartości  $H$ . Na końcu obliczono także wartość procentową różnicy między przewidywaną, a rzeczywistą wartością indeksu.

Tabela 3.1: Porównanie wyników dla metody DFA

<i>Hurst</i>	0,4	0,5	0,6	0,7	0,8	0,9
DFA	0,663	0,692	0,717	0,738	0,755	0,767
Różnica [%]	26,3	19,2	11,7	3,8	4,5	13,3

W przypadku estymacji współczynnika metodą DFA (Tabela 3.1), największa dokładność została uzyskana dla danych o wartości  $H = 0,7$ . W tym przypadku, różnica wynosi jedynie 3,8%, a średnia wartość różnic 13,13%.

Tabela 3.2: Porównanie wyników dla metody R/S z estymacją wartości w każdym punkcie

<i>Hurst</i>	0,4	0,5	0,6	0,7	0,8	0,9
R/S $n$	0,980	0,967	0,955	0,948	0,941	0,936
Różnica [%]	58,0	46,7	35,5	24,8	14,1	3,6

Współczynnik Hursta uzyskany za pomocą dokładnej metody R/S rośnie odwrotnie proporcjonalnie do obliczanego w programie "tsfBm" (Tabela 3.2) i zawsze jest większy od 0,9. Może być to spowodowane stosunkowo niską wartością parametru  $-m$  w generatorze danych wejściowych. Ponadto wartość średniej różnicy wynosi 30,45%.

Tabela 3.3: Porównanie wyników dla metody R/S z estymacją wartości w punktach  $2^n$

<i>Hurst</i>	0,4	0,5	0,6	0,7	0,8	0,9
R/S $2^n$	0,638	0,574	0,512	0,454	0,402	0,356
Różnica [%]	23,8	7,4	8,8	24,6	39,8	54,4

Dla uproszczonej metody R/S (Tabela 3.3), najdokładniejsze wyniki otrzymano dla błędzenia losowego. Średnia różnica jest równa 26,47%. Dla takiego zestawu danych średnia różnica jest większa niż otrzymana za pomocą metody DFA, ale mniejsza niż dla dokładnej metody R/S (z estymacją wartości w każdym punkcie).

## 4. Podsumowanie i wnioski końcowe

Indeks Hursta jest określany jako miara stopnia zależności dalekosiężnych w szeregach czasowych. Celem niniejszej pracy było opracowanie i implementacja funkcji estymujących jego wartość w programie Origin. W części teoretycznej przedstawiono definicję zależności dalekosiężnych oraz opisano czym jest indeks Hursta. Wyróżnia się trzy klasy podziału szeregów w zależności od wartości indeksu  $H$ : szereg antypersystentny (różowy szum), błądzenie losowe (biały szum) oraz szereg persystentny (czarny szum). Wyjaśniono w formie skrótowej również metodę jego wyznaczania dwoma metodami: R/S oraz DFA.

W części praktycznej przedstawiono sposób generacji szeregów, implementację i opis funkcji pozwalających na wyznaczenie indeksu  $H$  za pomocą metody Przeskalowanego Zasięgu i Beztrendowej Analizy Fluktuacyjnej. Następnie przedstawiono otrzymane wyniki oraz przeprowadzono analizę otrzymanych danych.

Na podstawie otrzymanych wyników można zauważyć, jak duże znaczenie ma dokładność w zaimplementowanych funkcjach. Ta sama metoda z estymacją wartości w każdym punkcie, w porównaniu z wynikami otrzymanymi w przypadku estymacji w punktach  $2^n$ , zwraca wyniki różniące się nawet o 60%. Indeks  $H$  otrzymany z użyciem metody DFA jest bardziej zbliżony do otrzymanego za pomocą mniej dokładnej funkcji R/S. Może to wskazywać na relatywnie niską dokładność procedury generującej dane lub konieczność wykonania większej liczby testów generacji danych wejściowych z różnymi wartościami parametru  $-m$ . Co prawda nie można określić dokładnej wartości, którą przyjmie kolejny element szeregu, lecz na podstawie dużej ilości danych możliwe jest określenie prawdopodobieństwa persystencji. Aktualnie trwają liczne badania określające użyteczność  $H$  podczas analiz rynków kapitałowych.

Zalecane byłoby stworzenie, z przedstawionych rozwiązań w programie „Origin”, aplikacji. Dostęp do tych funkcjonalności byłby łatwiejszy i wymagał tylko instalacji pliku. Ponadto przy tworzeniu funkcji można by stworzyć funkcjonalność umożliwiającą wybór kolumny z danymi (w tym przypadku jest to zawsze pierwsza kolumna).

Autor za własny wkład pracy uważa:

- 1) Zgłębienie literatury dotyczącej zależności dalekosiężnych i tematów pokrewnych.

- 2) Zebranie z wielu źródeł różnych ujęć opisujących ideę i metody obliczania wykładnika Hursta.
- 3) Nauka podstaw programowania w języku JAVA.
- 4) Nauka programu OriginLab od podstaw.
- 5) Stworzenie programu w OriginLab obliczającego indeks  $H$  dwoma metodami.
- 6) Rozwinięcie umiejętności programowania w języku Python.
- 7) Napisanie funkcji wyznaczającej wartość wykładnika Hursta w języku Python.
- 8) Analiza otrzymanych wyników.



## Literatura

- [1] <https://jbc.bj.uj.edu.pl/Content/467084/ruchy-browna.html>, [26.01.2023]
- [2] L. BACHELIER: Théorie de la spéculation Annales scientifiques de l'É.N.S. 3e série, tome 17 (1900), p. 21-86 <http://www.numdam.org/item/10.24033/asens.476.pdf>, [26.01.2023]
- [3] Zbigniew Jan Jurek: Matematyczne zarządzanie ryzykiem - losowością, Wydawnictwo Uniwersytetu Wrocławskiego, 2013
- [4] Michał Pietrzak: Wykorzystanie funkcji Holdera w modelowaniu cen spółek na WGPW, 2003,  
<https://dspace.uni.lodz.pl/xmlui/bitstream/handle/11089/6827/62-82.pdf?sequence=1&isAllowed=y>
- [5] [https://d2mvzyuse3lwjc.cloudfront.net/pdfs/Origin2019\\_Documentation/English/Booklet2019\\_E.pdf](https://d2mvzyuse3lwjc.cloudfront.net/pdfs/Origin2019_Documentation/English/Booklet2019_E.pdf), [26.01.2023]
- [6] <https://www.fuw.edu.pl/~glink/ekonofizyka/prezentacje/2015-05-12-miaraRyzykaWprowadzenie.pdf>, [26.01.2023]
- [7] [https://sj.wne.sggw.pl/pdf/EIOGZ\\_2006\\_n60\\_s231.pdf](https://sj.wne.sggw.pl/pdf/EIOGZ_2006_n60_s231.pdf), [26.01.2023]
- [8] Tomasz Popławski, Piotr Szelaąg: Wykorzystanie wykładnika Hursta do przewidywania niestabilności generacji wiatrowej, 2014  
<https://www.cire.pl/pliki/2/wykorzystaniewykadnikahursta.pdf>
- [9] Anna Kiłyk, Zofia Wilmowska: Wykorzystanie wykładnika Hursta do prognozowania zmian cen na giełdzie papierów wartościowych
- [10] Edgar E. Peters: Teoria Chaosu a rynki kapitałowe, Wydawnictwo: WIG-Press, 1997
- [11] Zuzanna Rzeszółko: Analiza właściwości fraktalnych szeregów czasowych wybranych indeksów giełdowych,  
Tom 17 Nr 3 (2016) <https://qme.sggw.edu.pl/article/view/3877/3504>, [26.01.2023]

- [12] Henryk Gurgul, Marcin Suder: Ekonometryczna analiza fraktalnych właściwości struktury przepływu gazu w wybranych stacjach I stopnia, 2012, [https://bazhum.muzhp.pl/media/files/Managerial\\_Economics/Managerial\\_Economics-r2012-t11/Managerial\\_Economics-r2012-t11-s77-99/Managerial\\_Economics-r2012-t11-s77-99.pdf](https://bazhum.muzhp.pl/media/files/Managerial_Economics/Managerial_Economics-r2012-t11/Managerial_Economics-r2012-t11-s77-99/Managerial_Economics-r2012-t11-s77-99.pdf), [26.01.2023]
- [13] <http://www.johncon.com/ntropix/utilities.html>, [26.01.2023]
- [14] Igor Belov, Audrius Kabašinskas, Leonidas Sakalauskas: A study of stable models of stock markets, Institute of Mathematics and Informatics, Operational Research Sector at Data Analysis Department Akademijos St. 4, Vilnius 08663, Lithuania
- [15] Felipe Contreras: Series de Tiempo Para Complejidad 2 / SC2 Antecedentes, 2020
- [16] Marina Resta: Hurst Exponent and its Applications in Time-series Analysis, Recent Patents on Computer Science 2012, 5, 211-219, <https://docserver.ingentaconnect.com/deliver/connect/ben/22132759/v5n3/s5.pdf?expires=1674910631&id=0000&titleid=99002320&checksum=BDC57D341CBAEA20FAA1BC8623124153&host=https://www.ingentaconnect.com>, [28.01.2023]
- [17] T. Karagiannis, M. Faloutsos, and M. Molle. The SELFIS Tool. Accessed: Mar. 2, 2021 <http://alumni.cs.ucr.edu/~tkarag/Selfis/Selfis.html>, [26.01.2023]