Modelovanie časových radov z monitorovacieho systému Hawkular

Bc. Pavol Loffay[1]

28. novembra 2015

Abstrakt: Práca spracováva predikciu časových radov prevziatych z monitorovacieho systému Hawkular[a]. Tento systém dokáže monitorovať Java aplikácie, alebo fyzický hardvér na ktom je spustený. Z množiny pozorovaných metrík bola vybratá vyťaženosť Java hromady (heap). Pre túto časovú rady boli zostrojené modeli: *ARIMA* a exponenciálne vyhladzovanie. Bola demonštrovaná adaptívna filtrácia pomocou algoritmu *LMS* na výpočet koeficientov *AR* modelu na generovaných dátach.

Kľúčové slová: Časová rada; ARIMA; LMS; Exponenciálne vyhladzovanie; Hawkular;

JEL klasifikácia: C53

[a]Dostupné na `http://www.hawkular.org`

1 Úvod

Monitorovanie doležitých business aplikácii medzi ktoré napríklad patria bankové systémy alebo rôzne servery, na ktorých bežia služby ktoré sú využívané 24/7 je veľmi dôležité. Vačšina monitorovacích systémov dokáže upozorniť administrátora na vyťaženie pri prekročení hraničnej hodnoty. V monitorovacom systéme Hawkular je možné zapnúť predikciu, takže administrátor bude upozornený vopred ak by náhodov malo dôjsť k prekročeniu spomenutej hraničnej hodnoty. Použíté modely časových rád v systéme Hawkular sú varianty exponenciálneho vyhladzovania. Modely *ARIMA* nemohli byť použité z dôvodu nestacionarity dát a náročnosti na výpočet – systém je schopný analyzovať aj niekoľko tisíc model naraz.

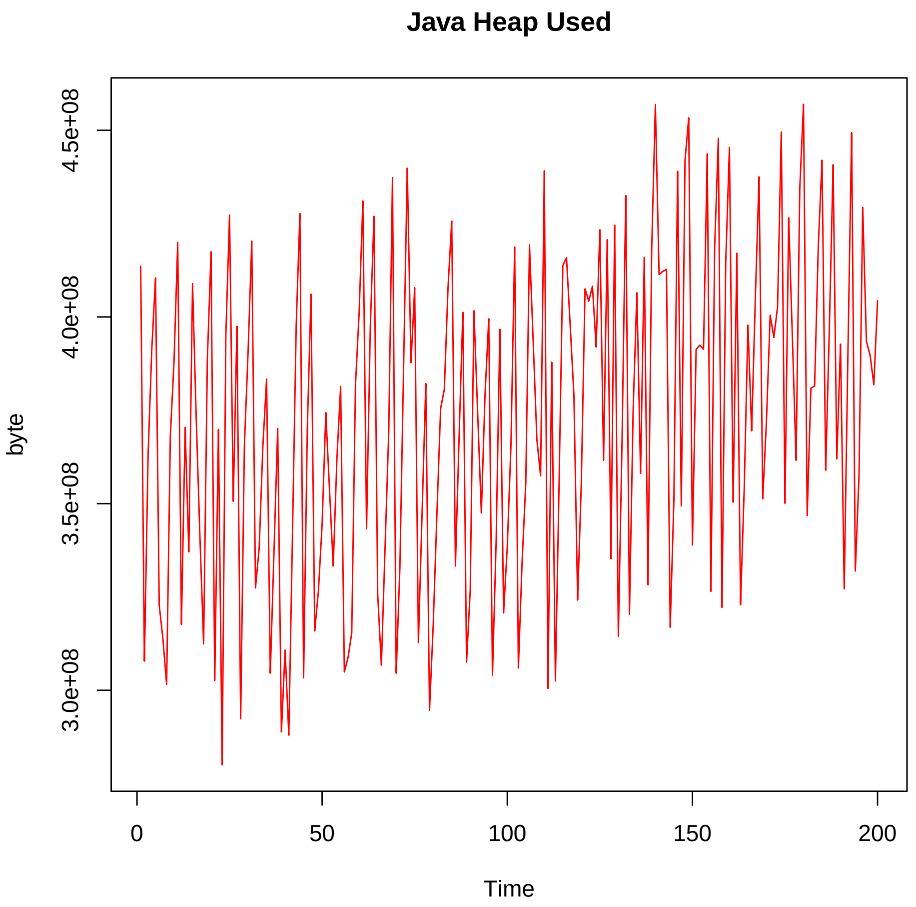
Táto práci sa zameria na konštrukciu optimálneho *ARIMA* modelu, ktorý následne porovná s exponenciálnym vyhladzovaním konkrétne Holtovou metódou s linárnym trendom. Následne bude demonštrovaná adaptívna filtrácia. Pre výpočetnú nenáročnosť bola použitá kombinácia exponenciálneho vyhladzovania a adaptívnej filtrácie v sýstéme Hawkular.

2 Identifikácia ARIMA modelu

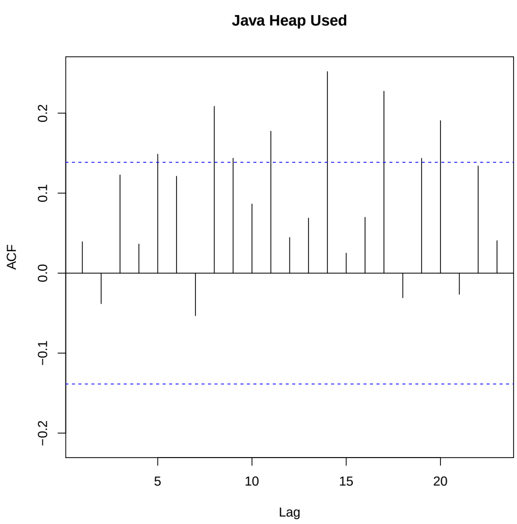
V tejto kapitole budeme analyticky identifikovať najvhodnejší *ARIMA* model, ktorý následne porovnáme s výsledom funkcie `auto.arima` z balíka `forecast` v jazyku *R*.

[1]Masarykova univerzita, Fakulta informatiky, obor: Service Science Management Engineering, p.loffay@mail.muni.cz

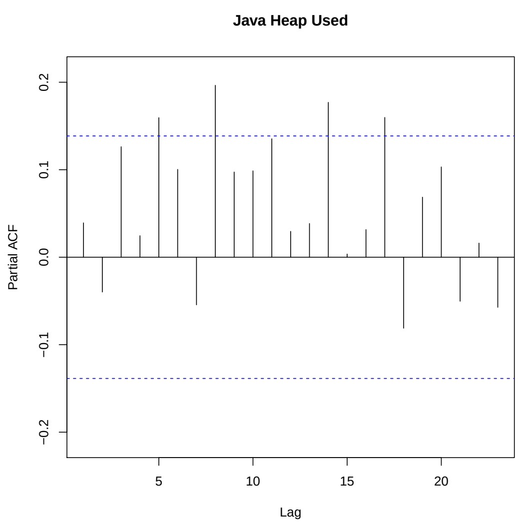
Začneme vykreslením časovej rady. Na obrázku 1 je možné vidieť, že časová rada obsahuje mierny lineárny trend, čo indikuje nestacionaritu. Vykresíme autokorelačnú (zkrátene *ACF*) a parciálknu autokorelačnú funkciu (zkrátene *PACF*).



Obr. 1: Vyťaženosť Java Heap-u v čase.



(a) Autokorelačná funkcia.



(b) Parciálna autokorelačná funkcia.

Z grafu autokorelačnej funkcie 2a je vidieť že hodnoty sú kladné, relatívne blízko nuly a neklesajú. Na grafe *ACF* taktiež nie sú prítomné periodicky posunuté vysoké hodnoty, ktoré by indikovali sezónnosť. Keďže hodnoty *ACF* so zvyšujúcim spozdením neklesajú k nule rozhodli sme sa urobiť *ADF* test na testovanie stacionarity.

```
> adfTest(heap, lags = 0, type='nc');
Title:
 Augmented Dickey-Fuller Test
Test Results:
  PARAMETER:
    Lag Order: 0
  STATISTIC:
    Dickey-Fuller: -1.1436
  P VALUE:
    0.2508
```

Nulovú hypotézu *ADF* testu o prítomnosti jednotkového koreňa, na hladine významnosti 0.05 nezamietame. Z toho vyplýva, že naša časová rada je nestacionárna. Pre dôkladné overenie skusíme *KPSS* test, ktorého nulová hypotéza znie, že časová rada je stacionárna.

```
> kpss.test(heap);
KPSS Test for Level Stationarity
data:  heap
KPSS Level = 2.1652, Truncation lag parameter = 3, p-value = 0.01
```

Na hladine významnosti 0.05 by sme nulovú hypotézu zamietli. Oba testy nám ukázali, že časová rada nie je stacinárna. Následne môžeme pomocou *KPSS* testu testovať či je daná časová rada trend stacinárna.

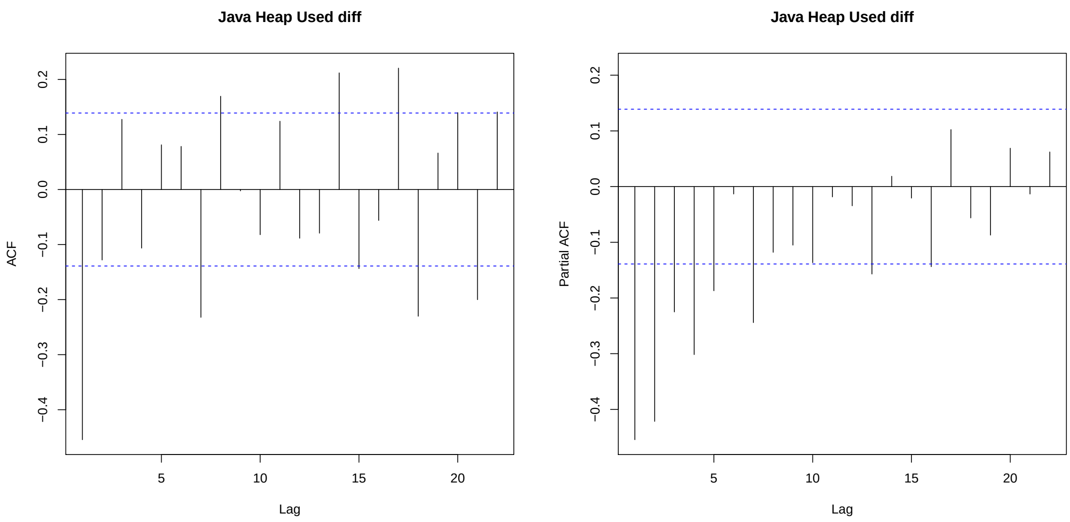
```
> kpss.test(heap, null='Trend');
KPSS Test for Trend Stationarity
data:  heap
KPSS Trend = 0.087693, Truncation lag parameter = 3, p-value = 0.1
```

```
> ndiffs(heap)
[1] 1
```

```
> diff = diff(heap)
```

Z výstupu je jasné, že rada je trend stacinárna takže nulovú hypotézu na hladine 0.05 nezamietame.

Časovú radu stacionarizijeme diferenciovaním a pokračujeme vykreslením *ACF* a *PAC* upravenej časovej rady. Významny bod useknutia sa nám v oboch grafoch 3 nepodarilo nájsť. To indikuje, že výsledny model sa bude skladať z *AR* aj *MA* zložky. Teraz by sme mali hľadať krivku *U*, ktorá od nejakého bodu pripomína krivku v tvare linearnych kombinácii klesajúcich geometrických postupností sinusoid s geometricky klesajúcimi amplitudami. Túto krivka v grafe *ACF* ani *PACF* nie je jednoznačne prítomná **??**. Rozhodli sme sa vybrať model *ARIMA(2,1,1)*. Do modelu sme zahrnuli dve najväčšie hodnoty *PACF* (model *AR*) a jednu *ACF* (model *MA*)



Obr. 3: ACF a PACF diferencovanej časovej rady.

Alternatívny spôsob voľby modelu je pomocou informačných kritérii. Tento spôsob je vhodný pre plne automatizované spracovanie [2] napríklad v ekonometrických softvéroch. K identifikácia modelu *ARMA(p, q)* sa pristupuje ako k minimalizácii funkcie 2.1.

$$(\hat{p}, \hat{q}) = arg \min_{(k,l)} A(k,l) \tag{2.1}$$

A je vhodné kritérium pre výpočet ktorého musíme odhadnúť model *ARMA(k,l)*. Pri minimalizácii postupne inkrementujeme oba parametre k, l. Informačných kritérii existuje viac. V tejto práci sme zvolili Akaikeho informačné kritérium:

$$A(k,l) = AIC(k,l) = ln\hat{\sigma}^2_{k,l} + \frac{2(k+l+1)}{n} \tag{2.2}$$

Z rovnice 2.2 je zrejmé, že kritérium penalizuje veľké rády k a l. $\hat{\sigma}^2_{k,l}$ je smerodajná odchylka reziduí modelu. Poďme si vypísať niekoľko kandidátov *ARIMA* modelov pomocu príkazu auto.arima().

```
> auto.arima(diff, approximation=FALSE, trace=TRUE, ic='aic', allowdrift=FALSE)
 ARIMA(2,1,2)                     : 7556.055
 ARIMA(0,1,0)                     : 7696.058
 ARIMA(1,1,0)                     : 7651.407
 ARIMA(0,1,1)                     : 7557.045
 ARIMA(1,1,2)                     : 7560.654
 ARIMA(3,1,2)                     : 7557.714
 ARIMA(2,1,1)                     : 7553.937
 ARIMA(1,1,1)                     : 7557.937
 ARIMA(3,1,1)                     : 7555.879
 ARIMA(2,1,0)                     : 7613.902

 Best model: ARIMA(2,1,1)
Series: diff
ARIMA(2,1,1)
Coefficients:
          ar1      ar2      ma1
      -0.0985  -0.1774  -0.9441
s.e.   0.0716   0.0716   0.0199
```

Ako je vidieť funkcia zvolila model *ARIMA(2,1,1)* ktorého *AIC* kritérium bolo najnižšie. Odhadnutý model môžeme zapísať v tvare:

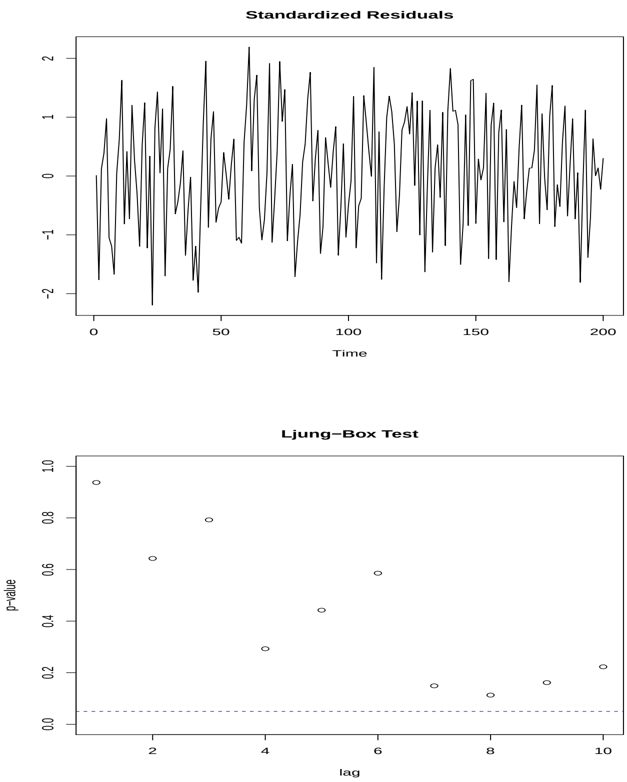
$$Y_t = -0.985Y_{t-1} - 0.1774Y_{t-2} - 0.9441\varepsilon_{t-1} + \varepsilon_t \tag{2.3}$$

Ukázali sme, že je možné konštrulovať správny *ARIMA* model aj analytickým spôsobom. Chcel by som avšak poznanenať, že voľbu modelu je lepšie prenechať overenému štatistickému softvéru.

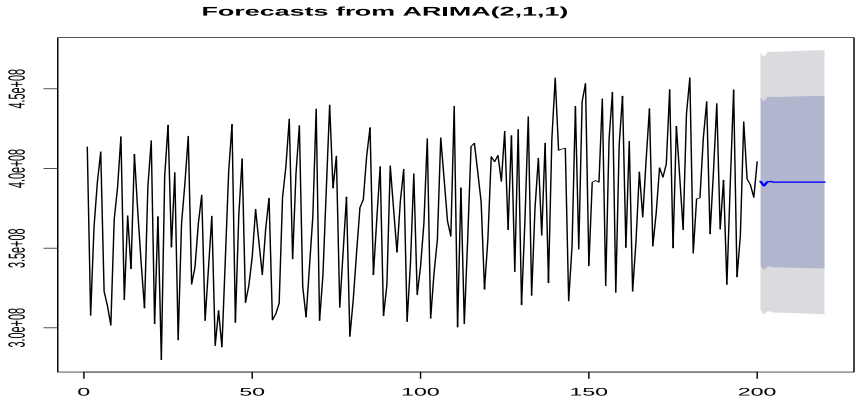
Po úspšnom odhade rádu modelu je by sme chceli zmieniť ako sa počítajú jednotlivé koeficienty. Pre *AR* model platí, že sa dajú vypočítať pomocou *OLS* alebo Yule – Walkerových rovníc [1]. Výpočet kofeicientov *MA* modelu je zložitejší a je možný pomocou rekurzívnej Levison-Durbin metódy. Odhadom presných parametrov modelu sa v tejto práci ďalej nebudene zaoberať.

Na záver sa pozrieme na rezíduá odhadnutého modelu. Ak sme postupovali správne rezíduá by mali pripomínať biely šum a nemali by byť korelované (inakšie by sme ich vedeli modelovať). Toto tvrdenie si overíme Ljung – Box testom, ktorého nulová hypotéza hovorí o tom, že dáta sú nezávislé distribuované. Z grafu 4 môžeme prehlásiť, že nulovú hypotézu na hladine významnosti 0.05 nezamietame.

Na následujúcich grafoch si vykresíme rezíduá modelu, ich autokorelačnú funkciu a p-hodnoty pre rôzdne opozdenia Ljung – Box testu.



Obr. 4: Rezíduá odhadnutého *ARIMA* modelu.



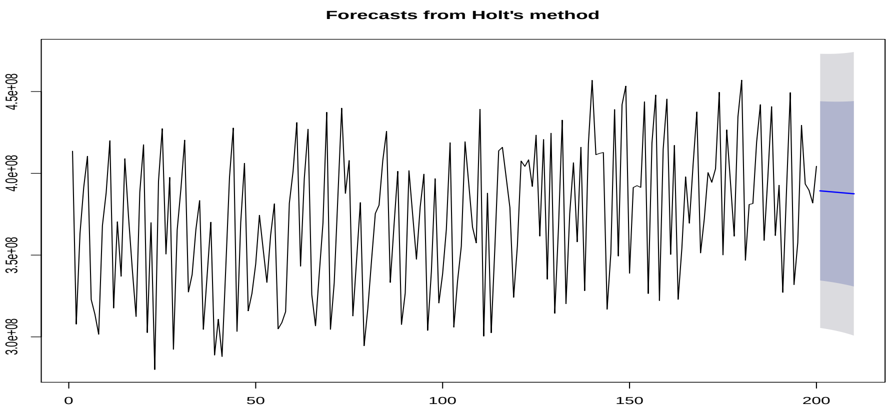
Obr. 5: ARIMA, predikcia na 20 krokov do predu.

3 Exponenciálne vyhladzovanie – Holtova metóda

Zobecnením dvojitého exponenciálneho vyhladzovania je takzvaná Holtova metoda. Táto metóda používa dve vyrovnávacie konštanty: α pre vyrovnanie úrovne l_t a β pre vyrovnanie smernice b_t (lineárneho trendu). Výhoda tejto metódy je možnosť použitia v prúdovom spracovaní, kde nie je možné získať staré hodnoty časovej rady. Rovnice majú tvar 3.1. Parametre α, β patria do intervalu $\alpha, \beta \in (0,1)$. Pre exponenciálne vyhladzovanie platí, čím je hodnota parametra α menšia, tak väčšia váha je daná pozorovaniam z vzdialenej minulosti.

$$\begin{aligned} \hat{y}_{t+h} &= l_t + hb_t \\ l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \end{aligned} \tag{3.1}$$

V jazyku *R* použijeme funkciu *holt*, ktorá počíta predikciu na k krokov do predu. Súčasťou funkcie je aj výpočet parametrov α a β.



Obr. 6: Holt, predikcia na 20 krokov do predu.

4 Adaptívna filtrácia – LMS algoritmus

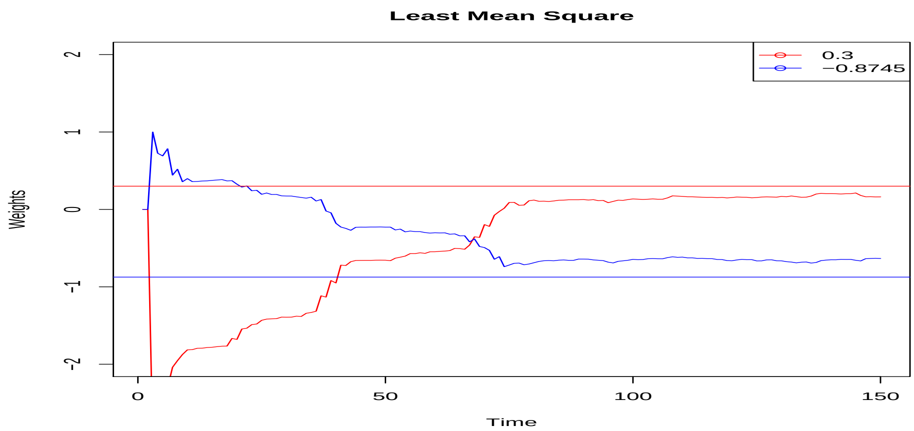
Odhad parametrov autoregresívneho modelu je možné vypočítať pomocou algoritmu *LMS* (Least Mean Square). Výpočet venktoru váh je uvedený v rovnici 4.1. Chyba e je rozdiel súčasnej hodnoty časovej rady s odhadnutov pomocou váh *LMS*. S väčším počtom pozorovaní je výpočet váh presnejší. Algoritmus má nevýhodu, že dopredu musíme odhanúť parameter α. Toto býva väčšinou problém a preto sa volí buď normalizovaná verzia algoritmu alebo algoritmus RLS (Recursive Least Square) pri ktorý tento parameter neobsahuje. V porovnaní RLS algoritmus konverguje rýchlejšie ale je výpočetne náročnejsí.

$$w(t+1) = w(t) + \alpha * e(t) * x(t) \tag{4.1}$$

Funkčnosť algoritmu budeme demonštrovat na gererovaných dátach, pri ktorých vieme aký proces ich generoval. Takto budeme môcť výsledky overiť s výstupom s *LMS* algoritmu.

```
n = 1500;
series = arima.sim(n, model=list(ar=c(0.3, -0.8745)), rand.gen=rnorm);
result = lms(series, alpha, AR);
"Estimated weights"
[1]  0.3397577 -0.8287951
```

Z uvedeného výstupu z R vidíme že časová rada bola generovaná pomocou $y = 0.3 * y_{t-1} - 0.8745 * y_{t-2}$ a na vstup vstup dát z normálneho rozdelenia so strednou hodnotou 0 a smerodatnou odchylkou 1. Z grafu 7 je vidieť priebeh výpočtu váh pomocou *LMS* algoritmu.



Obr. 7: Priebeh výpočtu váh AR procesu pomocu LMS algoritmu.

5 Záver

Na záver by som porovnal kvadratickú sumu chýb (*SSE*) oboch modelov. Pre model *ARIMA* chyba vyšla $3.377059 * 10^17$ pre Holtovu metódu $3.657135 * 10^17$. Vidíme, že rozdiel nieje významne iný. Keď opticky porovnáme grafy 5, 6 vidíme, že exponenciálne vyhladzovanie strmšie klesá.

Voľba vhodnéh modelu niekedy nezávisi len na najmenšej chybe predpovedi. Niekedy je nutné voliť výpočetne nenáročny model, aby bolo napríklad možné spracovávať obrovské množstvo časových rád paralelne.

Výsledky adaptívnej filtrácie ukázali, že je možná vypočítať parametre *AR* modelu pomoocu jednoduchých algoritmov ako je napríklad *LMS* ak máme dostatočne veľký počet pozorovaní. Kombináciou adaptívnej filtrácie a exponenciálneho vyhladzovania by sme mohli dospieť k zaujímavému modelu, ktorý by mohol mať odbré prediktívne vlastnosti a nízku výpočetnu náročnosť.

Poďakovanie

Na záver by som chcel poďakovať Ing. Danielovi Němcovi, Ph.D. za návrh na vypracovanie tejto témi a za veľmi príjemné a užitočné konzultácie. Ďalej by som chcel poďakovať Ester Železňákovej za gramatickú korektúru textu.

Literatúra

[1] Brockwell, P.; Davis, R.: *Time Series: Theory and Methods*. Praha: Springer-Verlag New York, 2009, ISBN 978-0-387-97429-3.

[2] Tomáš, C.: *Finanční ekonometrie*. Ekopress, 2008, ISBN 978-80-86929-43-9.

A Prílohy

- Zdrojové súbory v jazyku R v adresáry `scripts`
- Dáta analyzovanej časovej rady v adresáry `scripts`
- Zdrojový text tejto správy v LaTeX – e