

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA

BAKALÁŘSKÁ PRÁCE

Analýza časových řad metodou přeškálovaných rozsahů
– R/S analýza



Katedra matematické analýzy a aplikací matematiky

Vedoucí diplomové práce: **Mgr. Ondřej Vencálek, Ph.D.**

Vypracoval: **Adam Olejníček**

Studijní program: **B1103 – Aplikovaná matematika**

Studijní obor: **ME Matematika-ekonomie se zaměřením na bankovníctví / pojišťovnictví**

Forma studia: **prezenční**

Rok odevzdání: **2022**

BIBLIOGRAFICKÁ IDENTIFIKACE

Autor: Adam Olejníček

Název práce: Analýza časových řad metodou přeškálovaných rozsahů – R/S analýza

Typ práce: Bakalářská práce

Pracoviště: Katedra matematické analýzy a aplikací matematiky

Vedoucí práce: Mgr. Ondřej Vencálek, Ph.D.

Rok obhajoby práce: 2022

Abstrakt: Cílem této práce je seznámit čtenáře s jednou ze statistických metod analýzy časových řad – R/S analýzou a s jejím výstupem, Hurstovým exponentem. V první kapitole bude tato metoda spolu s jejími souvislostmi představena. Dále bude obsahovat popis principu této metody a algoritmus výpočtu. Na závěr této kapitoly bude uvedeno, jak lze provést tuto analýzu pomocí programovacího jazyka R. Druhá kapitola bude zaměřena zejména na experimenty, ve kterých bude ověřena funkčnost této metody. Ve třetí kapitole bude představeno několik funkcí, vytvořených pro usnadnění experimentů. V poslední, čtvrté kapitole, bude tato metoda použita pro analýzu reálných časových řad.

Klíčová slova: Časová řada, R/S analýza, Hurstův exponent

Počet stran: 48

Počet příloh: 1

Jazyk: český

BIBLIOGRAPHICAL IDENTIFICATION

Author: Adam Olejníček

Title: Time series analysis using the rescaled range method – R/S analysis

Type of thesis: Bachelor thesis

Department: Department of Mathematical Analysis and Application of Mathematics

Supervisor: Mgr. Ondřej Vencálek, Ph.D.

The year of presentation: 2022

Abstract: The aim of this thesis is to introduce one of the statistical methods of time series analysis – R/S analysis and its output, the Hurst exponent. In the first chapter, this method will be introduced along with its context. It will also include a description of the principle of this method and the algorithm of calculation. At the end of this chapter, it will be shown how this analysis can be performed using the programming language R. The second chapter will mainly focus on experiments in which the functionality of this method will be verified. The third chapter will introduce several functions created to facilitate the experiments. In the last chapter, chapter four, this method will be applied to real time series analysis.

Number of pages: 48

Number of appendices: 1

Language: Czech

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně pod vedením pana Mgr. Ondřeje Vencálka, Ph.D. a všechny použité zdroje jsem uvedl v seznamu literatury.

V Olomouci dne

.....

Adam Olejníček

Poděkování

Rád bych zde poděkoval vedoucímu své práce, panu Mgr. Ondřeji Vencálkovi, Ph.D. za jeho odbornou pomoc a za jeho ochotu při tvorbě této práce.

Obsah

| | |
|--|-----------|
| Úvod | 8 |
| 1 Analýza časových řad metodou přeskálovaných rozsahů | 9 |
| 1.1 Vysvětlení základních pojmů | 9 |
| 1.2 Hurstův exponent..... | 10 |
| 1.3 Hurstův exponent a fraktální dimenze | 11 |
| 1.4 Odhad Hurstova exponentu metodou přeskálovaných rozsahů | 13 |
| 1.5 Výpočet pomocí programovacího jazyka R..... | 17 |
| 2 Experimentální analýza | 21 |
| 2.1 Normálně rozdělená..... | 21 |
| 2.2 Lineární závislost | 21 |
| 2.3 Alternující řada | 25 |
| 2.4 Alternující řada se zvyšujícím se rozsahem | 26 |
| 2.5 Rovnoměrně rozdělená..... | 28 |
| 2.6 Kumulované součty..... | 29 |
| 2.7 Reálná data | 31 |
| 2.8 Shrnutí výsledků..... | 34 |
| 2.9 Boxplot rozdělení Hurstových exponentů..... | 34 |
| 3 Funkce..... | 36 |
| 3.1 HurstN | 36 |
| 3.2 HurstNL..... | 37 |
| 3.3 HurstI | 38 |
| 3.4 Hurstbox..... | 38 |
| 4 Analýza reálných dat..... | 39 |
| 4.1 Google..... | 39 |
| 4.2 Facebook (Meta Platforms)..... | 40 |
| 4.3 Dow Jones Industrial Average (DJIA) | 41 |
| 4.4 Shrnutí..... | 42 |
| Závěr..... | 44 |

Úvod

V dnešní době, která je plná různých informací, existuje již jen málo lidí, kteří si dovolí zpochybňovat důležitost dat a informací. Dokonce se o datech čím dál častěji mluví jako o jedné z nejcennějších komodit. V praxi můžeme narazit na mnoho souborů dat mnoha různých druhů. Jedním z nejdůležitějších jsou soubory dat, kde každý údaj pochází z určitého konkrétního období (z konkrétní hodiny, dne, měsíce, roku atd.) a jsou chronologicky seřazeny. Takovými soubory dat se říká časové řady. Analýzou časových řad, možná právě kvůli touze předpovídat budoucnost, se věnovalo, věnuje a nadále i jistě bude věnovat mnoho odborníků.

Uplynula již dlouhá doba, kdy si lidé poprvé uvědomili důležitost časových řad. Za tuto dobu bylo objeveno mnoho metod, s jejichž pomocí můžeme získat spoustu užitečných informací o mnoha různých časových řadách. Různé metody pocházejí z různých oborů. My se v této práci budeme věnovat jedné ze statistických metod analýzy časových řad. Tato metoda se nazývá „R/S analýza“ a její výstupní hodnotou je „Hurstův exponent“. Tato práce je rozčleněna do čtyř kapitol, ve kterých se pokusím odpovědět na otázky: „Co je to R/S analýza? Jak souvisí s fraktály? Jak se počítá? Skutečně funguje, tak jak by fungovat měla? Funguje i na reálných datech?“. Obsah těchto kapitol si nyní ve stručnosti představíme.

V první kapitole se čtenář dozví základní informace o této metodě a o jejím výstupu. Dozví se také, co je to fraktální dimenze a jak s naší metodou souvisí. Popíšeme si algoritmus R/S analýzy, a na závěr této kapitoly si popíšeme, jak lze tuto metodu provést za pomoci programovacího jazyka R.

Ve druhé kapitole si ověříme, zda metoda funguje podle našich představ. Tuto metodu zde použijeme na několik časových řad, kde většina těchto řad bude vytvořena přímo pro daný experiment. Pro každý experiment si uvedeme naše očekávání a poté závěr. Všechny získané výsledky si na konci této kapitoly shrneme.

Třetí kapitola je věnována popisu funkcí, které jsem si vytvořil pro usnadnění výzkumu. Také si zde s pomocí těchto funkcí provedeme několik pokusů, kde si ukážeme, jak reaguje R/S analýza na změny vlastností dané časové řady, například na změnu délky časové řady.

Ve čtvrté kapitole, která je i poslední, provedeme R/S analýzu na reálných datech. Analyzovaná data se budou týkat vývoje hodnot akcií různých známých společností (případně celého indexu akcií). Na závěr této kapitoly si všechny získané výsledky opět shrneme.

1 Analýza časových řad metodou přeškálovaných rozsahů

V této práci se budeme zabývat analýzou časových řad pomocí metody přeškálovaných rozsahů, též nazývané jako R/S analýza. Díky metodě přeškálovaných rozsahů budeme schopni získat informaci o perzistenci časové řady. Jako perzistentní označíme časovou řadu, která má tendenci, zachovávat svůj trend. Časová řada s touto vlastností se nazývá též jako tzv. proces s dlouhodobou pamětí. Opakem perzistentní časové řady je časová řada antiperzistentní. V případě antiperzistentní časové řady, se hodnoty časové řady nedrží rostoucího ani klesajícího trendu a jedná se o řadu tzv. alternující. Alternující časovou řadou máme v tomto případě na mysli časovou řadu, kde je větší pravděpodobnost, že po vysoké hodnotě bude následovat hodnota nízká, a naopak po hodnotě nízké opět hodnota vysoká. Tato metoda nám také poskytne informaci, jak moc je daná časová řada perzistentní či antiperzistentní. Poslední možností je, že časovou řadu vyhodnotíme jako tzv. bílý šum (tedy, že hodnoty přírůstků časové řady odpovídají bílému šumu). V takovém případě jsou na sobě hodnoty časové řady vzájemně nezávislé a není tedy možné předpovídat hodnoty následující. Znalost toho, o jaký typ časové řady se jedná, nám může pomoci v predikci časové řady do budoucna.

Analýza metodou přeškálovaných rozsahů (anglicky rescaled range analysis, zkráceně R/S analýza) nám dá jako výstup odhad hodnoty Hurstova exponentu.

Jak bylo v úvodu řečeno, odhady hodnot Hurstova exponentu se v této práci pokusíme získat pomocí metody přeškálovaných rozsahů (dále jen R/S analýzy) časové řady. Podotýkám, že touto metodou získáme pouze odhad, který vychází oproti skutečné hodnotě mírně zkreslený (vyšší, než je ve skutečnosti, později si to ověříme). Tuto analýzu je dobré provádět na časových řadách, jejichž délka odpovídá mocnině čísla 2, tedy na řadách, kde počet záznamů je roven délce 512, 1024, 2048, 4096, atd. V následující části si definujeme několik pojmů.

1.1 Vysvětlení základních pojmů

V této sekci si představíme několik základních pojmů, které jsou v této práci používány. Tyto pojmy se týkají zejména časových řad a jejich vlastností.

- **Perzistencí časové řady** rozumíme sílu korelace mezi jednotlivými hodnotami časové řady. Podle této vlastnosti rozlišujeme časové řady perzistentní, antiperzistentní nebo časové řady, které se chovají jako bílý šum.

- **Bílým šumem** se označuje náhodný signál s rovnoměrnou výkonovou spektrální hustotou. [1] O časové řadě, jejíž hodnoty se vyvíjejí bez jakékoliv závislosti na hodnotách předchozích řekneme, že se chová jako bílý šum. Hodnoty takové časové řady mají nulovou autokorelaci.
- **Náhodnou procházkou** označujeme časovou řadu, jejíž přírůstky odpovídají bílému šumu, avšak každá hodnota přírůstku je přičtena k předchozí hodnotě časové řady, a tedy každá hodnota vychází z hodnoty jí předcházející. Jinými slovy, jedná se o časovou řadu kumulovaných součtů hodnot časové řady odpovídající bílému šumu.
- Časovou řadou s **dlouhodobou pamětí**, rozumíme časovou řadu, která je perzistentní. Rozumíme tím tedy vlastnost, kdy množina předchozích hodnot má přímý vliv na hodnoty aktuální.
- **Hurstův exponent** je výstupní hodnotou R/S analýzy časové řady, který může nabývat hodnot z intervalu $[0, 1]$. Shrnutí hodnot, kterých může Hurstův exponent nabývat i s příslušnou interpretací si uvedeme v tabulce:

| <i>Odhad Hurstova exponentu:</i> | <i>Chování časové řady:</i> |
|----------------------------------|-----------------------------|
| > 0.5 | <i>Perzistentní</i> |
| $= 0.5$ | <i>Náhodná procházka</i> |
| < 0.5 | <i>Antiperzistentní</i> |

1.2 Hurstův exponent

Na úvod této kapitoly by bylo vhodné, představit si stěžejní pojem této práce a sice, co je to časová řada. Časová řada, je chronologicky uspořádaná posloupnost dat (hodnot). U časové řady lze zkoumat mnoho vlastností, které se týkají sezónnosti, trendu nebo například dříve zmiňované vlastnosti – perzistence. S výzkumem těchto vlastností nám může pomoci ukazatel, který se nazývá Hurstův exponent. Tento exponent se označuje písmenem „H“. Nejlepší odhady tohoto exponentu nám poskytuje metoda přeškálovaných rozsahů (R/S analýza). Autorem tohoto koeficientu a celé metody přeškálovaných rozsahů je Harold Edwin Hurst (1900-1978), po němž je tento koeficient pojmenován. Hurst objevil tuto metodu analýzy časových řad, při navrhování přehrady na řece Nil. Na základě historických dat o výšce hladiny Nilu se snažil odhadnout optimální výšku přehrady. Při svém výzkumu dospěl

k překvapivému závěru, a sice že časová řada historických dat o výšce hladiny Nilu se ani zdaleka nepodobá náhodné procházce a má spíše perzistentní charakter. [2, str. 54-55]

Je více způsobů, jak získat odhad Hurstova exponentu. Můžeme jej získat pomocí metody DFA (Detrended Fluctuation Analysis) nebo metody Periodogram Regression. Nejčastěji se však pro odhad tohoto koeficientu využívá, z důvodu nejmenší směrodatné odchylky odhadů, právě R/S analýza. [3] V mojí práci se budu věnovat zkoumání náhodnosti časové řady pomocí R/S analýzy. R/S analýza a v jejím rámci odhadovaný Hurstův exponent, nám poskytuje nástroj, který nám může pomoci určit, zda je zkoumaná časová řada náhodná či nikoliv. Ukázalo se totiž, že Hurstův exponent vychází pro všechny náhodné časové řady (alespoň přibližně) stejný, a sice roven hodnotě $H = 0,5$. Jednou z velkých výhod R/S analýzy je, že se jedná o neparametrickou metodu [2, str. 61], nemusíme tedy o datech předpokládat, že pochází z jakéhokoliv konkrétního rozdělení. Hurstův koeficient může nabývat hodnot z rozmezí $[0, 1]$, přičemž hodnota blízká pravé hraniční hodnotě intervalu, tedy hodnotě 1, nám o časové řadě napoví, že se jedná o řadu perzistentní (tedy, že se jedná o časovou řadu s dlouhodobou pamětí) a naopak, když vyjde blíže levé hraniční hodnotě intervalu (blízko 0), získáme tím informaci, že se jedná o řadu antiperzistentní (takováto časová řada dlouhodobou paměť postrádá). Pokud koeficient vyjde blízko hodnoty 0.5, pak můžeme o zkoumané časové řadě říct, že se řada chová podobně jako „bílý šum“. [4, str. 226]

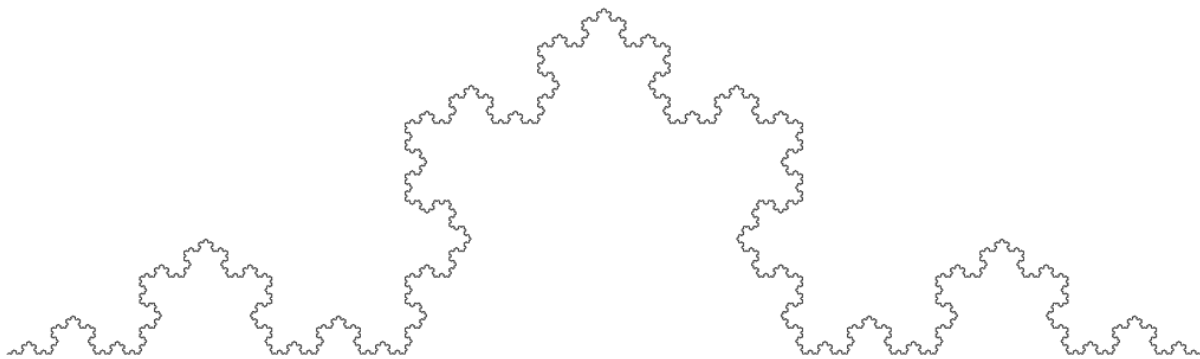
Mimo výše zmíněné, Hurstův exponent také souvisí s fraktální dimenzí, tomuto vztahu je věnována následující kapitola.

1.3 Hurstův exponent a fraktální dimenze

Jak bylo zmíněno výše, Hurstův exponent souvisí s fraktální dimenzí¹ zkoumané křivky (křivky opisující časovou řadu). V této kapitole se nebudeme omezovat na křivku opisující časovou řadu, stačí když se budeme zabývat obecnou křivkou. Fraktální dimenzi křivky můžeme zjistit pomocí jednoduchého vztahu $D = 2 - H$, kde „D“ označuje fraktální dimenzi křivky a „H“ hodnotu Hurstova exponentu [4, str. 226]. Fraktální dimenze křivky může být definována různými způsoby. Podle jedné definice nám udává míru, jak moc je křivka členitá. Podle jiné definice souvisí fraktální dimenze s tím, jak moc křivka zaplňuje plochu [5]. Může nabývat hodnoty v intervalu $[1, 2)$, přičemž pro geometricky hladký útvar se fraktální dimenze

¹ Fraktální dimenze se někdy též nazývá jako Hausdorffova dimenze nebo Hausdorff – Besicovitchova dimenze

shoduje s klasickou topologickou dimenzí křivky, je tedy rovna hodnotě 1. Geometricky hladkým útvarům máme na mysli klasickou „hladkou“ křivku. S rostoucí „hrubostí“ křivky, její fraktální dimenze roste k hodnotě 2, avšak této hodnoty dosáhnout nemůže. Lze říct, že fraktální dimenze křivky nám poskytuje informaci o tom, jak moc je křivka členitá, nebo jak hustě zaplňuje prostor². Fraktální dimenze křivky může být větší nebo rovna její dimenzi topologické. Jako užitečné se zdá její využití při zkoumání ekonomických časových řad, jako například časové řady vývoje cen akcií za určitý časový horizont. Příklad, jak křivka časové řady s vysokou i nízkou fraktální dimenzí vypadá nalezneme například v kapitole 2.2 na straně 12 a v kapitole 2.3 na straně 15. Pro ilustraci si zde ukážeme příklad takovéto křivky.

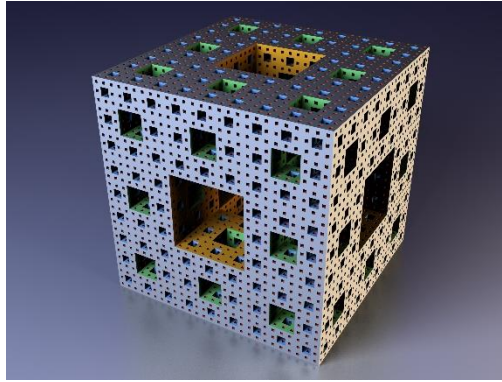


Obrázek 1: Kochova křivka [7]

Na obrázku můžeme Kochovu křivku, jejíž topologická dimenze je 1 a fraktální 1,26. Jedná se o jednu z prvních popsaných fraktálních křivek.

Fraktální dimenzi můžeme určit i pro dvoudimenzionální geometrický objekt. I v tomto případě odpovídá fraktální dimenze míře členitosti objektu, a tedy tomu, jak moc se zkoumaný objekt určité topologické dimenze podobá objektu s topologickou dimenzí o jedničku vyšší. Pro lepší představu si uvedeme příklad takovéhoho objektu.

² Fraktální dimenze je definována různě. Toto je nástin, jak můžeme chápat jednu z možných definic fraktální dimenze [6, str. 342]



Obrázek 2: Mengerova houba[8]

Na obrázku vidíme Mengerovu houbu, jejíž topologická dimenze odpovídá hodnotě 2 a její fraktální dimenze se blíží hodnotě 3 (přesná hodnota fraktální dimenze tohoto objektu není známá).

Zajímavostí těchto objektů je, že ač křivka s fraktální dimenzí blízkou hodnotě 2 pokrývá značnou část prostoru, její obsah je stále nulový. Analogicky u dvourozměrného objektu s fraktální dimenzí blízkou hodnotě 3. Takovýto objekt zabírá značnou část prostoru, avšak jeho objem je nulový.

1.4 Odhad Hurstova exponentu metodou přeškálovaných rozsahů

V této kapitole se pokusím popsat a vysvětlit princip výpočtu odhadu Hurstova exponentu pomocí R/S analýzy. Jak již bylo zmíněno Hurstův exponent budu odhadovat pomocí R/S analýzy především z důvodu nejmenší směrodatné odchylky odhadů získaných touto metodou v porovnání s odhady získaných jinými metodami. U této metody je známo že odhady nadhodnocuje, toto nadhodnocení potom klesá s rostoucí délkou zkoumané časové řady. U této metody je obvyklé zkoumat časové řady délek mocniny čísla 2, tedy řady délek 512, 1024, 2048, 4096, atd. [15]. V naší práci se tedy omezíme na zkoumání časových řad takovýchto délek.

Pro jednodušší pochopení principu R/S analýzy zde uvedu obrazný popis této metody:

„Představme si člověka přesně uprostřed široké silnice, kdy každým krokem kráčí vpřed a současně kousek do strany, doprava, či doleva, například opilého člověka. Uvažujme, že jeho posuny do strany jsou dány zcela náhodně. Kde se bude tento člověk nacházet po 10 jeho krocích? Budeme-li očekávat, že opět přesně uprostřed silnice, bude to nejlepším možným odhadem, jiný nezkreslený a konzistentní odhad neexistuje. Současně si však můžeme být zcela jisti, že se po 10 krocích nebude nacházet uprostřed silnice. Zajímá nás, jak široké rozpětí

silnice využil. Dostal-li se během těchto 10 kroků maximálně 1 metr vlevo od středové čáry a maximálně 2 metry vpravo od středové čáry, pak „pokryl“ 3 metry šířky silnice. Pokračuje-li dále se stále stejnou střední hodnotou i rozptylem posunů do strany během chůze, je jisté, že pokrytí šířky silnice např. po 40 krocích nebude menší než po 10. Navíc můžeme očekávat že pokrytí šířky silnice se oproti 10 krokům přibližně zdvojnásobí. Cestu tohoto člověka můžeme rozdělit na 1, 2, 4, 8, 16, atd. intervalů a sledovat, jak rostlo průměrné pokrytí šířky silnice v závislosti na počtu kroků odpovídající intervalům zvolené délky. Pohyboval-li se s konstantní střední hodnotou i rozptylem posunů, průměrné pokrytí šířky roste s každým zdvojnásobením počtu kroků přibližně $\sqrt{2}$ – krát, čemuž odpovídá odhad Hurstova exponentu přibližně 0,5. Tento postup odpovídá matematicky zapsanému postupu R/S analýzy s jedním rozdílem, v rámci R/S je maximální rozpětí („pokrytá šířka“) vždy vyděleno lokální směrodatnou odchylkou.“ [10]

Nyní si popíšeme postup pro výpočet Hurstova exponentu pomocí R/S analýzy [10]:

1. Mějme zadanou časovou řadu tvořenou hodnotami x_1, x_2, \dots, x_N pozorovaných v časech $t = 1, 2, \dots, N$. Taková časová řada obsahuje $N - 1$ intervalů délky 1 mezi jednotlivými pozorováními
2. Časovou řadu rozdělíme na m sousedních nepřekrývajících se intervalů délky n , bude tedy platit vztah $N = m \cdot n$
3. Pro každý z těchto m intervalů spočítáme aritmetický průměr pozorovaných hodnot obsažených v daném intervalu, získáme tedy m hodnot aritmetických průměrů $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m$. Aritmetické průměry spočítáme podle vzorce:

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij} \quad j = 1, 2, \dots, m$$

4. Pro každý interval vytvoříme řadu kumulovaných odchylek od hodnoty aritmetického průměru hodnot v daném intervalu.

$$z_{kj} = \sum_{i=1}^k (x_{ij} - \bar{x}_j) \quad k = 1, 2, \dots, n, \quad j = 1, \dots, m.$$

Tím získáme posloupnost n hodnot kumulovaných odchylek, pro každý jeden interval. Celkem tedy získáme posloupnost délky $m \cdot n$ o hodnotách $z_{11}, z_{21}, z_{31}, \dots, z_{1m}, z_{2m}, \dots, z_{nm}$. Hodnotu k -tého částečného součtu odchylek od aritmetického průměru hodnot obsažených v j -tém intervalu značíme koeficientem z_{kj} .

5. Pro každý interval spočítáme

- Rozpětí kumulovaných odchylek od průměrné hodnoty podle vzorce

$$R_j = \max_{k=1, \dots, n} (z_{kj}) - \min_{k=1, \dots, n} (z_{kj}) \geq 0 \quad j = 1, 2, \dots, m$$

Počítáme vlastně rozdíl mezi maximální a minimální hodnotou posloupnosti kumulovaných součtů odchylek pro daný (j -tý) interval.

- Směrodatné odchylky původních hodnot časové řady obsažených ve zkoumaném (j -tém) intervalu

$$s_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}$$

- Dále pro každý z intervalů spočítáme standardizované rozpětí, tedy poměr mezi rozpětím a směrodatnou odchylkou hodnot v daném intervalu

$$R_j/s_j$$

6. Vypočítáme průměrnou hodnotu R/S pro intervaly dané délky n a jí odpovídajícímu dělení původní řady.

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

$(R/S)_n$ je tedy průměrná hodnota R/S ze všech m intervalů o délce n (pro způsob dělení na intervaly o délkách n). Toto spočteme pro všechny způsoby dělení časové řady na intervaly o délkách n rovných mocnině čísla 2. Tedy pro intervaly o délkách $2^0, 2^1, 2^2, \dots, 2^m$.

Hurst předpokládal obecný typ závislosti poměru R/S na délce n ve tvaru

$$(R/S)_n = Cn^H$$

Kde C označuje konstantu a H je Hurstův exponent³. Tento vztah popisuje závislost růstu hodnoty R/S na délkách jednotlivých intervalů, na které je daná časová řada rozdělena, umocněných Hurstovým exponentem.

Hodnotu Hurstova exponentu odhadneme pomocí lineární regrese pro logaritmované hodnoty, kde na ose x budou vyneseny logaritmované délky intervalů a na ose y budou logaritmované hodnoty průměrného standardizovaného rozpětí odpovídající danému n

$$\log_2(R/S)_n = \log_2 C + H \log_2 n$$

Dvojkový logaritmus bereme z důvodu, že danou časovou řadu budeme dělit na intervaly o délce odpovídající mocnině čísla 2. Díky tomu budou hodnoty vykreslené v grafu logaritmické regrese rovnoměrně rozprostřeny. Hurstův exponent je roven směrnici regresní přímky pro logaritmickou regresi. Příklad grafu takovéto regrese nalezneme v této práci na stránce 29.

³ je to vlastně i důvod, proč se tato hodnota nazývá Hurstův exponent

1.5 Výpočet pomocí programovacího jazyka R

V této podkapitole si představíme kód napsaný v programovacím jazyce R, který nám na zadané časové řadě provede algoritmus R/S analýzy, a jako výstupní hodnotu nám vrátí odhad hodnoty Hurstova exponentu příslušného této řadě. V následující části budu popisovat konkrétní kód, pomocí kterého jsme odhadli hodnotu Hurstova exponentu pro časovou řadu hodnot cen akcií Apple. Popis ke každému řádku budu psát nad danou část kódu.

V úvodu uložíme do proměnné `ts.Open` hodnoty zkoumané časové řady. V tomto případě do této proměnné ukládáme data o cenách akcií Applu ze začátku každého pracovního dne, tato data se nacházejí v souboru `AAPL` ve sloupci `AAPL.Open[11]`.

```
ts.Open <- AAPL$AAPL.Open
```

Zkontrolujeme, zda délka sledované časové řady odpovídá hodnotě mocniny čísla 2. Zjistíme, že daná časová řada je tvořena 2048 údaji, což odpovídá číslu 2^{11} .

```
length(ts.Open)
```

V dalších třech řádcích kódu si připravíme pomocné proměnné, do kterých budeme později ukládat vypočtené hodnoty.

```
H.koef = numeric()
```

```
RkuS = numeric()
```

```
H.koef = 0
```

```
RkuS = 0
```

Dále si do proměnné `N` uložíme délku zkoumané časové řady a do proměnné `X` si uložíme data časové řady.

```
N = 2^11
```

```
X = ts(ts.Open)
```

Vytvoříme si vektorovou proměnnou `n.seznam`, do níž si uložíme hodnoty délek jednotlivých intervalů, na které budeme časovou řadu dělit. Proměnná `n.seznam` bude obsahovat hodnoty $\frac{2048}{2^0}; \frac{2048}{2^1}; \frac{2048}{2^2}; \dots; \frac{2048}{2^{11}}$, konkrétně bude tedy tvořena hodnotami 2048; 1024; 512; 256; ..., 1. Délka tohoto vektoru odpovídá číslu 12.

```
n.seznam = N / (2^(0:11))
```

Pro lepší představu o zkoumaných datech si necháme vykreslit graf této časové řady.

```
ts.plot(X)
```

V další části bude následovat první for cyklus. Tento cyklus bude probíhat od 1 do 12, tedy přes počet hodnot vektorové proměnné `n.seznam`, a bude nám určovat, pro jakou délku intervalu budeme standardizované rozpětí počítat.

```
for (i in 1:length(n.seznam))
```

```
{
```

```
  n = n.seznam[i]
```

```
  R = 0; S = 0; rkus = 0
```

Uvnitř prvního for cyklu bude probíhat ještě jeden for cyklus, který bude procházet jednotlivé intervaly vzniklé dělením časové řady na intervaly délky `n` (podle prvního for cyklu) a pro každý z těchto intervalů spočítá standardizovaný rozsah.

```

for (ii in 1:i)
{
  x = X[(ii-1)*n+1:n]
  y = x-mean(x)
  z = cumsum(y)
  R[ii] = max(z)-min(z)
  S[ii] = sd(x)
  rkus[ii] = R[ii]/S[ii]
}

```

Pomocí následujícího příkazu si spočítáme střední hodnotu vektoru standardizovaných rozsahů pro každý z intervalů délky n . Nyní se nacházíme na konci jednoho opakování vnějšího cyklu. Tuto hodnotu si následně uložíme do vektorové proměnné `RkuS` na i -tou pozici.

```

RkuS[i] = mean(rkus)
}

```

Po tom, co skončí první (vnější) `for` cyklus, získáme vektor `RkuS`. Tento vektor bude obsahovat na každé pozici průměrnou hodnotu poměrů R/S ze všech intervalů dané délky n , pro každou délku intervalů, na které je původní časová řada dělena. Z důvodu opačného pořadí hodnot uvnitř vektorů `n.seznam` a `RkuS` musíme pořadí hodnot otočit. To provedeme pomocí následujících příkazů.

```

n.seznam = rev(n.seznam)
RkuS      = rev(RkuS)

```

V další části již přejdeme k závěrečné logaritmické regresi přeškálovaných rozsahů vzhledem k jednotlivým délkám intervalů. Následující příkaz nám vykreslí graf této logaritmické regrese.

```
plot(log2(RkuS) ~ log2(n.seznam))
```

Jak již víme, Hurstův exponent přibližně odpovídá směrnici regresní přímky této logaritmické regrese. Nejdříve si tedy uložíme parametry této přímky do pomocné proměnné `m`

```
m = lm(log2(RkuS) ~ log2(n.seznam))
```

Nyní si z vytvořené proměnné vybereme hodnotu regresního koeficientu, který se nachází na druhém místě vektoru `m`.

```
H.koef = as.numeric(coef(m)[2])
```

Na závěr si hodnotu Hurstova exponentu uložíme do proměnné `HurstOpen` a necháme si ji vypsát.

```
HurstOpen = H.koef
```

```
HurstOpen
```

Tímto máme popsány celý zdrojový kód, pro výpočet Hurstova exponentu v programovacím jazyce R.

V následující části budou popsány experimenty, při kterých jsem si ověřoval, zda Hurstův koeficient získaný R/S analýzou vychází dle očekávání či nikoliv, případně míru nadsazenosti jeho odhadů touto metodou. V další části pak popíšu vytvořené funkce, které nám usnadní výzkum. Veškeré experimenty a analýzy jsem provedl pomocí programovacího jazyka R.

2 Experimentální analýza

V této kapitole si popíšeme experimenty, pomocí nichž jsem zkoumal, zda R/S analýza skutečně funguje dle mých představ. Veškeré experimenty jsem prováděl ve statistickém softwaru RStudio. Do kódu pro algoritmus, který provede na zadaných datech R/S analýzu, jsem zadával různé časové řady (tvořené reálnými daty i daty uměle vytvořenými pro daný experiment) a sledoval jsem, jakou hodnotu mi tato analýza vrátí a zda je tato hodnota v souladu s mým očekáváním, či nikoliv. Nadpisy jednotlivých podkapitol budou indikovat povahu hodnot generované časové řady.

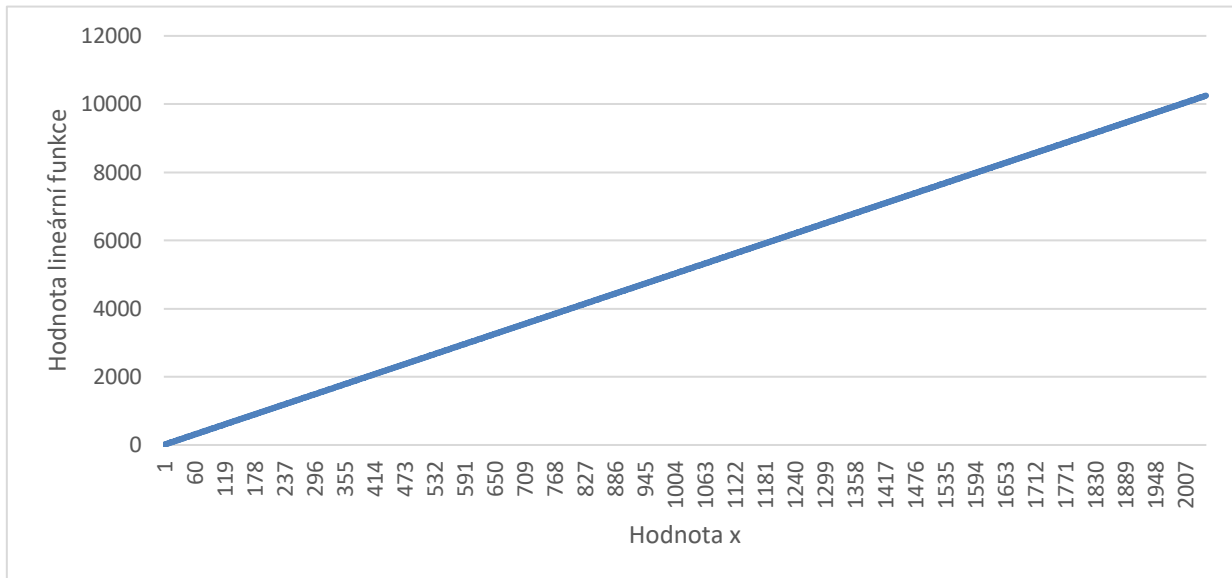
2.1 Normálně rozdělená

Jako první jsem provedl tuto analýzu na časové řadě, jejíž data pochází z normálního rozdělení s nulovou střední hodnotou a jednotkovým rozptylem (získaných příkazem `rnorm(N, 0, 1)`, kde N nabývá hodnoty $2^{11} = 2048$ a určuje nám, kolik hodnot chci vygenerovat (tedy délku naší časové řady), hodnoty 0 a 1 zde určují parametry generujícího rozdělení). Je zřejmé, že by tato řada neměla být perzistentní ani antiperzistentní, a že by její hodnoty měly odpovídat hodnotám časové řady bílého šumu. Očekáváme tedy hodnoty v rozmezí, řekněme, 0.45 až 0.65 (hodnoty spíše vyšší, než pro bílý šum očekáváme, jak jsem již zmínil, z důvodu nadsazenosti odhadů hodnot Hurstova exponentu získaných pomocí R/S analýzy). Pro lepší představu jsem nechal vygenerovat 10 řad po 2048 hodnotách a pro každou řadu jsem odhadl Hurstův exponent. Všechny výsledné hodnoty jsem poté uložil do vektoru, který jsem nazval `Hurstnorm`. Nyní si tyto výsledné hodnoty vypíšeme: 0.560; 0.605; 0.591; 0.577 0.566; 0.573; 0.603 0.593; 0.633; 0.610. Aritmetický průměr získaných hodnot je roven 0.591. Boxplot si pro tento případ nebudeme vykreslovat, z důvodu nízkého počtu hodnot. Výsledky R/S analýzy pro tuto časovou řadu jsou v souladu s naším očekáváním. Dále si také můžeme všimnout míry nadsazenosti odhadů tohoto koeficientu (asi o 0.09 vyšší, než jak by měl vycházet).

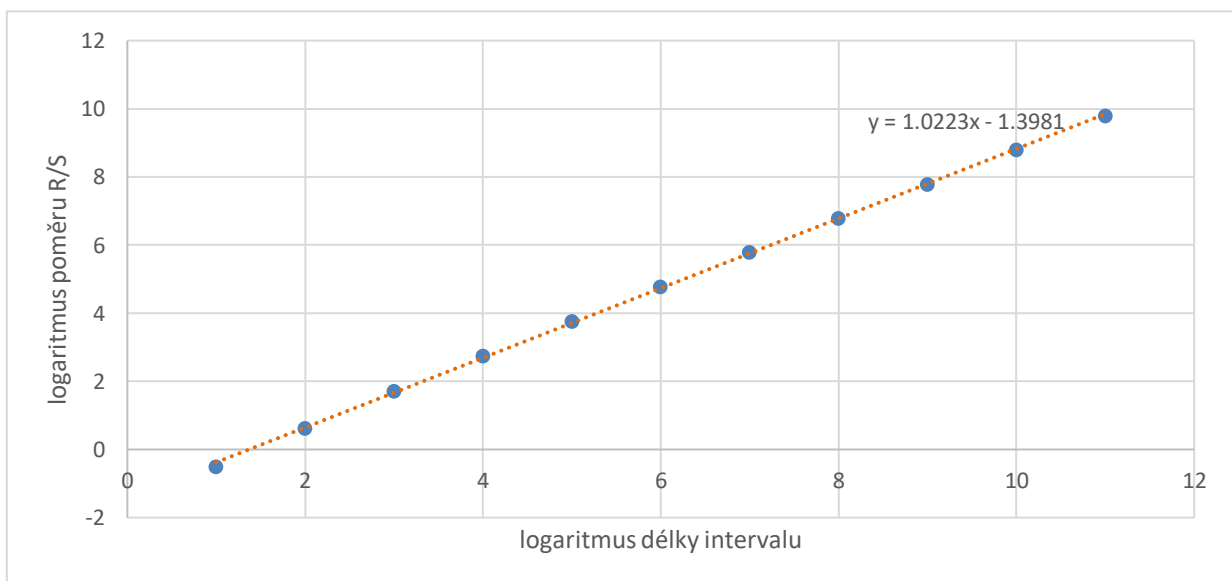
2.2 Lineární závislost

V této podkapitole si ověřím, jak je úspěšná R/S analýza při odhalování vysoké perzistence časové řady, a to tak, že spočítám Hurstův koeficient pro časovou řadu, jejíž hodnoty budou generovány lineární funkcí tvaru $y_n = an + b$, pro $n = 1, 2, 3, \dots, 2024$. Hodnoty a a b zvolíme například jako čísla 5 a 8. Výsledná rovnice generující naši časovou řadu tedy bude $y_n = 5n + 8$. Jak již bylo naznačeno výše, u takovéto časové řady očekáváme

hodnoty Hurstova exponentu velmi blízké hodnotě 1. Algoritmus nám vrátil hodnotu 1.022. Tuto hodnotu si následně uložíme do proměnné `Hurstlin`. Získaná hodnota mě v tomto případě překvapila nikoliv rozdílností od očekávané, ale tím, že překročila svůj obor hodnot, avšak můžeme si všimnout, že jej překročila pouze velmi mírně, tudíž bych toto nadhodnocení opět připsal použité metodě (R/S analýze). Následně si necháme vykreslit graf generovaných hodnot (tedy hodnot, na nichž jsme prováděli tento experiment) do spojnicového grafu spolu s grafem výsledné log-log regrese:



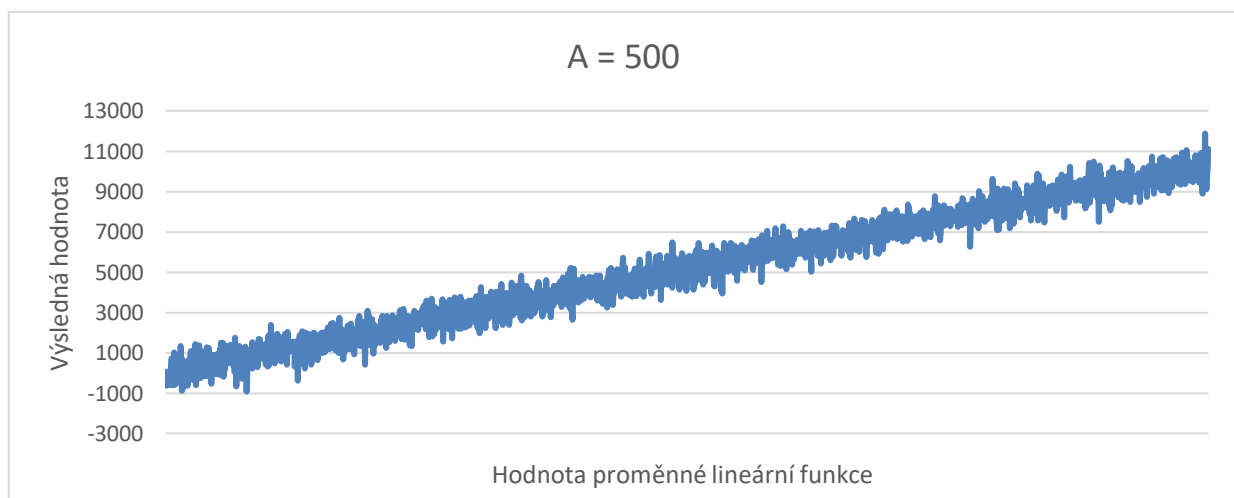
Obrázek 3: Hodnoty č.ř. lineární funkce



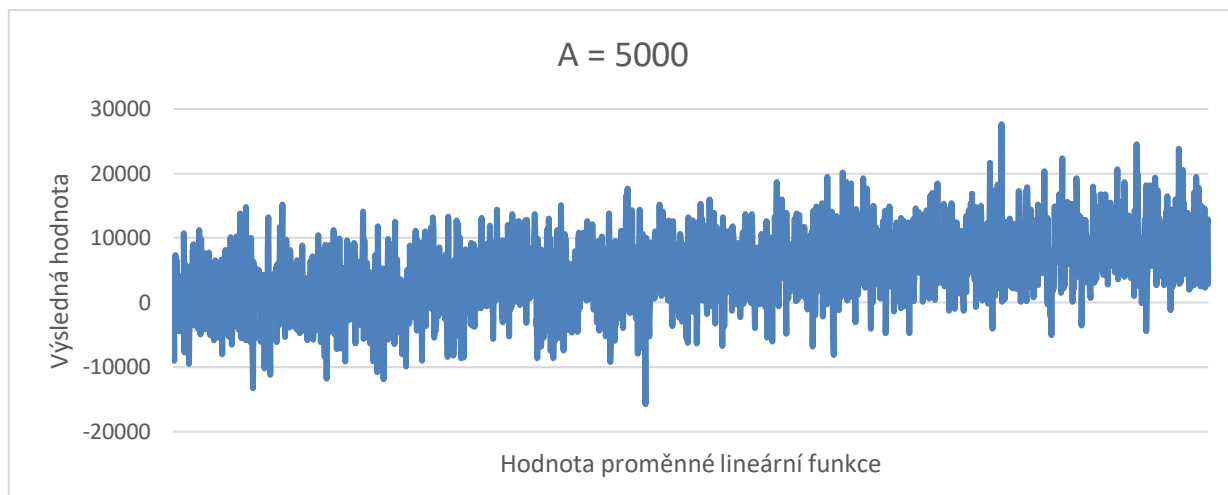
Obrázek 4: Graf závěrečné logaritmické regrese

Na těchto grafech vidíme, jak vypadá vývoj hodnot časové řady generované pomocí lineární funkce a závěrečnou logaritmickou regresi, kde směrnice regresní přímky je rovna Hurstovu koeficientu.

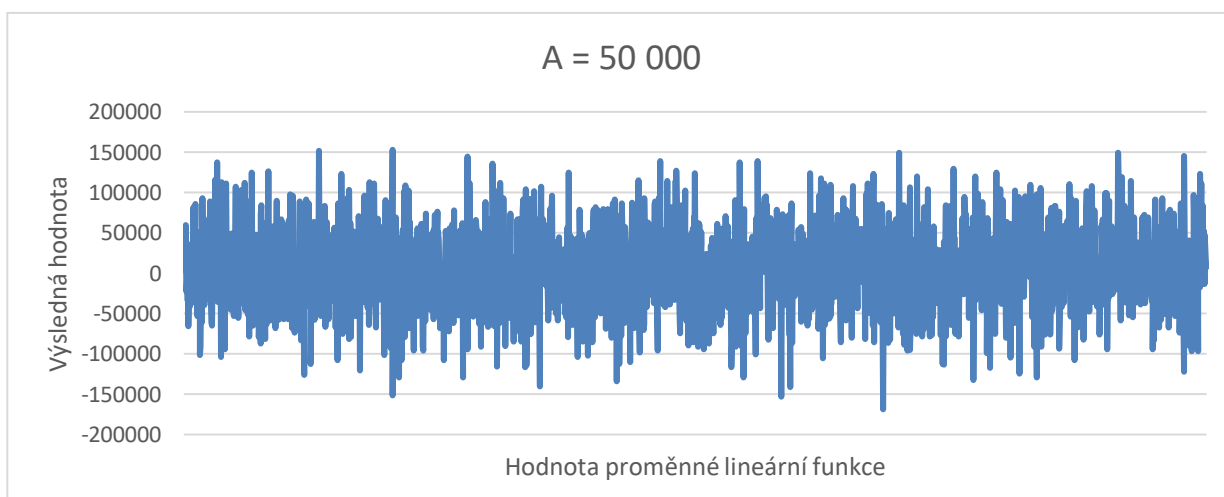
Mohlo by být zajímavé zkoumat, jak se bude chovat řada, generovaná pomocí lineární funkce v případě, že do předpisu funkce zakomponujeme náhodnou složku. Náhodnou složku ke generující funkci přičteme (tedy ke každé generované hodnotě bude přičteno náhodné číslo). Konkrétně budeme přičítat ke každé vygenerované hodnotě, hodnotu získanou z normálního rozdělení se střední hodnotou 0 a směrodatnou odchylkou postupně 500, 5 000 a 50 000. Takto vysoké hodnoty směrodatných odchylek volíme proto, že hodnoty generované původní posloupností se pohybují v řádu tisícovek, a tedy pro nižší hodnoty směrodatných odchylek by byl vliv náhodné složky zanedbatelný. Příkaz, který nám generuje hodnoty časové řady, bude po úpravě vypadat takto: $5 * (1 : 2048) + 8 + \text{rnorm}(2048, 0, A)$, kam místo poslední hodnoty, proměnné „A“, budeme dosazovat postupně 500, 5 000 a 50 000. Nyní si vykreslíme grafy vygenerovaných hodnot.



Obrázek 3: Lineární č.ř. s náhodnou složkou s rozptylem 500



Obrázek 4: Lineární č.ř. s náhodnou složkou s rozptylem 5000



Obrázek 5: Lineární č.ř. s náhodnou složkou s rozptylem 50 000

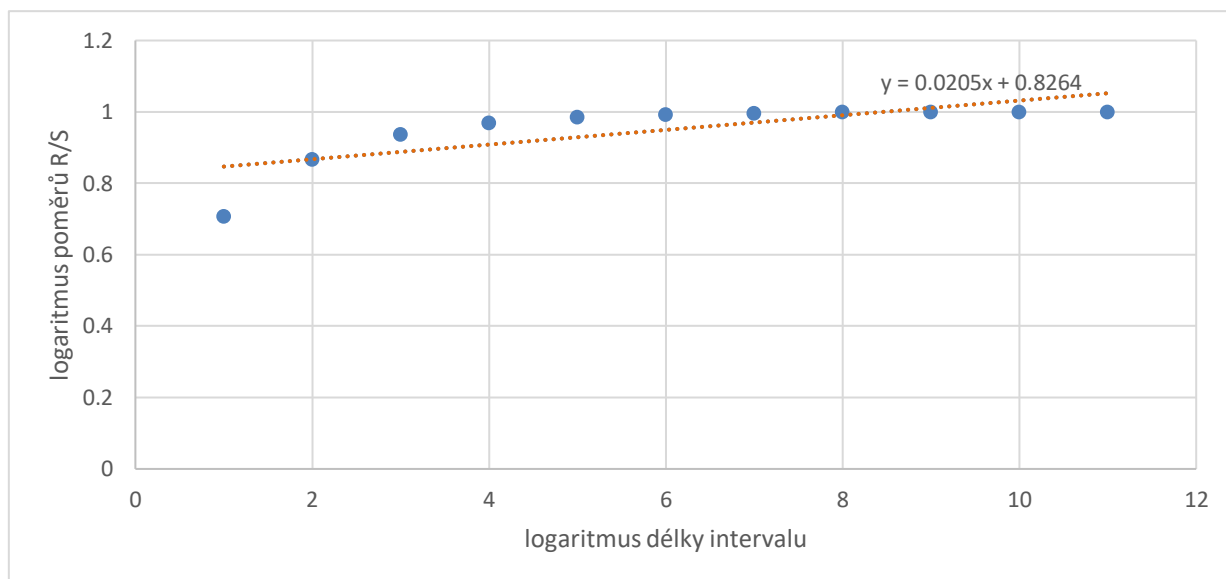
Na grafech si můžeme všimnout zvyšující se rozkmitanosti časové řady. Pro hodnotu rozptylu náhodné složky 500 se řada rozkmitala pouze mírně. Z grafu lze rozpoznat, že se výsledná řada stále velmi silně drží trendu, jde tedy stále vysoce perzistentní řadu. Hurstův koeficient v tomto případě vyšel 1.029 . Pro rozptyl 5 000 lze na časové řadě stále rozpoznat její tendenci držet se rostoucího trendu, avšak její rozkmitanost je již natolik velká, že se díky ní rostoucí trend stává méně zřetelný a časová řada se tedy blíží náhodné procházce. V tomto případě tedy očekáváme nižší hodnotu Hurstova koeficientu, která bude indikovat nižší perzistenci. Jeho hodnota vyšla 0.833 , tedy podle očekávání. Poslední graf nám vykresluje časovou řadu tvořenou opět stejným způsobem jako předchozí dvě, avšak s tím rozdílem, že

nyní je náhodná složka generovaná z normálního rozdělení s rozptylem 50 000. Tato hodnota rozptylu je již natolik velká, že náhodná složka začne generovat hodnoty převyšující všechny ostatní parametry generující funkce, a tedy už z grafu bychom měli být schopni poznat, že se jedná o časovou řadu, jejíž chování odpovídá bílému šumu. Hurstův koeficient pro tuto poslední řadu vyšel 0.612, tedy skutečně blízký hodnotě Hurstova koeficientu pro náhodnou procházku. V poslední části této podkapitoly bych si chtěl ověřit, jakou hodnotu Hurstova koeficientu nám R/S analýza vrátí pro časovou řadu stejného typu jako předchozí, tedy generovanou lineární funkcí, avšak s tím rozdílem, že lineární funkce bude klesající. Její předpis bude $y_n = -5n + 8$. Pro tuto řadu nám, dle očekávání, algoritmus vrátil stejnou hodnotu jako pro řadu generovanou lineární rostoucí funkcí, tedy hodnotu 1.022. Vidíme tedy, že podle Hurstova koeficientu můžeme odhalit perzistenci řady, avšak o směru trendu dané časové řady, nám tento koeficient nic nenapoví.

Hlavním cílem tohoto experimentu bylo vyzkoušet, jak si metoda povede při odhalování vysoce perzistentní časové řady. Dále by bylo vhodné vyzkoušet, jak si tato metoda poradí s opačným extrémem, tedy jak moc úspěšná bude při odhalování vysoce antiperzistentní řady.

2.3 Alternující řada

Nyní provedeme R/S analýzu na uměle vytvořené časové řadě, tvořené opět 2048 hodnotami (tedy počtem hodnot 2^{11}), ovšem hodnoty této časové řady budou rovny $1, -1, 1, -1, 1, \dots$, bude se tedy jednat o prvních 2048 členů posloupnosti $y_n = (-1)^n$, pro $n = 1, 2, 3, \dots, 2048$. Tuto řadu si necháme vygenerovat v programu R pomocí příkazu `rep(c(-1, 1), 1024)`, kde první zadávaný parametr, je vektor hodnot, které chceme generovat a druhým parametrem je počet, kolikrát chceme dvojici hodnot -1 a 1 generovat. Bude se tedy jednat o časovou řadu délky 2048. Očekávaný odhad Hurstova exponentu je v tomto případě hodnota blízká dolní hranici intervalu hodnot, kterých může Hurstův exponent nabývat (opačné hodnoty než-li v předchozím případě), tedy 0. Provedl jsem tedy na této časové řadě R/S analýzu a získal jsem hodnotu Hurstova koeficientu 0.021, tuto hodnotu jsem následně uložil do proměnné `HurstAlt`. Algoritmus opět naplnil mé očekávání a úspěšně odhalil silnou antiperzistenci časové řady. Na závěr si ještě vykreslíme graf výsledné logaritmické regrese i s předpisem regresní funkce.

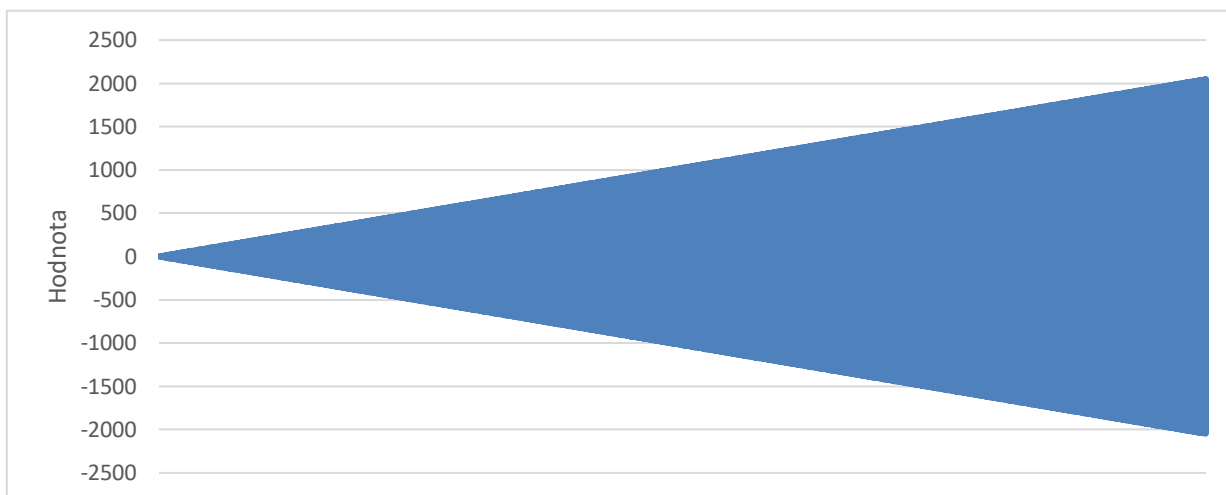


Obrázek 6: Logaritmická regrese

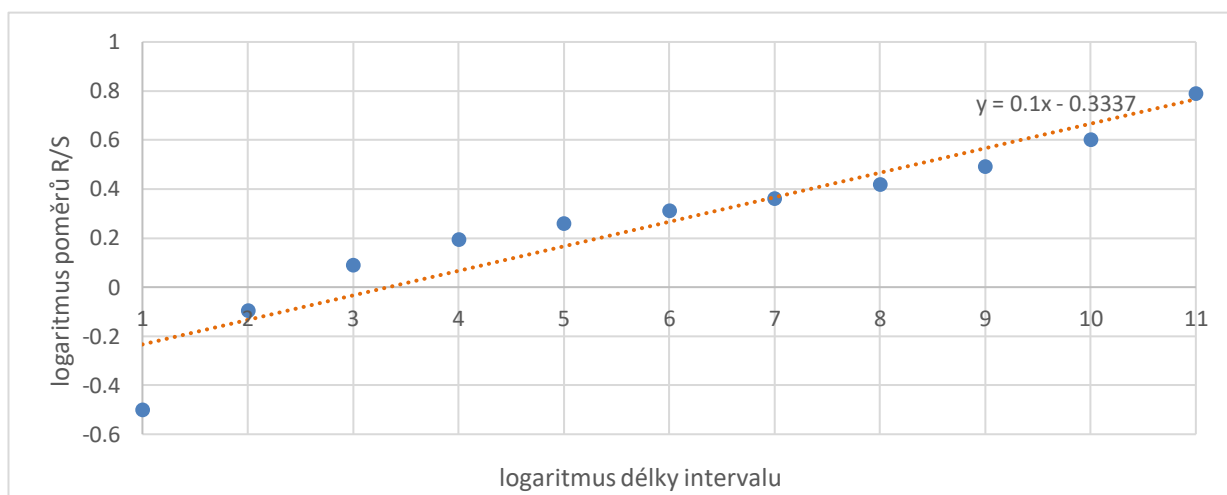
Dále jsem vyzkoušel, jakým způsobem R/S analýza zareaguje na změnu rozsahu generovaných hodnot. Vytvoříme si tedy časovou řadu stejným způsobem jako předchozí, avšak s tím rozdílem, že funkci generující posloupnost hodnot vynásobíme pro každý člen tisícem. Časová řada bude v tomto případě nabývat střídavě hodnoty -1000 a 1000. V tomto případě vyšel Hurstův koeficient roven hodnotě 0.034, tuto hodnotu jsem uložil do proměnné `Hurstalt1000`. Tento výsledek nás opět nepřekvapil, jelikož tato řada má, co se její perzistence týče, totožné vlastnosti jako řada předchozí. Vykreslovat si grafy není v tomto případě nutné.

2.4 Alternující řada se zvyšujícím se rozsahem

V této podkapitole provedu zobecnění tohoto experimentu a nechám si vygenerovat hodnoty časové řady získané podle předpisu $y_n = n(-1)^n$. Hodnoty tedy nejen, že budou měnit znaménko, ale budou se zároveň v absolutní hodnotě zvětšovat (rozsah hodnot se bude zvětšovat). Hodnoty této časové řady a graf logaritmické regrese pro přeškálované rozsahy si vykreslíme do následujících grafů:



Obrázek 8: časová řada HurstAlt1

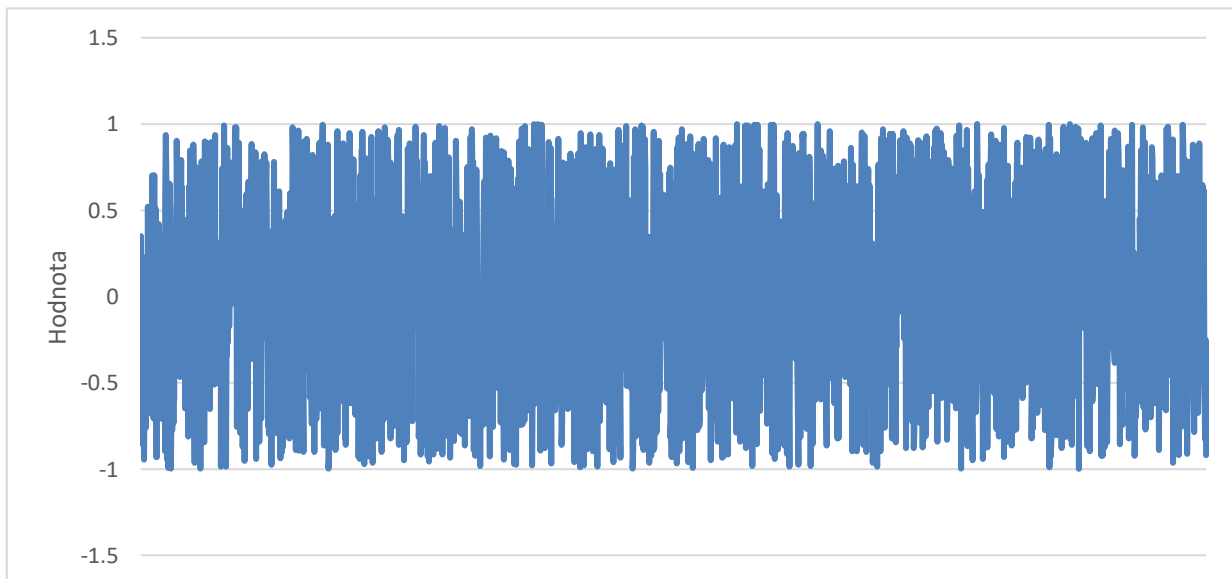


Obrázek 7: logaritmická regrese

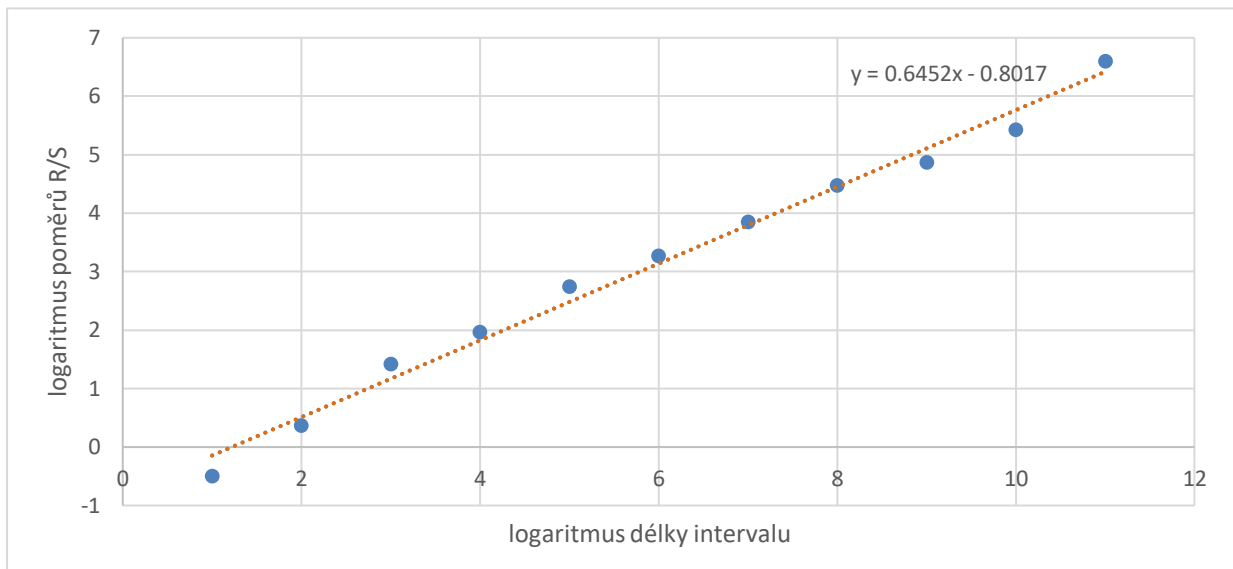
Hodnota směrnice regresní přímky vyšla v tomto případě 0.1. Tedy o něco málo vyšší než v předchozím případě, avšak stále dosti nízká. Tento experiment tedy splnil má očekávání. Výslednou hodnotu jsem si uložil do proměnné `HurstAlt1`. Na prvním obrázku si můžeme všimnout, že křivka vyplňuje rovinu dost „hustě“, dokonce tak, že se zdá, že je vykreslen celý trojúhelník, tedy 2D útvar, nikoliv 1D křivka. Fraktální dimenze bude proto blízká 2. To odpovídá hodnotě Hurstova koeficientu blízké nule.

2.5 Rovnoměrně rozdělená

V této části jsem použil algoritmus na spočtení Hurstova exponentu pro číselnou řadu tvořenou hodnotami získanými z rovnoměrného rozdělení (pomocí příkazu `runif(N, min = -1, max = 1)`, kde první parametr určuje počet generovaných hodnot, druhá hodnota určuje dolní hranici intervalu, ze kterého chceme hodnoty generovat a poslední hodnota nám určuje horní hranici). Jelikož jsme opět získali soubor nezávislých stejně rozdělených hodnot (pravděpodobnost, že po nízké hodnotě bude následovat hodnota vysoká je stejná, jako pravděpodobnost, že po nízké hodnotě bude následovat opět hodnota nízká), a data by tedy měla být generována náhodně ze zadaného intervalu $(-1,1)$. Očekávám, že Hurstův exponent vyjde opět blízky hodnotě 0.5 a bude tedy opět indikovat bílý šum. Nechal jsem si tedy opět vygenerovat 2048 hodnot z tohoto rozdělení. Hurstův exponent v tomto případě vyšel 0.645 a byl uložen do proměnné `HurstRovn`. Jak vidíme, výsledek pokusu je v souladu s očekáváním (opět mírně nadsazen). Na následujících obrázcích můžeme vidět dva grafy, první pro hodnoty této časové řady a druhý pro závěrečnou logaritmickou regresi.



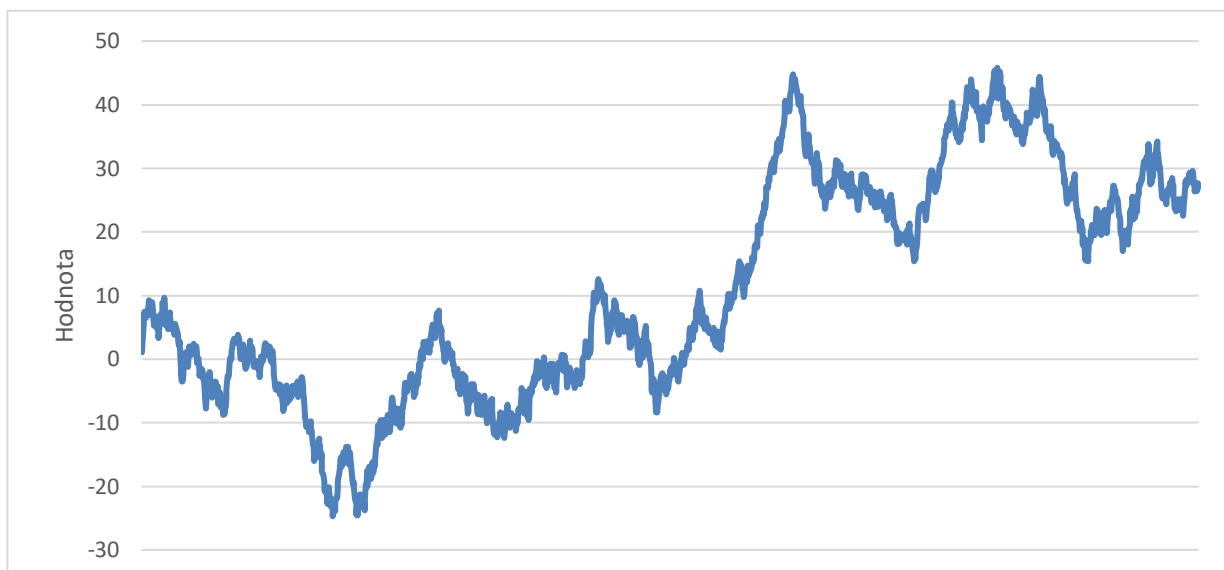
Obrázek 9: č.ř. tvořená hodnotami z rovnoměrného rozdělení



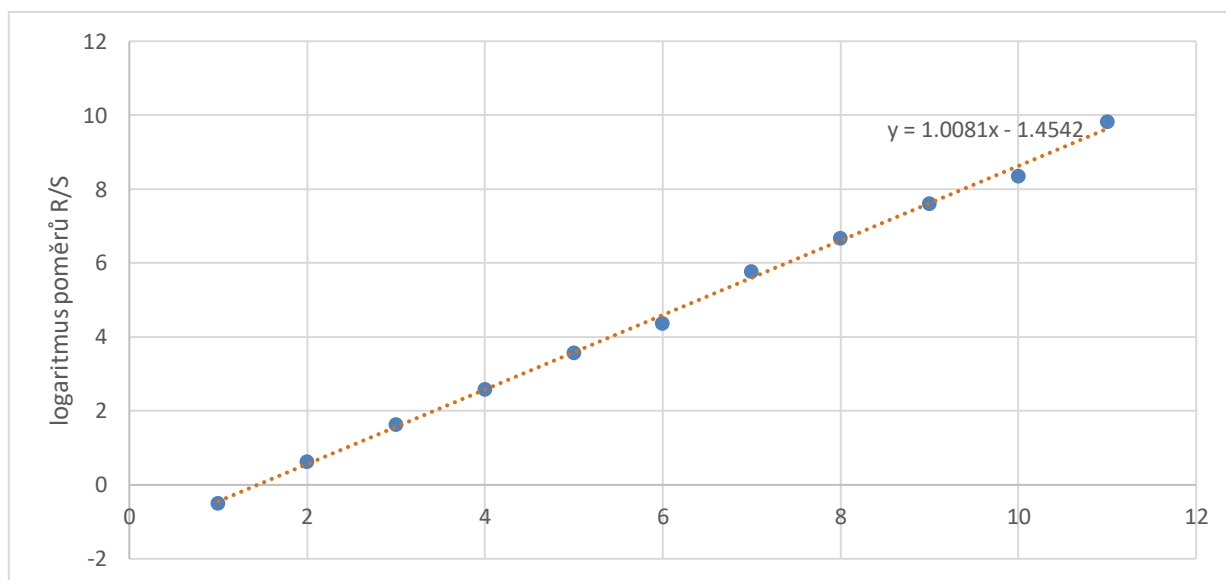
Obrázek 10: logaritmická regrese

2.6 Kumulované součty

V této podkapitole si vyzkoušíme, jakým způsobem nám R/S analýza vyhodnotí časovou řadu tvořenou kumulativními součty hodnot generovaných z normálního rozdělení s nulovou střední hodnotou a jednotkovým rozptylem. Hlavní rozdíl, oproti časové řadě s nezávislými hodnotami, které také pocházeli z normálního rozdělení se stejnými parametry je skutečnost, že každá z generovaných hodnot vychází z místa, kde předchozí hodnota končila. Tedy každá následující hodnota je přírůstkem hodnoty předchozí. Z této skutečnosti je již zřejmé, že mezi jednotlivými hodnotami existuje určitá souvislost, a tudíž hodnoty jsou na sobě závislé. Taková časová řada již nebude odpovídat bílému šumu, ale náhodné procházce. Časovou řadu vytvořím pomocí příkazu `ts(cumsum(rnorm(N, 0, 1)))`. Výsledná hodnota Hurstova exponentu pro tuto řadu vychází 1.008, tedy přibližně rovna hodnotě 1. Vidíme tedy, že R/S analýza odhalila v tomto případě, že mezi daty existuje určitý vztah. Nyní si vykreslíme graf takto generované časové řady společně s grafem závěrečné logaritmické regrese.



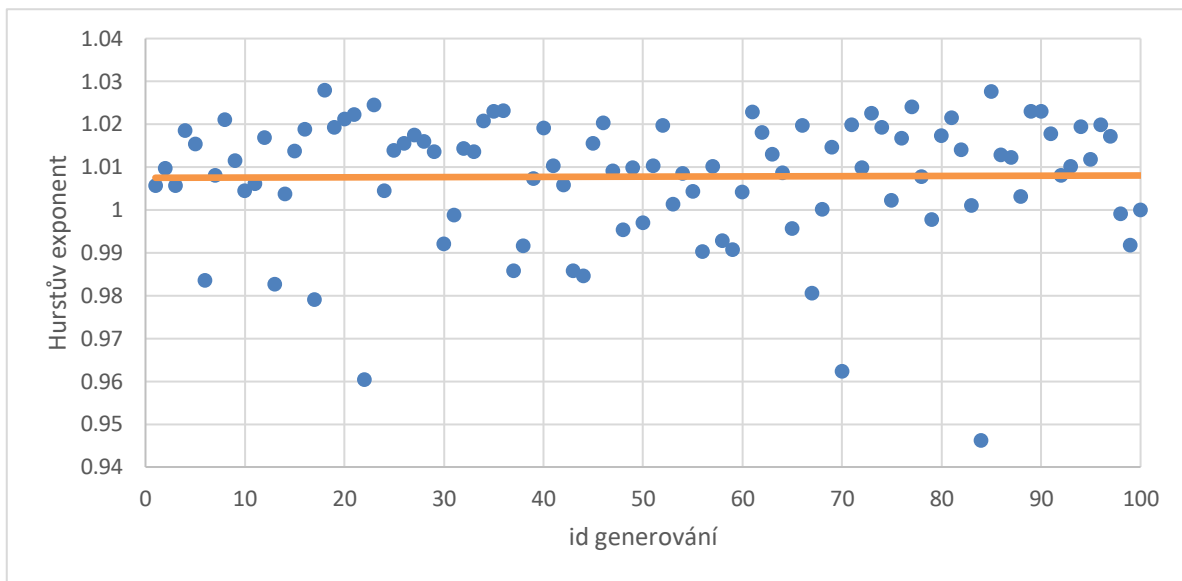
Obrázek 11: č.ř. kumulovaných součtů hodnot $N(0,1)$



Obrázek 12: logaritmická regrese

Na prvním grafu si můžeme všimnout rostoucího trendu této časové řady. Na druhém grafu můžeme vidět logaritmickou regresi i s předpisem regresní funkce.

Mohlo by být zajímavé prozkoumat, jak budou odhady Hurstových exponentů vycházet pro více sad generovaných pomocí kumulovaných součtů hodnot z normálního rozdělení. Tímto způsobem si vygenerujeme 100 sad po 2048 hodnotách, pro každou sadu spočítáme odhad Hurstova exponentu a všechny získané hodnoty uložíme do vektoru `Hurstcumsum100`. Zjistíme, že aritmetický průměr hodnot tohoto vektoru je 1.008. Na závěr si vykreslíme tyto hodnoty do bodového grafu.



Obrázek 13: Hurstův exponent pro Hurstcumsum100

Z grafu si můžeme všimnout, že se hodnoty Hurstova exponentu pro časové řady generované tímto způsobem, pohybují okolo hodnoty 1, avšak najdeme zde i odlehlejší hodnoty směrem k hodnotám nižším. V jednom případě vyšla hodnota Hurstova exponentu 0.946, ovšem většina hodnot se pohybuje poblíž aritmetického průměru, jehož hodnotu zde reprezentuje oranžová čára.

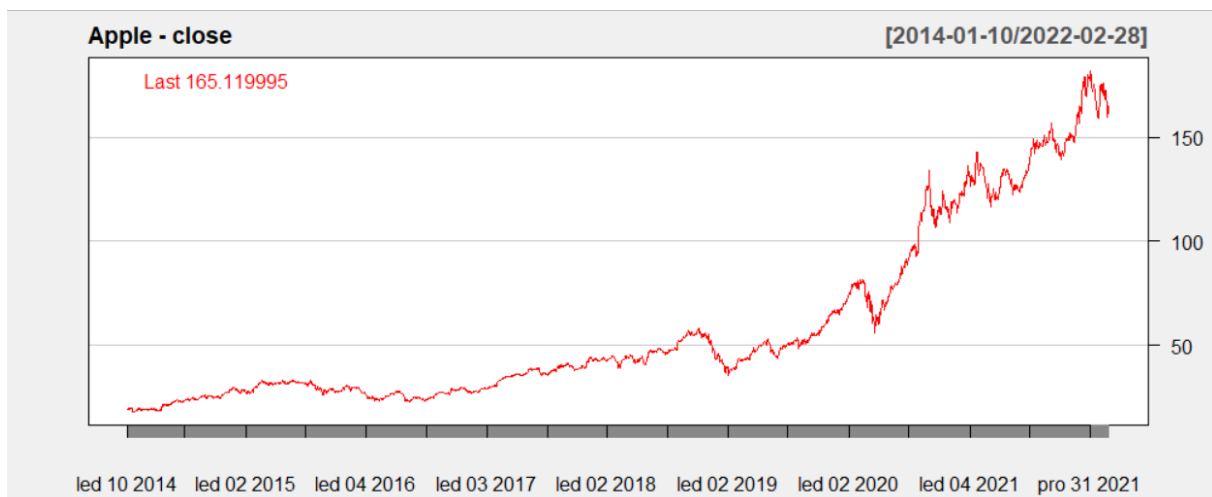
2.7 Reálná data

V této podkapitole použiji algoritmus na reálných časových řadách, konkrétně na řadách `AAPL.Open` a `AAPL.Close`[11], což jsou soubory dat o ceně akcií Applu v každém jednom pracovním dni v rozmezí od 10.1.2014 po 1.3.2022 (ze začátku a z konce dne). Toto časové rozmezí jsem vybral z toho důvodu, že počet pracovních dní, kdy se ceny na burze mohli měnit, je roven mocnině 2, konkrétně máme tedy záznamy pro $2^{11} = 2048$ dní. Mé očekávání bylo, že se hodnoty Hurstova koeficientu pro tuto časovou řadu budou blížit „náhodné procházce“ a skutečně ukázalo se, že vývoj hodnot akcií Applu (ať už `Open` nebo

Close) vycházejí podobně jako pro náhodnou procházku. Nejdříve si vykreslím grafy vývoje těchto cen:



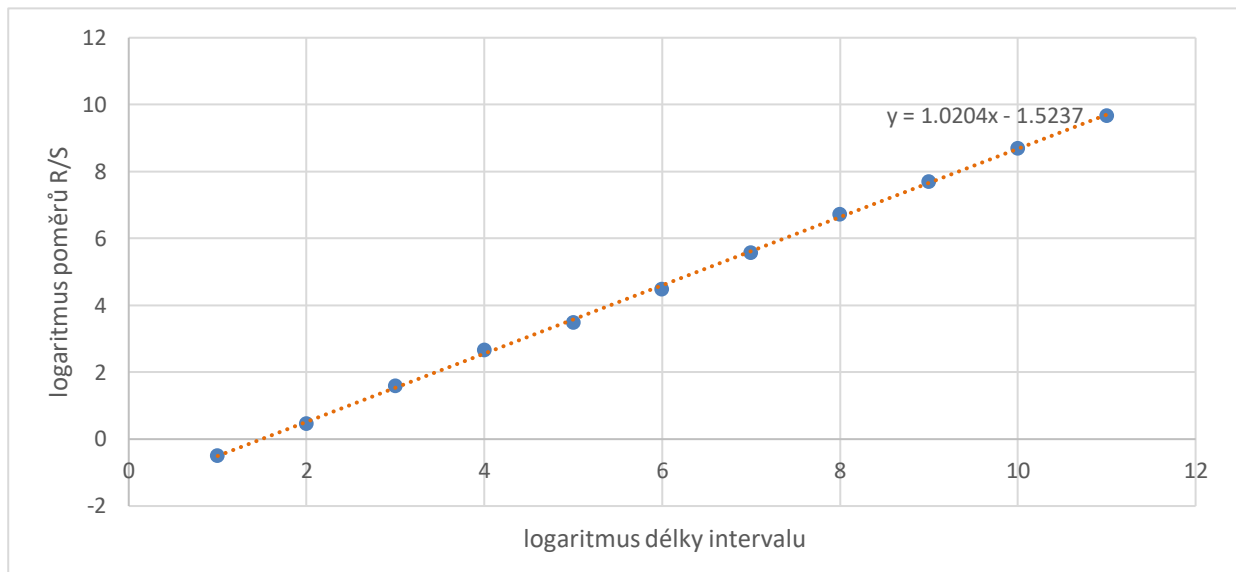
Obrázek 14: Graf časové řady cen akcií Applu (ze začátku dne)



Obrázek 15: Graf časové řady cen akcií Applu (z konce dne)

Odhad hodnoty Hurstova koeficientu pro řadu `AAPL.Open` vyšel 1.020 a pro řadu `AAPL.Close` to bylo 1.019 , tato data si uložím do proměnných `HurstOpen` a `HurstClose`. Jak vidíme, Hurstův exponent nám vyšel pro řady, které by měly být náhodné, velice vysoký a zároveň téměř stejný pro obě řady. Tento výsledek odpovídá tomu, že nám R/S analýza je schopna odhalit závislost hodnot časové řady. Můžeme tedy říci, že obě řady vykazují vysoce perzistentní chování (drží se rostoucího trendu). Mírné převýšení

oboru hodnot pro Hurstův exponent připisují opět použitému způsobu analýzy. Na závěr si vykreslíme graf logaritmické regrese pro data `AAPL.Open`.



Obrázek 16: logaritmická regrese Apple Open

Na tomto grafu si můžeme všimnout, že hodnota Hurstova exponentu skutečně odpovídá hodnotě směrnice regresní přímky.

Nyní, abych si ověřil, že řada `AAPL.Close` (případně `AAPL.Open`) je skutečně natolik perzistentní, náhodně permutuji data `AAPL.Close`, čímž zákonitě musím porušit pozorovanou perzistenci. Tuto permutaci provedu 10 možnými způsoby a pro každý způsob spočítám Hurstův exponent. Tyto exponenty vyšly postupně pro každou z 10 náhodných permutací: 0.593; 0.652; 0.605; 0.549; 0.607; 0.623; 0.597; 0.562; 0.634; 0.545. Jak si můžeme všimnout, Hurstův exponent vyšel ve všech případech dle očekávání, a sice blízky hodnotě 0.5. Výsledné hodnoty jsem uložil do vektorové proměnné `CloseSchuff`, která má tedy délku 10. Na závěr ještě spočítám aritmetický průměr těchto hodnot, tato hodnota vyšla 0.597. Chování této řady již odpovídá chování bílého šumu. Na základě těchto zjištění usuzuji, že časové řady akcií Apple mají dlouhodobou paměť a vykazují skutečně silně perzistentní chování. Zřejmě se tedy nejedná o chybu algoritmu.

2.8 Shrnutí výsledků

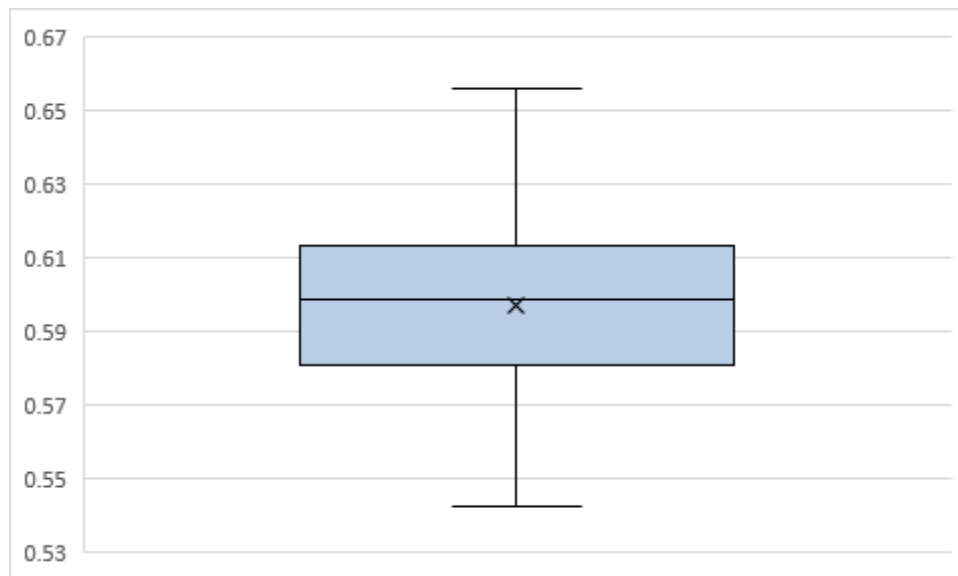
V této sekci jsem přehledně do tabulky shrnul všechny vypočtené hodnoty Hurstova exponentu pro každou řadu, kterou jsme nyní zkoumali (`HurstOpen`, `HurstClose`, `CloseSchuff`, `HurstAlt`, `Hurstlin`, `Hurstnorm` a `Hurstrovn`, `Hurstcumsum100`). Vzhledem k tomu, že některé vektory obsahovali hodnot 100, do této tabulky jsem vybral prvních 10 hodnot. Následně jsem si již mohl vypsát výsledné hodnoty přehledně do následující tabulky:

| HurstOpen | Hurstcumsum100 | CloseSchuff | HurstAlt | Hurstlin | Hurstnorm | Hurstrovn |
|-----------|----------------|-------------|----------|----------|-----------|-----------|
| 1.020 | 1.006 | 0.593 | 0.034 | 1.022 | 0.560 | 0.607 |
| | 1.009 | 0.652 | | | 0.605 | |
| | 1.006 | 0.605 | | | 0.591 | |
| | 1.019 | 0.549 | | | 0.577 | |
| | 1.015 | 0.607 | | | 0.566 | |
| | 0.984 | 0.623 | | | 0.574 | |
| | 1.008 | 0.597 | | | 0.603 | |
| | 1.021 | 0.562 | | | 0.593 | |
| | 1.012 | 0.634 | | | 0.633 | |
| | 1.005 | 0.545 | | | 0.610 | |

Tabulka 1: Výsledky experimentů

2.9 Boxplot rozdělení Hurstových exponentů

Jeden z pokusů popsaných v předchozí části spočíval v generování 10 sad hodnot z normálního rozdělení a následném spočítání Hurstových exponentů pro časové řady tvořené generovanými hodnotami (zajímalo nás, jak budou hodnoty Hurstova koeficientu rozděleny). Nyní tento pokus rozšířím, a to tím způsobem, že si nechám vygenerovat 100 sad po 2048 hodnotách z rozdělení $N(0,1)$, pro každou sadu si spočítám Hurstův koeficient a všechny získané koeficienty uložím do vektoru `Hurstnorm100`. Následně si ověřím, jak bude vypadat rozdělení hodnot v tomto vektoru. Myslím si, že tento pokus má smysl především z důvodu, že pro 100 hodnot je již vhodné vykreslit si boxplot. Výsledné hodnoty tohoto experimentu nebudu vypisovat všechny, bude stačit určit aritmetický průměr hodnot vektorové proměnné `Hurstnorm100`. Aritmetický průměr získáme pomocí příkazu `mean(Hurstnorm100)`, který nám prozradí, že hodnota průměru je 0.597 a pro lepší vizualizaci si necháme vykreslit zmiňovaný boxplot. Boxplot vypadá následovně:



Obrázek 17: Boxplot hodnot Hurstnorm100

Pro přehlednější shrnutí přesných hodnot číselných charakteristik použijeme funkci `summary(Hurstnorm100)`. Tato funkce nám vrátí některé z číselných charakteristik popisující rozdělení hodnot v tomto vektoru. Shrnu je do následující tabulky:

| | |
|-----------------|------|
| minimum | 0.54 |
| 1. kvartil | 0.58 |
| medián | 0.60 |
| střední hodnota | 0.60 |
| 3. kvartil | 0.61 |
| maximum | 0.66 |

Tabulka 2: číselné charakteristiky Hurstnorm100

Vidíme, že výsledné hodnoty jsou opět v souladu s naším očekáváním.

V této části bylo popsáno několik experimentů, provedených ať už na časové řadě tvořené uměle generovanými daty z nějakého pravděpodobnostního rozdělení, alternující řadě a řadě generované lineární funkcí. Dále jsme si také vyzkoušeli, jaké hodnoty nám algoritmus vrátí pro skutečná data (časovou řadu hodnot akcií Applu). V další části se kvůli zjednodušení budu věnovat tvorbě funkcí, které nám budou počítat Hurstův koeficient pro časové řady, tvořené hodnotami z různých pravděpodobnostních rozdělení, na základě námi zadaného parametru.

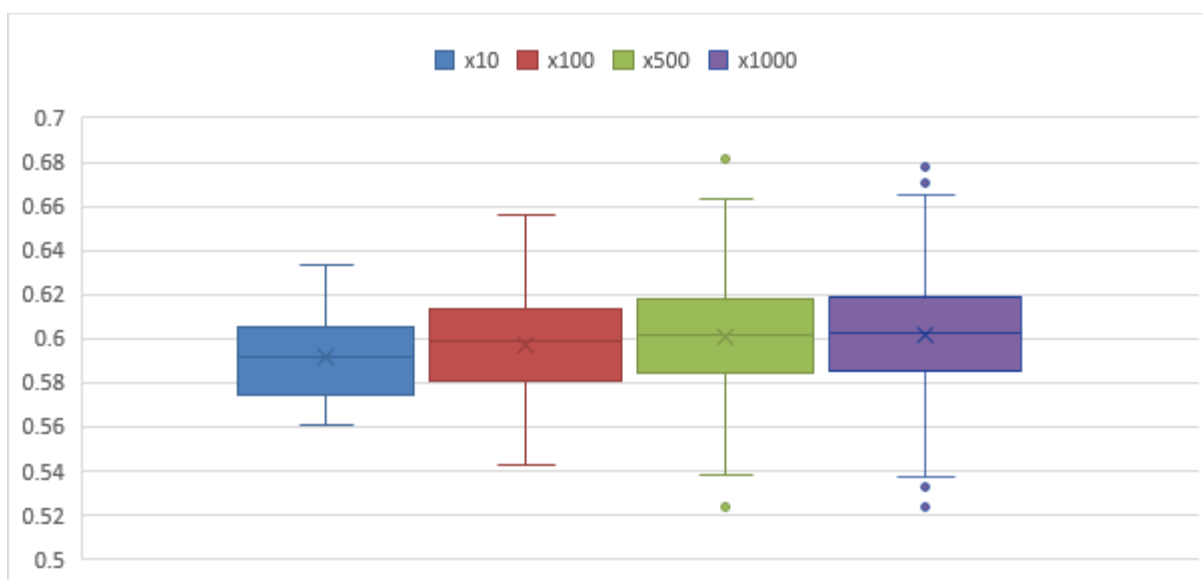
3 Funkce

Pro tvorbu funkcí jsem si vytvořil nový skript, který jsem nazval „Funkce.R“. Zde budu vytvářet a ukládat zdrojové kódy pro různé funkce, které nám usnadní experimenty. V úvodní části je opět použit příkaz `getSymbols`, díky kterému získáme zmiňovaná data o akcích Applu. Jednotlivé podkapitoly budou pojmenovány podle popisované funkce

3.1 HurstN

První vytvořenou funkci jsem nazval `HurstN(k)`, jedná se o funkci jednoho argumentu (k). Tato funkce umí generovat sady po 2048 hodnotách z normálního rozdělení (s nulovou střední hodnotou a jednotkovým rozptylem), pro které spočítá Hurstův exponent.

Argument funkce určí, kolik sad hodnot chceme vygenerovat, a tedy pro kolik sad chceme spočítat Hurstův koeficient. Všechny výsledky, tedy Hurstův koeficient pro každou sadu, nám funkce následně uloží do vektorové proměnné `Hurstnormk` a na závěr všechny získané hodnoty vypíše. Tuto funkci jsem dále využil ke zkoumání rozdělení hodnot výsledných exponentů pro hodnoty parametru $k = 10, 100, 500, 1000$. Tedy rozdělení hodnot Hurstových exponentů pro 10, 100, 500 a 1000 sad. Pro každý z těchto 4 vektorů, jsem vykreslil boxplot rozdělení hodnot těchto vektorů. Všechny čtyři boxploty jsem pro lepší srovnání vykreslil do jednoho grafu. Tento experiment nám pomůže získat představu o chování Hurstových koeficientů pro rostoucí počet sad generovaných hodnot.

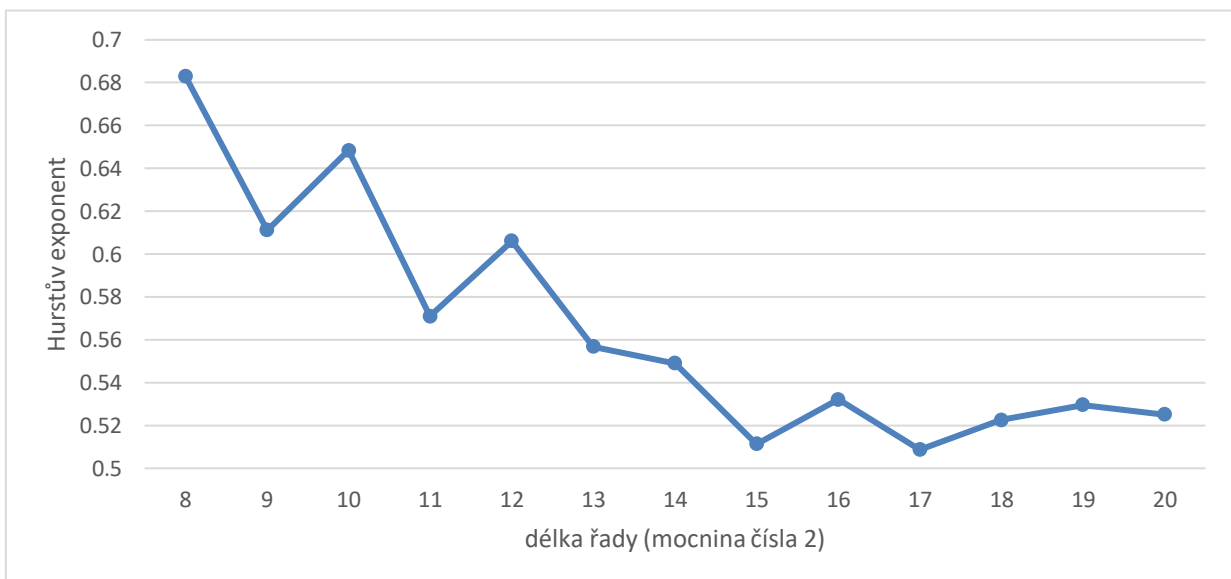


Obrázek 18: Boxploty pro 10, 100, 500 a 1000 generování

Z boxplotů můžeme zjistit, že s rostoucím počtem generovaných sad se zvětšuje rozptyl a pro počty 500 a 1000 se nám již objeví i odlehlé hodnoty. Také si můžeme povšimnout velice mírného růstu mediánu hodnot k hodnotě 0.6.

3.2 HurstNL

Druhou vytvořenou funkci jsem pojmenoval `HurstNL`. Opět se jedná o funkci jednoho argumentu. Tato funkce vygeneruje již pouze jednu sadu hodnot z $N(0,1)$. Argument k nám v tomto případě určí počet generovaných hodnot (jelikož se omezují na počty hodnot o mocnině čísla 2, argument nám určuje mocninu dvou, tedy 2^k , rozumné je tedy zadávat argument k z rozmezí, řekněme, od 8 do 20). Zvláštností je, že počet dělení intervalů jsem nechal stejný jako v předchozím případě, a sice $2^{11}, 2^{10}, 2^9, \dots, 2^0$, není tedy závislý na hodnotě parametru k . S touto funkcí jsem provedl experiment, ve kterém jsem zkoumal závislost vývoje Hurstova exponentu na počtu generovaných hodnot. Nechal jsem si tedy spočítat Hurstův koeficient pro počty generovaných hodnot $2^8, 2^9, \dots, 2^{20}$, dohromady tedy získám 12 hodnot. Tyto hodnoty si následně vykreslím do dvou grafů (spojnicového a bodového):



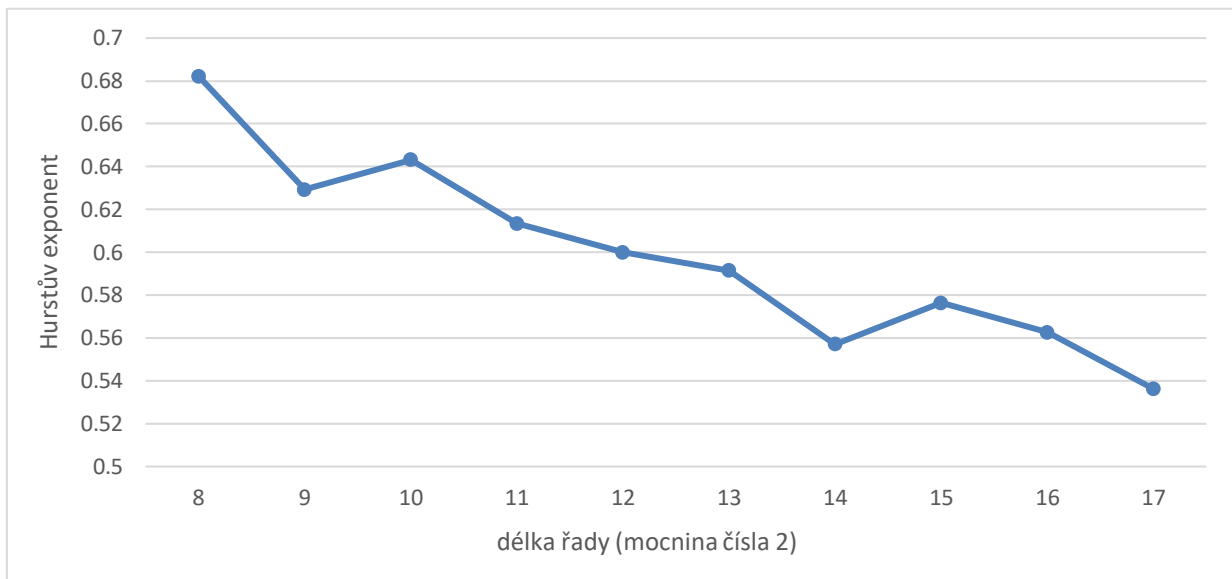
Obrázek 19: Hurstův koeficient pro počty generovaných hodnot 2^8 až 2^{20}

Z grafů si můžeme povšimnout tendence Hurstova exponentu, který, pro rostoucí počet generování, klesá k hodnotě 0.5, tedy k hodnotě, která svědčí o „bílém šumu“. Zdá se, že čím více hodnot máme, tím více se výsledek blíží očekávané hodnotě, což znamená větší přesnost odhadu.

3.3 HurstI

Další funkce je pouhým zobecněním předchozí funkce. Zobecnění spočívá v proměnlivosti počtu dělicích intervalů, ten je nyní také závislý na parametru k . Funkci jsem nazval `HurstI`. Pomocí této funkce jsem vypočítal hodnoty Hurstových exponentů pro počty generování k a zároveň počty dělicích intervalů $2^0, 2^1, \dots, 2^k$

Tuto funkci jsem nechal vykreslit grafy pro počty generování a počty dělicích intervalů rovným hodnotám $2^8, 2^9, \dots, 2^{17}$. Výsledné hodnoty jsem si opět nechal vykreslit v podobě spojnicového grafu:

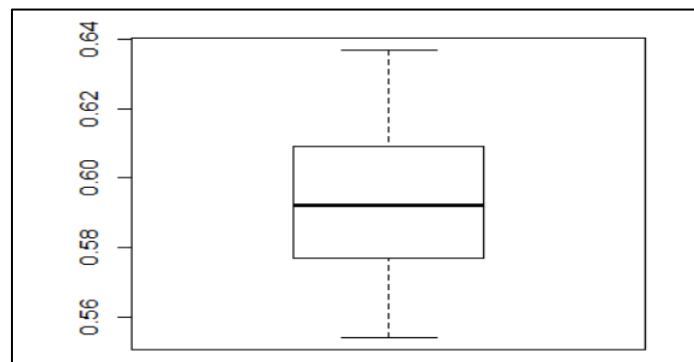


Obrázek 20: HurstI pro $k = 2^8, 2^9, \dots, 2^{17}$

Jediný rozdíl v porovnání s předchozím grafem je rychlejší konvergence hodnot Hurstova koeficientu k hodnotě 0.5.

3.4 Hurstbox

Funkci, kterou si v této podkapitole představíme, jsem nazval `Hurstbox`. Tato funkce funguje téměř stejným způsobem jako funkce `HurstN`, tedy generuje k sad po 2048 hodnotách z $N(0,1)$ a pro každou sadu spočte Hurstův exponent. Rozdíl oproti předchozí funkci spočívá ve výstupu. Výstupem funkce `Hurstbox` je boxplot rozdělení hodnot vypočtených Hurstových exponentů. Díky této funkci získáme vizuální přehled o charakteristikách rozdělení Hurstových exponentů pro daný počet generování. Funkci si vyzkoušíme pro 30 generování. Její výsledek můžeme vidět níže:



Obrázek 21: Výstup funkce Hurstbox(30)

Boxplot vygenerovaný zmiňovanou funkcí odpovídá našemu očekávání a ukazuje, že se hodnoty Hurstových exponentů pro 30 sad pohybují blízko hodnoty svědčící pro bílý šum.

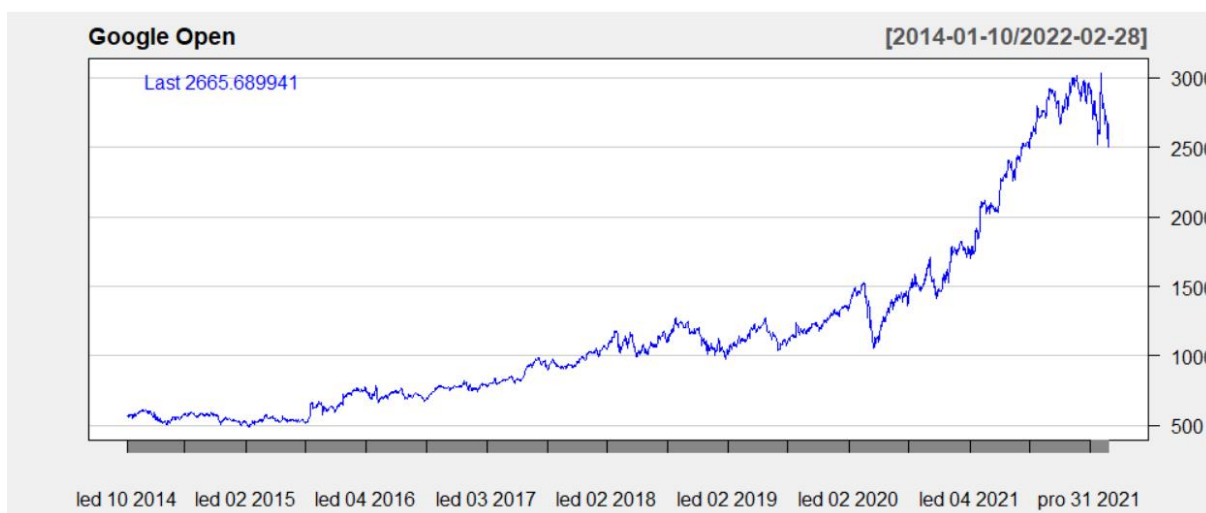
4 Analýza reálných dat

V této kapitole budu odhadovat Hurstovy exponenty reálných časových řad, konkrétně ekonomických časových řad. Budeme zkoumat časové řady vývoje cen akcií firem *Google*, *Facebook* (dnes již *Meta Platforms*), a nakonec vývoj hodnot akcií *Dow Jonesova indexu*. Vývoj cen akcií budeme zkoumat u všech těchto časových řad od data 1. 1. 2014 po datum 28. 2. 2022. Počet zkoumaných údajů je roven počtu pracovních dní v tomto rozmezí, tedy 2^{11} , což je 2048. Následující podkapitoly budou pojmenovány podle společností, jejichž akcie budeme analyzovat. V úvodu každé podkapitoly si každou společnost ve stručnosti představíme.

4.1 Google

Jako první provedeme R/S analýzu časové řady hodnot akcií společnosti Google. Google je americká společnost známá především díky svému internetovému vyhledávači. Tato společnost však poskytuje ještě mnoho dalších služeb, jako například vlastní webový prohlížeč *Google Chrome*, emailovou schránku *Gmail*. Dále pod Google spadá také *Android*, což je nejrozšířenější operační systém pro většinu dnešních chytrých telefonů nebo například *Google Earth*, což je virtuální online globus. Mezi další služby poskytované společností Google se řadí například *Překladač Google*, *Učebna Google*, *G Suite*, *Google Ads*, *Google AdSense*, a další [12].

Nyní se již budeme věnovat R/S analýze. Nejdříve si necháme vykreslit časovou řadu vývoje cen akcií na začátcích každého pracovního dne, a to pomocí příkazu `chartSeries` [12]. Hlavním argumentem této funkce je zkoumaná časová řada, dále můžeme zadat název vykreslovaného grafu, který bude vypsán vlevo nad graf. Můžeme upravit i ostatní atributy, jako je téma (můžeme určit, zda chceme graf tmavý nebo světlý) nebo barvu čáry vykreslující graf. Zmiňovaný graf této časové řady vypadá následovně:



Obrázek 22: Graf časové řady vývoje hodnot akcií Google

Již na první pohled je patrný rostoucí trend, poměrně silná perzistence a vzájemná závislost hodnot. Nyní si naši domněnku ověříme pomocí R/S analýzy. Ta nám v případě této řady vrátila hodnotu Hurstova exponentu rovnu 1.01 , což nám potvrzuje naši domněnku o silné perzistenci a dlouhodobé paměti této časové řady. Tuto hodnotu si uložíme do proměnné `googleOpen`. Můžeme tedy říct, že se časová řada drží rostoucího trendu a že se jedná o časovou řadu s dlouhodobou pamětí. Podívejme se na další časovou řadu.

4.2 Facebook (Meta Platforms)

V této části se podíváme na časovou řadu cen akcií společnosti *Facebook (Meta Platforms)*. Nejdříve si tuto firmu opět ve stručnosti představíme. Meta Platforms je známá především díky své službě sociálních sítí (Facebook). Pod tuto značku spadá též například aplikace *Messenger*, což je platforma, která poskytuje možnost komunikace na dálku (chat, hovor, videohovor) a sociální síť *Instagram*. Pod tuto značku spadá i mnoho dalších aplikací. Tuto službu aktivně využívá přibližně 2.5 miliardy uživatelů (data z 2/2020) [13].

Opět si v úvodu vykreslíme časovou řadu vývoje cen akcií této společnosti. Ceny akcií v jednotlivých dnech pochází ze začátku každého pracovního dne. Tuto řadu si vykreslíme stejným způsobem jako řadu předchozí. Graf této časové řady vypadá takto:



Obrázek 23: Graf časové řady vývoje hodnot akcií Facebooku

Z grafu je poměrně zřetelné, že se časová řada drží rostoucího trendu (až na strmý pád na konci). Na první pohled si můžeme všimnout výrazně rozkmitanějšího vývoje cen akcií této společnosti v porovnání s vývojem cen akcií Googlu. Provedeme tedy R/S analýzu a určíme odhad Hurstova exponentu. Ten pro tuto řadu vyšel 1.017 . Tedy opět poměrně vysoký. O řadě můžeme říct, že je opět silně perzistentní a má dlouhodobou paměť.

4.3 Dow Jones Industrial Average (DJIA)

U nás též nazývaný jako *Dow Jonesův index*, je jeden z nejznámějších ukazatelů vývoje cen akcií na americkém akciovém trhu. Jedná se o druhý nejstarší index. První záznam tohoto indexu pochází ze dne 26. května 1896, kdy tento index obsahoval data o akciích dvanácti zejména průmyslových společností. Nyní se tento index skládá z akcií 30 amerických společností, jejichž akcie jsou nejvýznamnější na americkém akciovém trhu. Hodnota tohoto indexu je počítána jako vážený průměr hodnot akcií obsažených v tomto indexu. Váhy jsou jednotlivým společnostem přidělovány podle ceny jejich akcií, jedná se tedy o cenově vážený ukazatel. Čím vyšší je cena, za kterou se akcie určité společnosti na trhu prodávají, tím vyšší váhu má tato společnost v rámci tohoto indexu. Mezi nejznámější společnosti, jejichž akcie jsou v tomto indexu obsaženy jsou například *Apple*, *Boeing*, *Caterpillar*, *Coca-Cola*,

Microsoft, Nike. Všechny společnosti obsažené v tomto indexu se obchodují buďto na burze NYSE (New York Stock Exchange) nebo NASDAQ [14].

Nyní přejdeme k R/S analýze. Nejdříve si opět necháme, vykreslit graf časové řady vývoje cen tohoto indexu. Konkrétně ceny ze začátku každého pracovního dne, kdy se na burzách obchodovalo. Budu zkoumat hodnoty z rozmezí 2. 5. 2014 po 2. 6. 2022. Počet zkoumaných hodnot bude tedy opět odpovídat mocnině čísla 2, konkrétně $2^{11} = 2048$. V tomto případě budou na ose y opět hodnoty akcií, avšak v tomto případě budou tyto hodnoty v tisících USD. Graf této řady vypadá takto:



Obrázek 24: č.ř. hodnot akcií Dow Jonesova Indexu (v tisících USD)

Tato řada má ještě o něco méně zřetelný rostoucí trend než řada předchozí. Můžeme si také povšimnout jednoho výrazného strmého poklesu někdy zhruba v květnu roku 2020. Dále vidíme, že se tato řada, téměř po celou zkoumanou dobu drží rostoucího trendu až na její poslední část, kde nastává mírný pokles až k hodnotě 29 912 USD pro 2. 6. 2022. Odhad hodnoty Hurstova exponentu v tomto případě vyšel 1.024 . Tato hodnota nám opět indikuje, že se jedná o silně perzistentní řadu, která se drží rostoucího trendu a jejíž hodnoty jsou na sobě navzájem závislé.

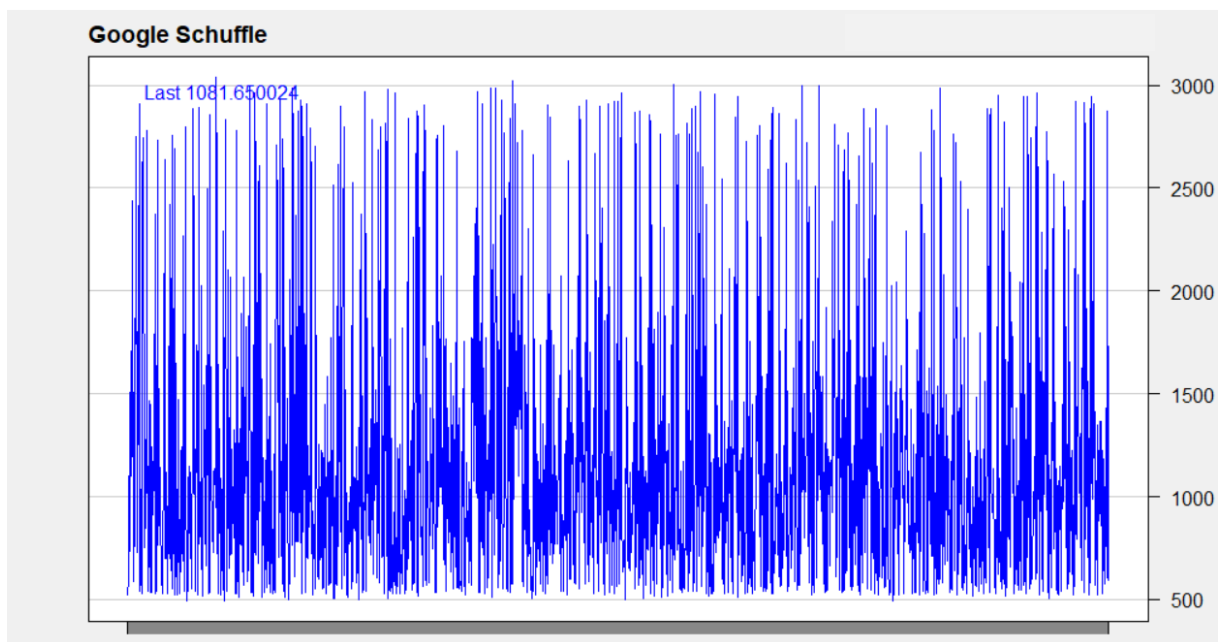
4.4 Shrnutí

U všech zkoumaných časových řad vyšel koeficient poměrně vysoký. Shrňme si výsledky odhadů Hurstových koeficientů pro každou z těchto tří řad přehledně do tabulky

Tabulka 3: Shrnutí vypočtených Hurstových koeficientů

| Apple | Google | Meta Platforms | Dow Jonesův index |
|-------|--------|----------------|-------------------|
| 1.020 | 1.010 | 1.017 | 1.024 |

Dále mě napadlo, opět náhodně permutovat data v časové řadě a vytvořit tak novou časovou řadu. Tím vlastně porušíme dlouhodobou paměť původní časové řady. A tedy Hurstův koeficient by měl tuto časovou řadu vyhodnotit jako bílý šum. Nejprve si opět vykreslíme graf této časové řady:



Obrázek 25: Graf časové řady náhodně permutovaných dat o hodnotě akcií Google

Z grafu si můžeme všimnout, že se z časové řady skutečně vytratil rostoucí trend. Větší hustotu dat v oblasti nižších hodnot bych přisuzoval tomu, že původní řada obsahovala více hodnot nízkých v porovnání s počtem hodnot vysokých. Podívejme se nyní, jaký odhad Hurstova exponentu nám pro tuto řadu vyjde. Po provedení R/S analýzy této řady zjistíme, že jeho odhad je roven 0.593 . Tato hodnota odpovídá skutečnosti, že jsme náhodnou permutací dat porušili dlouhodobou paměť časové řady, a výsledná křivka časové řady skutečně odpovídá bílému šumu.

Závěr

Hlavním cílem této bakalářské práce bylo seznámit čtenáře se statistickou metodou pro zkoumání dlouhodobých časových řad, R/S analýzou, a s jejím výstupem, Hurstovým exponentem. Dozvěděli jsme se, že tato metoda funguje na principu hledání poměru mezi rozsahem a odchylkou časové řady. Ukázali jsme, jak nám u této metody analýzy časových řad může pomoci programovací jazyk R. Víme již také, že pomocí této metody můžeme zkoumat perzistenci časové řady. Jinými slovy, tato metoda nám o časové řadě prozradí, zda má dlouhodobou paměť či nikoliv; tedy zda mezi hodnotami existuje vzájemná závislost nebo ne. Na teoretických příkladech jsme si ověřili její funkčnost a na praktických příkladech jsme si ukázali její využití při zkoumání časových řad vývoje hodnot akcií různých velkých společností.

Sekundárním cílem této bakalářské práce bylo zdokonalit se v programovacím jazyce R. Mohu potvrdit, že tento cíl byl v rámci této práce dosažen a nemohu vyloučit, že se v tomto jazyce budu zdokonalovat i nadále.

Myslím, že si tato metoda zasloužila pozornost, která jí byla věnována v této práci. Také se domnívám, že by si zasloužila větší pozornost odborníků, než se jí dostává, a to zejména z důvodu, že u této metody je stále co objevovat.

Seznam Obrázků

| | |
|--|----|
| Obrázek 1: Kochova křivka [7] | 12 |
| Obrázek 2: Mengerova houba[8]..... | 13 |
| Obrázek 3: Lineární č.ř. s náhodnou složkou s rozptylem 500 | 23 |
| Obrázek 4: Lineární č.ř. s náhodnou složkou s rozptylem 5000 | 24 |
| Obrázek 5: Lineární č.ř. s náhodnou složkou s rozptylem 50 000..... | 24 |
| Obrázek 6: Logaritmická regrese | 26 |
| Obrázek 7: logaritmická regrese | 27 |
| Obrázek 8: časová řada HurstAlt1 | 27 |
| Obrázek 9: č.ř. tvořená hodnotami z rovnoměrného rozdělení..... | 28 |
| Obrázek 10: logaritmická regrese..... | 29 |
| Obrázek 11: č.ř. kumulovaných součtů hodnot $N(0,1)$ | 30 |
| Obrázek 12: logaritmická regrese..... | 30 |
| Obrázek 13: Hurstův exponent pro Hurstcumsum100..... | 31 |
| Obrázek 14: Graf časové řady cen akcií Applu (ze začátku dne) | 32 |
| Obrázek 15: Graf časové řady cen akcií Applu (z konce dne)..... | 32 |
| Obrázek 16: logaritmická regrese Apple Open..... | 33 |
| Obrázek 17: Boxplot hodnot Hurstnorm100 | 35 |
| Obrázek 18: Boxploty pro 10, 100, 500 a 1000 generování | 36 |
| Obrázek 19: Hurstův koeficient pro počty generovaných hodnot 28 až 29..... | 37 |
| Obrázek 20: Hurstl pro $k = 28, 29, \dots, 217$ | 38 |
| Obrázek 21: Výstup funkce Hurstbox(30) | 39 |
| Obrázek 22: Graf časové řady vývoje hodnot akcií Google | 40 |
| Obrázek 23: Graf časové řady vývoje hodnot akcií Facebooku..... | 41 |
| Obrázek 24: č.ř. hodnot akcií Dow Jonesova Indexu (v tisících USD)..... | 42 |
| Obrázek 25: Graf časové řady náhodně permutovaných dat o hodnotě akcií Google..... | 43 |

Seznam použitých zdrojů

- [1] *Bílý šum* [online]. [cit. 2022-06-10]. Dostupné z:
https://cs.wikipedia.org/wiki/B%C3%ADI%C3%BD_%C5%A1um
- [2] PETERS, E. E.: *Fractal market analysis*. John Wiley and sons, Inc., New York, 1994. ISBN 0-471-58524-6.
- [3] *Metody pro odhad Hurstova exponentu* [online]. [cit. 2022-06-12]. Dostupné z:
<http://akson.sgh.waw.pl/~mo23628/art/ref1.pdf>
- [4] SPOTT, J.C.: *Chaos and Time- Series Analysis*. Oxford University Press, Oxford 2003. ISBN 0-19-850840-9
- [5] *Fractal dimension* [online]. [cit. 2022-06-13]. Dostupné z:
http://en.wikipedia.org/wiki/Fractal_dimension
- [6] HILLBORN R.C.: *Chaos and nonlinear dynamics: an introduction for scientists and engineers*. Oxford University Press, Oxford, 2000. ISBN 0-19-850723-2.
- [7] *Kochova křivka* [online]. [cit. 2022-06-14]. Dostupné z:
https://cs.wikipedia.org/wiki/Kochova_k%C5%99ivka#/media/Soubor:Koch_curve.svg
- [8] *Mengerova houba* [online]. [cit. 2022-06-14]. Dostupné z:
https://cs.wikipedia.org/wiki/Mengerova_houba#/media/Soubor:Menger-Schwamm-farbig.png
- [9] *Nadhodnocení odhadů metodou R/S* [online]. [cit. 2022-06-15]. Dostupné z:
http://mpira.ub.uni-muenchen.de/16446/1/MPRA_paper_16446.pdf
- [10] ZEMAN, Martin. *Hurstův exponent a náhodnost v časových řadách* [online]. Praha, 2010 [cit. 2022-06-15]. Dostupné z: <https://theses.cz/id/1qp9ed/>. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Jiří Trešl.

[11] JEFFREY A. RYAN and JOSHUA M. ULRICH (2022). *quantmod: Quantitative Financial Modelling Framework. R package version 0.4.20.*
<https://CRAN.R-project.org/package=quantmod>

[12] *Google* [online]. [cit. 2022-06-20]. Dostupné z:
<https://cs.wikipedia.org/wiki/Google>

[13] *Facebook* [online]. [cit. 2022-06-20]. Dostupné z:
<https://cs.wikipedia.org/wiki/Facebook>

[14] *Dow Jones Industrial Average* [online]. [cit. 2022-06-20]. Dostupné z:
https://cs.wikipedia.org/wiki/Dow_Jones_Industrial_Average