

**UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**TAKENSOVA VETA A PREDPOVEDANIE
EKONOMICKÝCH/FINANČNÝCH ČASOVÝCH
RADOV**

Diplomová práca

BRATISLAVA 2013

Bc. Andrea Čížková

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



TAKENSOVA VETA A PREDPOVEDANIE
EKONOMICKÝCH/FINANČNÝCH ČASOVÝCH
RADOV

Diplomová práca

Študijný program: Ekonomická a finančná matematika

Študijný odbor: 1114 Aplikovaná matematika

Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky

Školiteľ: Doc. Mgr. Marián Grendár, PhD.

BRATISLAVA 2013

Bc. Andrea Čížková



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Andrea Čížková
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Takensova veta a predpovedanie ekonomických/finančných časových radov

Cieľ: ARIMA modelovanie predpokladá lineárny, stacionárny časový rad. Štrukturálny prístup a Kalmanov filter predpokladajú lineárny časový rad. Takensova veta z teórie dynamických systémov je základom pre prístup k predpovedaniu nelineárnych časových radov. Cieľom diplomovej práce je podať prehľad o tomto prístupe a ilustrovať ho na ekonomickom/finančnom časovom rade.

Vedúci: doc. Mgr. Marián Grendár, CSc.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.
Dátum zadania: 25.01.2012

Dátum schválenia: 26.01.2012
prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Abstrakt

Čížková, Andrea: Takensova veta a predpovedanie ekonomických/finančných časových radov [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky.

Vedúci diplomovej práce: Doc. Mgr. Marián Grendár, PhD.

Bratislava 2013

V diplomovej práci je popísaná teória vnorenia, ktorej základom je Takensova veta. Táto teória je ilustrovaná na finančných časových radoch výmenných kurzov. Ďalej je uvedená obchodná stratégia F. Strozziovej [12], ktorá je založená na rekonštrukcii stavového priestoru, a podľa ktorej je možné získať kladný čistý pri obchodovaní s výmennými kurzami. V závere porovnáваме presnosť predikcie metódy najbližších susedov, ktorá súvisí s teóriou vnorenia s ARMA modelom a modelom založenom na štrukturálnom prístupe.

Kľúčové slová: Takensova veta • vnáranie časových radov • metóda najbližších susedov.

Abstract

Čížková, Andrea: Takens' theorem and forecasting of economic/financial time series [Master Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics.

Supervisor: Doc. Mgr. Marián Grendár, PhD.

Bratislava 2013

In the thesis the embedding theory is described, which is based on Takens' theorem. This theory is illustrated in foreign exchange time series. Next, F. Strozzi's trading methodology [12] based on state space reconstruction is presented. It is possible to obtain a positive gain in all those time series using this methodology. Finally, the accuracy in prediction of method of nearest neighbors (related to the embedding theory), ARMA model and model based on structural approach are compared.

Keywords: Takens' theorem • Time series embedding • Nearest neighbor method.

Pod'akovanie

Ďakujem svojmu vedúcemu diplomovej práce za odborné vedenie, pripomienky a cenné rady pri písaní tejto diplomovej práce. Ďakujem taktiež svojmu priateľovi a rodine za trpezlivosť a podporu.

Čestné prehlásenie

Čestne prehlasujem, že som diplomovú prácu vypracovala samostatne s použitím uvedenej literatúry a s odbornou pomocou školiteľa.

.....

Bc. Andrea Čížková

Obsah

Úvod	1
1 Teória vnorenia	2
1.1 Rekonštrukcia stavového priestoru	2
1.2 Dimenzia vnorenia	4
1.3 Časový posun	5
1.4 Takensova veta	7
1.5 Stacionarita	8
2 Vnárание finančných časových radov	10
2.1 Stroziovej stratégia	10
2.2 Jednoduchý príklad	12
2.3 Analýza a výsledky	13
2.3.1 Dáta	13
2.3.2 Vnárание výmenných kurzov	14
3 Vnárание ekonomických časových radov	18
3.1 Metóda najbližších susedov	18
3.2 ARMA model	21
3.3 Štrukturálny prístup	23
3.4 Zhrnutie	25
Záver	26
Literatúra	28
Príloha	31

Úvod

Cieľom práce je podať prehľad o predpovedaní ekonomických a finančných časových radoch pomocou Takensovej vety teórie dynamických systémov. ARIMA modelovanie lineárny a stacionárny časový rad. Štrukturálny prístup predpokladá lineárny časový rad. Takensova veta je základom pre prístup k predpovedaniu nelineárnych časových radov.

Diplomová práca je rozdelená do troch kapitol. V prvej kapitole uvádzame zhrnutie teórie vnorenia, ktorej základom je práve Takensova veta. Uvedieme metódy na určenie odhadu parametrov vnorenia.

V druhej kapitole tieto metódy ilustrujeme na finančných časových radoch, konkrétne na výmenných kurzoch medzi americkým dolárom a desiatimi ďalšími menami v roku 2011. Ďalej tu uvedieme obchodnú stratégiu F. Strozziovej, v ktorej sa využíva teória vnorenia a pomocou ktorej je možné dosiahnuť kladný zisk, za predpokladu nulových transakčných nákladov pri obchodovaní s týmito výmennými kurzami.

V tretej kapitole ilustrujeme teóriu vnorenia na ekonomickom časom rade. Budeme ho predpovedať pomocou metódy najbližších susedov, ktorá súvisí s teóriou vnorenia. Predpoveď pomocou metódy najbližších susedov porovnáme s ARIMA modelom a štrukturálnym prístupom.

Teória vnorenia

1.1 Rekonštrukcia stavového priestoru

Metódy nelineárnych časových radov v tejto práci sú založené na teórii dynamických systémov. Dynamický systém je matematická formalizácia pre fixný “predpis”, ktorý popisuje časovú závislosť polohy buď v jeho okolitom prostredí. Rôzne spôsoby merania času a špeciálne vlastnosti okolitého prostredia môžu poskytnúť predstavu o obširnej triede objektov, ktoré sú opísané týmto “predpisom”. Čas môže byť zaznamenaný v celých číslach, reálnymi alebo komplexnými číslami, okolitý priestor môže byť napr. množina, bez potreby definovania hladkých časovo-priestorových štruktúr.

Formálna definícia dynamického systému založená na ergodickej teórii: dynamický systém je štvorica (X, S, P, T) , kde X je neprázdna množina, S je σ -algebra podmnožín množiny X , P je pravdepodobnostná miera na S , T je mieru zachovávajúce zobrazenie a (X, S, P) je pravdepodobnostný priestor.

Inak a zjednodušene povedané, dynamický systém sa skladá zo stavového (fázového) priestoru, ktorého súradnice popisujú stav systému v danom čase a z dynamických podmienok, ktoré popisujú zmenu tohto systému v čase.

Uvažujme ďalej deterministický dynamický systém. Pre jednoduchosť sa obmedzíme na prípad, keď fázový priestor je \mathbb{R}^m v konečnej dimenzii (m). Stav systému je daný vektorom $\mathbf{x} \in \mathbb{R}^m$.

V reálnom svete čelíme nasledovnému problému: výstup, ktorý vidíme pri nejakom experimente, nie je objekt fázového priestoru, ale iba nejaký časový rad, zväčša len séria skalárnych meraní. Keďže dynamiku skúmaného systému možno študovať na základe dynamiky vektorov v stavovom priestore, treba tieto pozorovania pre-

viest' na stavové vektory, čo znamená zrekonštruovať fázový priestor. Tento problém je možné technicky vyriešiť použitím metódy oneskorení (*method of delays*).

Väčšinou je časový rad, teda postupnosť skalárnych meraní nejakej veličiny, ktorá závisí od aktuálneho stavu systému, zaznamenaný v násobkoch pevnej časovej dĺžky

$$s_n = s(\mathbf{x}(n\Delta t)) + \eta_n \quad (1.1)$$

Na systém sa pozeráme cez určitú meraciu funkciu s a pozorovania obsahujú náhodné fluktuácie η_n , chyby merania, merací šum (*measurement noise*). Ďalej pre jednoduchosť zanedbáme efekt šumu.

Rekonštrukcia fázového priestoru pomocou metódy oneskorení (*delay reconstruction*) v m dimenziách je vytvorená vektormi \mathbf{s}_n

$$\mathbf{s}_n = (s_{n-(m-1)\tau}, s_{n-(m-2)\tau}, \dots, s_{n-\tau}, s_n) \quad (1.2)$$

Časový rozdiel medzi susednými zložkami vektorov oneskorení (*delay vectors*) τ sa označuje ako lag alebo čas oneskorenia (*delay time*). Všimnime si, že pre $\tau > 1$, sa zväčšuje iba časové okno pokryté každým vektorom, zatiaľ čo počet vektorov skonštruovaných zo skalárneho časového radu zostáva stále približne rovnaký. Toto je spôsobené tým, že vytvárame vektor pre každé skalárne pozorovanie s_n , $n > (m-1)\tau$. Množstvo matematických viet o vnorení (*embedding theorems*) sa zaoberá otázkou, za akých predpokladov je objekt vytvorený vektormi \mathbf{s}_n ekvivalentný¹ s originálnou trajektóriou \mathbf{x}_n . Ukazuje sa, že za celkom všeobecných predpokladov, atraktor vytvorený vektormi \mathbf{s}_n je ekvivalentný, tzv. generický s atraktorom v neznámom priestore, v ktorom existuje originálny systém, ak dimenzia m priestoru s oneskorenými súradnicami (*delay coordinate space*) je dostatočne veľká.

Pre ilustráciu uvidíme konkrétny príklad. Uvažujme dátový súbor, ktorý pozostáva z meraní $x_i(t)$ jednej premennej x_i , meraných každých Δt sekúnd, vid' tab. 1.1. Na to, aby sme vnorili tento dátový súbor, musíme skonštruovať m -dimenzionálne stavové vektory $\mathbf{s}(t)$ nasledovným spôsobom:

$$\mathbf{s}(t) = (x_i(t - (m-1)\tau), x_i(t - (m-2)\tau), \dots, x_i(t - \tau), x_i(t))$$

Napríklad, ak časový rad z tabuľky 1 chceme vnoriť do dvoch dimenzií s časovým posunom $\tau = 0,005$, prvé vektory vnorenia budú nasledovné:

(1,6352 1,6260), (1,6337 1,6263), (1,6322 1,6214), (1,6306 1,6214),
(1,6276 1,6183).

¹V zmysle, že sa dajú navzájom zobrazit' pomocou invertovateľnej hladkej funkcie.

$x_i(t)$	t	$x_i(t)$	t
1,6352	0,000	1,6214	0,008
1,6337	0,001	1,6183	0,009
1,6332	0,002	1,6183	0,010
1,6306	0,003	1,6168	0,011
1,6267	0,004	1,6137	0,012
1,6260	0,005	1,6107	0,013
1,6230	0,006	1,6076	0,014
1,6214	0,007	1,6045	0,015

Tabuľka 1.1: Príklad dátového súboru: merania jednej premennej x_i meraných každých $\Delta t = 0,001$ sekúnd.

Pre $m = 5$ a $\tau = 0,003$ prvých niekoľko vektorov má tvar:
 (1,6352 1,6306 1,6230 1,6183 1,6137), (1,6337 1,6276 1,6214 1,6183 1,6107),
 (1,6322 1,6260 1,6214 1,6168 1,6076), (1,6306 1,6230 1,6183 1,6137 1,6045).

Vedomosť správnej dimenzie vnorenia m a lagu τ uľahčuje analýzu dát (dôvody vid' nižšie), preto sa vyvinuli rôzne metódy na ich určenie.

1.2 Dimenzia vnorenia

Poznanie presnej hodnoty m je žiadúce, pretože chceme využiť determinizmus s minimálnym výpočtovým úsilím. Ak nejaké m -dimenzionálne vnorenie vedie ku vernej reprezentácii stavového priestoru, potom aj každá m' -dimenzionálna rekonštrukcia fázového priestoru pre $m' > m$ je tiež dobrá. Voľba praveľkej hodnoty m pre chaotické dáta spôsobí redundanciu, a teda zníži výkonnosť mnohých algoritmov. Kvôli nestabilite chaotického pohybu prvá a posledná zložka vektora oneskorenia majú tým menšiu závislosť, čím väčší je ich časový rozdiel. Takže voľba veľkej hodnoty m veľmi nepomôže a riskujeme "popletenie" algoritmov.

Ak predpokladáme, že dynamika vo fázovom priestore je reprezentovaná hladkým vektorovým poľom, potom susedné stavy by mali podliehať takmer rovnakému časovému vývoju. Takže dve trajektórie vychádzajúce z dvoch susedných stavov by mali byť stále blízky susedmi aj po krátkom čase úseku. Táto idea je základom tzv. metódy falošných najbližších susedov, ktorá slúži na nájdenie vhodnej dimenzie vnorenia.

Metódu falošných najbližších susedov (*false nearest neighbours method*) prvýkrát publikovali Kennel, Brown a Abarbanel [1]. V tejto práci ju uvedieme s menšími

úpravami, ktorými sa chceme vyhnúť určitým nesprávnym výsledkom pre šum [2].

Hlavnou myšlienkou je hľadanie takých bodov v dátach, ktoré sú susedmi v priestore vnorení (*embedding space*), ale ktoré by nemali byť susedmi, pretože ich budúci časový vývoj je príliš odlišný. Nech pre určité dáta je správna dimenzia vnorenia m_0 . Pozrime sa na tieto dáta v nižšej dimenzii $m < m_0$. Prechod z m_0 do m je projekcia, ktorá eliminuje určité osi zo súradnicového systému. Body, ktorých súradnice boli odstránené a boli veľmi odlišné, sa môžu stať “falošnými susedmi” v m -dimenziálnom priestore.

Pre každý bod časového radu vezmeme jeho najbližšieho suseda v dimenzii m a vypočítame pomer vzdialeností medzi týmito dvoma bodmi v dimenzii $m + 1$ a dimenzii m . Ak je tento pomer väčší ako hraničná hodnota r , tak tento sused bol falošný. Táto prahová hodnota musí byť dostatočne veľká, aby brala do úvahy aj exponenciálnu divergenciu spôsobenú deterministickým chaosom.

Označme štandardnú odchýlku dát ako σ a použime maximovú normu. Štatistiku, ktorú musíme počítať je nasledovná

$$X_{fnn}(r) = \frac{\sum_{n=1}^{N-m-1} \Theta \left(\frac{|\mathbf{s}_n^{(m+1)} - \mathbf{s}_{k(n)}^{(m+1)}|}{|\mathbf{s}_n^{(m)} - \mathbf{s}_{k(n)}^{(m)}|} - r \right) \Theta \left(\frac{\sigma}{r} - |\mathbf{s}_n^{(m)} - \mathbf{s}_{k(n)}^{(m)}| \right)}{\sum_{n=1}^{N-m-1} \Theta \left(\frac{\sigma}{r} - |\mathbf{s}_n^{(m)} - \mathbf{s}_{k(n)}^{(m)}| \right)} \quad (1.3)$$

kde $\mathbf{s}_{k(n)}^{(m)}$ je najbližší sused \mathbf{s}_n v dimenzii m , t.j. $k(n)$ je index prvku k časového radu, ktorý je odlišný od n a pre ktorý platí $|\mathbf{s}_n - \mathbf{s}_k| = \min$ a Θ je Heavisidova funkcia. Prvá Heavisidova funkcia v čitateľovi sa rovná jednej, ak najbližší sused je falošný, t.j. ak zväčšíme dimenziu vnorenia o jednotku, vzdialenosť vzrastie činiteľom väčším ako r . Druhá Heavisidova funkcia ruší všetky tie páry, ktorých počiatočná vzdialenosť bola väčšia než σ/r . Páry, ktorých počiatočná vzdialenosť je väčšia ako σ/r , podľa definície nemôžu byť falošní susedia, pretože tu neexistuje dostatok priestoru na odchylenie sa viac ako σ .

Falošní najbližší susedia sa môžu vyskytovať aj v správnej dimenzii vnorenia. Kvôli meraciemu šumu môžeme mať pri väčšom množstve dát vyšší počet falošných susedov. Pri väčšom množstve dát je najbližší sused typicky bližšie.

1.3 Časový posun

Získať dobrý odhad lagu, t.j. časového posunu τ je zložitejšie. Vety o vnorení nehovoria nič o lagu τ , pretože predpokladajú dáta s nekonečnou presnosťou. Teda z

matematického hľadiska τ možno vziať ľubovoľné. Ďalším predpokladom je aj nekonečný počet dát. Vnorenia s rovnakým m , ale odlišným τ sú pre dáta bez šumu v matematickom zmysle ekvivalentné, avšak v realite dobrá voľba τ uľahčuje analýzu dát. Ak sa časové oneskorenie τ vezme príliš malé, zložky vektorov vnorenia budú silne korelované. Všetky vektory \mathbf{s}_n budú zoskupené okolo diagonály v priestore vnorenia \mathbb{R}^m . Ak je τ priveľké, zložky vektorov vnorenia sú takmer nezávislé a tieto vektory vyplnia veľký priestor v \mathbb{R}^m . V tomto prípade môže byť zrekonštruovaný atraktor veľmi komplikovaný, hoci ten originálny je jednoduchý.

Aké kritériá odporučiť pri výbere τ ? Po prvé, môžeme použiť geometriu. Atraktor by mal byť “rozvinutý” (*unfolded*), t.j. rozšírenie atraktora vo všetkých dimenziách by malo byť zhruba rovnaké. Štatistiky ako *fill faktor* [3] alebo “presunutie z diagonály” [4] sa používajú na kvantifikovanie tohto argumentu. Rigoróznejší prístup je popísaný v [5]. Ale jedna z najprirodzenejších štatistík je autokorelačná funkcia signálu. Pri tomto prístupe sa vezme to τ , pri ktorom autokorelačná funkcia prvýkrát dosiahne hodnotu nula, t.j. jej prvý prechod cez nulu.

Rozumná námietka proti tomuto výberu lagu je, že je založený na lineárnej štatistike a neberie do úvahy nelineárne korelácie. Fraser a Swinney prišli s konceptom použitia vzájomnej informácie MI (*mutual information*) [6], konkrétne treba hľadať jej prvé minimum. MI vyjadruje informáciu, ktorú máme o hodnote $s(t + \tau)$, ak poznáme $s(t)$. Je to druh nelineárnej korelačnej funkcie, pomocou ktorej sa dá určiť, kedy sú hodnoty $s(t)$ a $s(t + \tau)$ dostatočne nezávislé, aby sa dali použiť ako súradnice vektorov vnorenia, na druhej strane nie nezávislé do takej miery, že by nemali spolu žiadnu súvislosť. Postup, ktorým sa dá MI odhadnúť, je nasledovný: na intervale pokrytom dátami vytvoríme histogram pre pravdepodobnostné rozdelenie dát s dĺžkou okna ε . Označme p_i ako pravdepodobnosť, že signál má hodnotu z i -teho stĺpca histogramu a p_{ij} ako pravdepodobnosť, že $s(t)$ je v stĺpci i a $s(t + \tau)$ je v stĺpci j . Potom MI má pre časový posun τ tvar:

$$I_\varepsilon(\tau) = \sum_{i,j} p_{ij}(\tau) \ln p_{ij}(\tau) - 2 \sum_i p_i \ln p_i \quad (1.4)$$

Obvykle sa používa fixná šírka stĺpcov histogramu.

Šírka stĺpcov by sa mala vybrať taká, aby bolo možné vidieť stabilnú/stálu závislosť na τ . Pre malé τ bude $I_\varepsilon(\tau)$ veľké. Potom bude viac alebo menej rýchlo klesať. Pre limitne veľké τ sú $s(t)$ a $s(t + \tau)$ nezávislé a pravdepodobnosť p_{ij} možno rozložiť na súčin $p_i p_j$ a MI sa v tomto prípade rovná nule. Prvé minimum funkcie $I_\varepsilon(\tau)$ značí taký časový lag, v ktorom hodnota $s(t + \tau)$ pridáva najlepšiu informáciu k

našej vedomosti, ktorú máme z $s(t)$.

1.4 Takensova veta

Skalárne meranie je projekcia nepozorovateľnej vnútornej premennej systému na reálnu os. Táto projekcia môže byť nelineárna a môže skombinovať rôzne vnútorné veličiny, takže výsledný výstup môže byť skreslený. Je zrejmé, že aj keby sme presne poznali proces merania, rekonštrukcia stavového priestoru originálneho systému z dát môže byť neuskutočniteľná. Našťastie, rekonštrukcia originálneho fázového priestoru pre analýzu dát nie je v skutočnosti potrebná a niekedy ani žiadúca, napr. keď je dimenzia atraktora oveľa menšia než dimenzia tohto priestoru. Je vhodné skonštruovať taký nový priestor, v ktorom je atraktor ekvivalentný s tým originálnym.

Chceme predovšetkým využiť determinizmus dát. Teda časový vývoj trajektórie v rekonštruovanom priestore by mal závisieť iba od jej aktuálnej pozície v tomto priestore. Jedinečnosť dynamiky v rekonštruovanom priestore nie je jediná vlastnosť, ktorú potrebujeme. Dimenzie, Lyapunovove exponenty a entropie sú invariantné iba pri hladkých nesingulárnych transformáciách. Preto aby sme garantovali, že veličiny vypočítané pre rekonštruovaný atraktor sú identické s tými v originálnom stavovom priestore, musíme požadovať, aby sa štruktúra tangentného (dotykového) priestoru zachovala pri rekonštrukcii. Teda vnorenie (*embedding*) kompaktnej hladkej variety (*manifold*) A do \mathbb{R}^m je definované ako mapa F , ktorá je imerziou (*immersion*) na A , t.j. C^1 mapa, ktorej Jakobian $DF(x)$ má všade plnú hodnotu. Kľúčovým bodom je ukázať, za akých podmienok zrekonštruované vektory vnorenia formujú vnorenie.

V roku 1936 Whitney [7] dokázal, že každá D -rozmerná hladká varieta sa môže vnoriť do priestoru \mathbb{R}^{2D+1} , a že množina máp formujúcich vnorenie je hustá a otvorená množina v priestore C^1 máp.

Whitneyho veta však nie je v praxi použiteľná. Predovšetkým bola dokázaná iba pre celé čísla D . Po druhé, nehovorí o tom, ako je hodnoverné, že daná mapa skutočne formuje vnorenie. Fakt, že “dobré” mapy sú husté v priestore C^1 máp iba garantuje, že existuje vnorenie na ľubovoľne malom okolí danej mapy [8].

Situácia pre rekonštrukciu atraktora A dynamického systému je veľmi špecifická. Atraktor je podmnožina stavového priestoru systému. Toto zaručuje, že zobrazenie F zo stavového vektora v určitom čase na jeho pozíciu o jeden časový posun neskôr

je jedinečné. Toto je dôležitá vlastnosť, ktorú má časovo-posunuté vnorenie zachovať. Uvažujme varietu v konfiguračnom priestore a náhodnú prechádzku na nej. Táto varieta sa nedá zrekonštruovať pomocou časovo-posunutého vnorenia. Body v tejto variete nepopisujú stav systému jednoznačne, a teda medzi po sebe idúcimi pozorovaniami nie je deterministická súvislosť.

Keďže dynamika sa môže chápať ako jedinečné zobrazenie z A do A , časovo-posunuté vnorenie je tiež časovo nezávislá mapa z A do \mathbb{R}^m . Označme, tak ako pred tým, \mathbf{x}_n ako stavový vektor v čase n , s ako meraciu funkciu a F ako mapu reprezentujúcu dynamiku, $\mathbf{x}_{n+1} = F(\mathbf{x}_n)$. Potom vektory vnorenia majú tvar

$$(s(\mathbf{x}_n), s(F(\mathbf{x}_n)), s(F \circ F(\mathbf{x}_n)), \dots)$$

Poznanie hodnôt $s(\mathbf{x}_{n_i})$ v po sebe idúcich časoch n_i je ekvivalentné s poznaním množiny rôznych súradníc v jednom momente, ak mapa F spája rôzne stupne voľnosti.

Takens dokázal [9], je to všeobecná vlastnosť, že mapa vnorenia dimenzie $m = 2D + 1$ je vnorenie kompaktnej variety dimenzie D , ak meracia funkcia $s : A \rightarrow \mathbb{R}$ je C^2 a ak buď dynamika alebo meracia funkcia je generická². D je celočíselná dimenzia hladkej variety, t.j. priestoru, ktorý obsahuje atraktor. Teda D môže byť oveľa väčšie ako dimenzia atraktora.

Inak povedané, pri vhodnej dimenzii m a lagu τ zrekonštruovaný a originálny stavový priestor sú topologicky identické. Formálnejšie, ak $m \geq 2D + 1$, kde D je skutočná dimenzia originálneho systému, tak zrekonštruované trajektórie a trajektórie v originálnom priestore sú difeomorfné.

Čitateľom, ktorí majú hlbší záujem o matematické pozadie tejto problematiky, odporúčame prácu *Embedology* [10].

1.5 Stacionarita

Pojem stacionarita zjednodušene znamená, že všetky relevantné parametre systému musia byť konštantné počas celej periódy merania (a tiež by mali ostať rovnaké, keď sa experiment zopakuje). Pre daný pravdepodobnostný proces, ktorý je charakterizovaný pravdepodobnostným rozdelením jeho premenných, stacionarita značí, že toto rozdelenie nezávisí od času. Ak je dynamika systému riadená nejakým deterministickým pravidlom, tak toto pravidlo sa v čase nesmie meniť. Formálnejšie,

²Tu generická znamená otvorená a hustá.

signál nazývame stacionárny, ak jeho združené pravdepodobnostné rozdelenie sa nemení pri posune v čase alebo priestore. Predpokladajme, že máme n vzoriek časového radu $s(t)$ meraných v časoch t_1 až t_n . Stacionarita znamená, že združená hustota pravdepodobnosti týchto n vzoriek je identická so združenou hustotou iných n vzoriek získaných v časoch t_{1+k} až t_{n+k} . A toto musí platiť pre všetky n a k .

Ďalej sa ponúka otázka, ako zistiť či dáta sú alebo nie sú stacionárne. Spomenieme tu jeden relatívne jednoduchý test stacionarity nazývaný *space-time separation plot* [11]. Hlavná myšlienka spočíva v tom, že v prítomnosti časovej korelácie pravdepodobnosť, že daný pár stavových bodov má vzdialenosť menšiu ako ε , nezávisí iba od pozície týchto stavov, ale aj od času, ktorý medzi nimi uplynul. Táto závislosť sa dá zistiť nakreslením počtu susedných bodov, ako funkcia dvoch premenných, časovej vzdialenosti Δt a priestorovej vzdialenosti ε . Technicky povedané, pre každé Δt sa vytvorí kumulovaný histogram priestorových vzdialeností ε . V prípade stacionarity nájdeme v obrázku saturáciu.

Vnárание finančných časových radov

2.1 Strozziovej stratégia

V tejto časti sme použili rekonštrukciu stavového priestoru na odhad objemu stavového priestoru a jeho variácie. Využili sme obchodnú stratégiu navrhnutou F. Strozzi [12], pomocou ktorej sme sa snažili určiť a kvantifikovať predvídateľnosť vo finančných časových radoch.

Na analýzu sme použili teóriu vnorenia. Táto teória zaručuje, že je možné zrekonštruovať stavový priestor dynamického systému použitím časovo posunutých vektorov vnorenia, ktoré sú vytvorené z pôvodných meraní. Toto ale znamená, že musíme určiť parametre vnorenia, t.j. časový posun a dimenziu vnorenia.

Keďže hlavným záujmom tejto stratégie bolo predpovedanie časového radu jeden krok dopredu, za optimálne parametre vnorenia sme považovali tie (podľa Strozzi), pri ktorých sa dosahoval maximálny zisk. Avšak treba mať na pamäti, že tieto parametre nie sú optimálne pre iné časové rady v inom roku alebo pri optimalizácii inej funkcie.

Ako bolo už bolo spomínané, rekonštrukcia stavového priestoru zachováva určité informácie o originálnom dynamickom systéme, z ktorého pochádzajú časové rady, ktoré skúmame. Všetky tieto informácie možno využiť pri štúdiu asymptotického správania daného systému. Pod pojmom asymptotické myslíme také vlastnosti, ktoré sa zachovávajú pri dostatočne dlhom čase, t.j. $t \rightarrow \infty$. Na určenie aktuálneho stavu systému sme použili lokálnu mieru, konkrétne divergenciu dynamického systému. Divergencia toku, ktorá je lokálne ekvivalentná stope Jakobianu, meria rýchlosť/mieru zmeny infinitezimálneho objemu stavového priestoru $V(t)$ na trajektórii $x(t)$.

Liouvilleova veta hovorí:

$$V(t) = V(0) \exp \left[\int_0^t \operatorname{div} \{ \mathbf{F}[\mathbf{x}(\tau)] \} d\tau \right] \quad (2.1)$$

kde

$$\operatorname{div} \mathbf{F}[\mathbf{x}(t)] = \frac{\partial F_1[\mathbf{x}(t)]}{\partial x_1} + \frac{\partial F_2[\mathbf{x}(t)]}{\partial x_2} + \dots + \frac{\partial F_d[\mathbf{x}(t)]}{\partial x_d} \quad (2.2)$$

Z rovnice (2.1) je možné napísať

$$V(t+h) = V(t) \exp \left[\int_t^{t+h} \operatorname{div}[\mathbf{J}(x)] d\tau \right], \quad (2.3)$$

rozvinutím exponenciálnej funkcie do Taylorovho rad dostaneme

$$V(t+h) = V(t) \exp \left[1 + \int_t^{t+h} \operatorname{div}[\mathbf{J}(x)] d\tau \right]. \quad (2.4)$$

Predchádzajúci integrál je možné vyjadriť ako

$$\int_t^{t+h} \operatorname{div}[\mathbf{J}(x)] d\tau = \frac{(\operatorname{div}[\mathbf{J}_{t+h}] + \operatorname{div}[\mathbf{J}_t])h}{2} \quad (2.5)$$

Vložením rovnice (2.5) do rovnice (2.4) a preusporiadaním členov dostaneme

$$\frac{\operatorname{div}[\mathbf{J}_{t+h}] + \operatorname{div}[\mathbf{J}_t]}{2} = \frac{1}{h} \frac{V(t+h) - V(t)}{V(t)} \quad (2.6)$$

Teda pre $h \rightarrow \infty$

$$\operatorname{div}[\mathbf{J}(x)] = \frac{\dot{V}(t)}{V(t)}. \quad (2.7)$$

Okrem toho sa divergencia zachováva pri rekonštrukcii stavového priestoru, a teda reflektuje lokálne vlastnosti originálneho dynamického systému.

Objem stavového priestoru v čase t sa dá vypočítať ako nasledovný determinant medzi blízkymi bodmi v stavovom priestore:

$$V(t) = \det \begin{bmatrix} s(t) - s(t - \Delta t) & 0 & \dots & 0 \\ 0 & s(t - \Delta t) - s(t - 2\Delta t) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & s(t - (d_E - 1)\Delta t) - s(t - d_E\Delta t) \end{bmatrix} \quad (2.8)$$

Investori vyvinuli mnoho rôznych indikátorov, ktorými sa snažia merať rýchlosť alebo zrýchlenie pohybov cien, a ktoré používajú pri určení svojich obchodných

stratégií. Niektoré populárne sú napr. miera zmeny (*rate of change*) a index relatívnej sily (*relative-strength index*). Všetky tieto indikátory sa snažia predpovedať budúce správanie finančných časových radov.

Ďalej budeme predpokladať, že naše aktíva môžeme meniť za nulové náklady. Stroziovej stratégia predikcie je daná nasledujúcim jednoduchým pravidlom: ak zmena objemu stavového priestoru klesne, t.j. $\Delta V(t) < \Delta V(t - 1)$, všetky naše aktíva v čase $t + 1$ zmeníme na $menu_2$. V opačnom prípade zmeníme všetky aktíva na $menu_1$ [12]. Zmyslom tejto stratégie je zistiť, či má objem stavového priestoru kladné zrýchlenie. Toto zrýchlenie budeme považovať za mieru sily výmenného kurzu. Čistý "zisk", výnos, pri tejto obchodnej stratégii sa vypočíta pomocou gain-loss funkcie g [13] ako

$$g = \frac{y(t + 1) - y(t)}{y(t)} \quad (2.9)$$

Táto funkcia vyjadruje relatívny výnos za jeden časový krok. Celkový výnos pre celý časový rad sa vypočíta ako

$$G = \sum_{i=m\Delta t}^n g_i \quad (2.10)$$

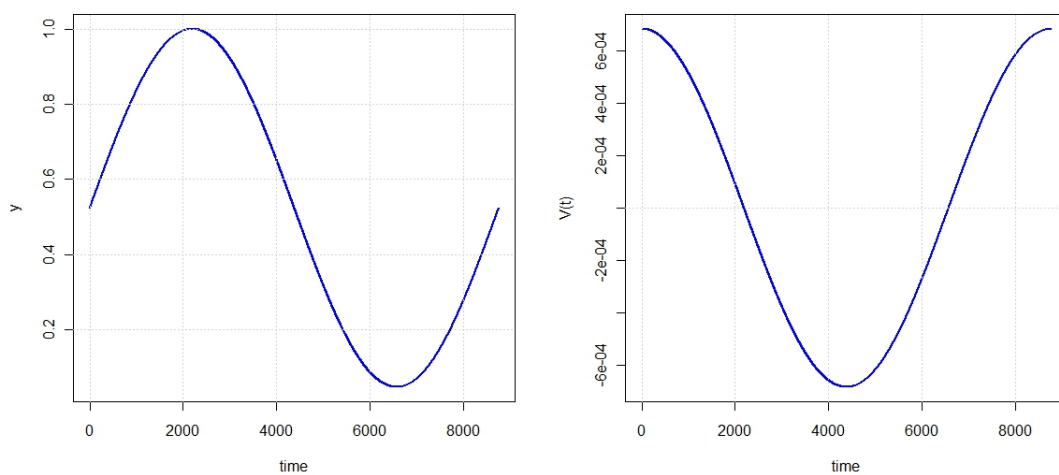
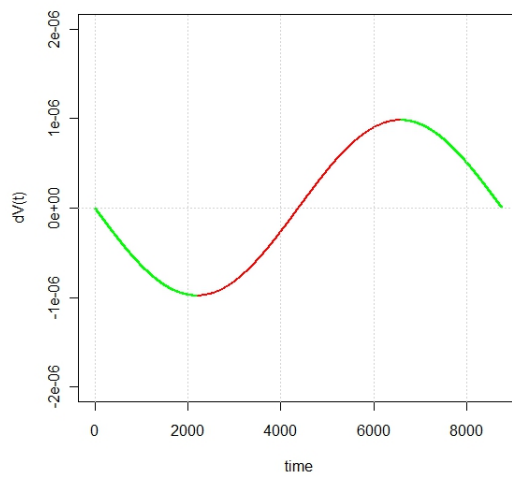
To znamená súčet všetkých možných výnosov a strát za celé časové obdobie. Takže ak $\Delta V(t) < \Delta V(t - 1)$ po prvýkrát, zmeníme naše aktíva do $menu_2$ v čase $t + 1$, ak nie, žiadna zmena sa neudeje. Ako uvidíme neskôr, táto stratégia vytvára značný počet transakcií, keďže ΔV osciluje okolo nuly. V prípade transakčných nákladov by táto stratégia zlyhala.

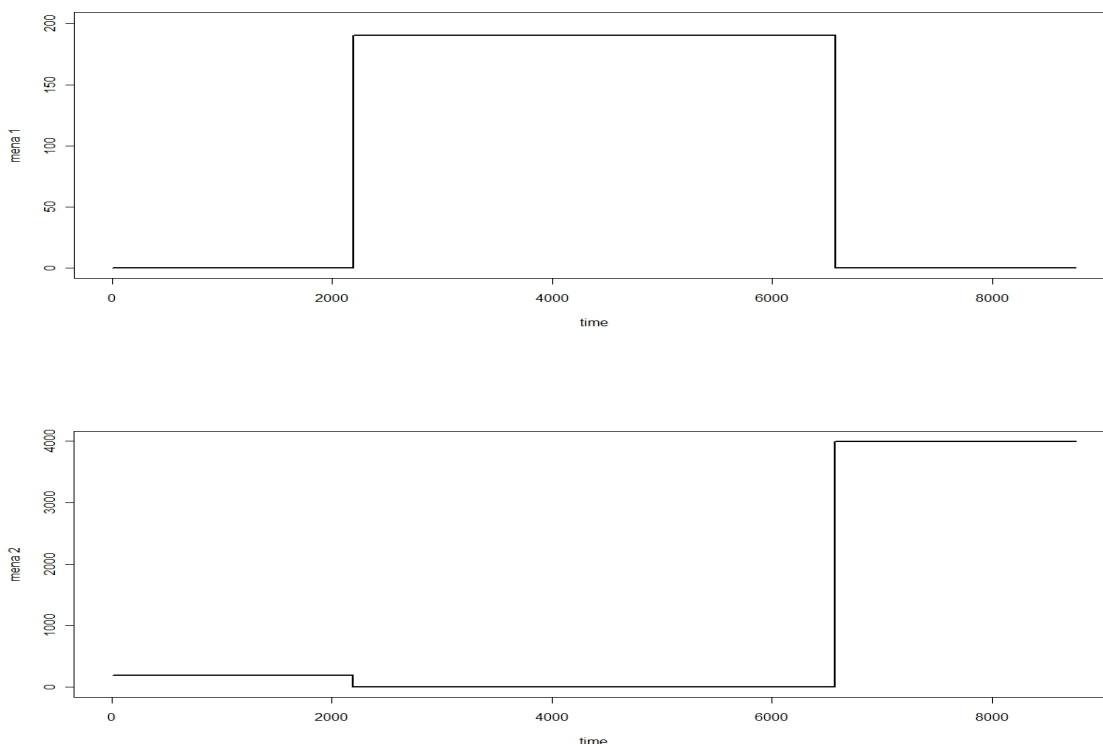
2.2 Jednoduchý príklad

Uvedenú stratégiu bližšie vysvetlíme na nasledovnom jednoduchom príklade [12]. Nech sa výmenný kurz riadi normalizovanou sínusovou funkciou

$$y = \frac{\sin(x) + 1.1}{2.1} \quad (2.11)$$

Obr. (2.1a) zobrazuje túto funkciu, obr. (2.1b) a (2.1c) objem stavového priestoru a jeho zmenu. Podľa vyššie uvedenej stratégie zmeníme naše aktíva v čase $t + 1$ na $menu_2$, keď zmena objemu stavového priestoru klesne a v prípade nárastu zmeny objemu na $menu_1$. Toto je znázornené na obr. (2.1c) červenou a zelenou farbou. Začneme s počiatočným objemom aktív 100 jednotiek v $menu_1$. Na obr. (2.2) možno vidieť vývoj množstva našich aktív v $menu_1$ a v $menu_2$ v čase. Vidíme, že počet transakcií je obmedzený, čo je spôsobené hladkým charakterom funkcie y .

(a) Normalizovaná sínusová funkcia y (b) Objem stavového priestoru $V(t)$ (c) Zmena objemu stav. priest. $\Delta V(t)$ **Obr. 2.1:** Príklad normalizovanej sínusovej funkcie, parametre: $\Delta t = 2, m = 1$



Obr. 2.2: Príklad normalizovanej sínusovej funkcie, parametre: $\Delta t = 2$, $m = 1$

2.3 Analýza a výsledky

2.3.1 Dáta

Dátový súbor, na ktorom použijeme vyššie popísané metódy, pochádza zo stránky <http://www.dukascopy.com> [14] a obsahuje bid a ask ceny US dolára vzhľadom na desať iných mien: austrálsky dolár (AUD), euro (EUR), britská libra (GBP), novozélandský dolár (NZD), kanadský dolár (CAD), švajčiarsky frank (CHF), japonský jen (JPY), nórska koruna (NOK), švédská koruna (SEK) a singapurský dolár (SGD), viď tabuľka 2.1. Rozsah údajov je od 1.1.2011 do 31.12.2011 zaznamenaných každú hodinu.

Ďalej budeme pracovať s logaritmickej strednou cenou y_m , ktorá dá vypočítať nasledovne:

$$y_m = \frac{\log(p_{ask}) + \log(p_{bid})}{2} \quad (2.12)$$

kde p_{bid} a p_{ask} sú bid a ask ceny US dolára vzhľadom na inú menu. Keďže chceme porovnávať rôzne dátové súbory, normalizovali sme ich medzi 0 a 1 a dostali sme

AUD/USD	USD/CHF
EUR/USD	USD/JPY
GBP/USD	USD/NOK
NZD/USD	USD/SEK
USD/CAD	USD/SGD

Tabuľka 2.1: Uvažované miery výmenných kurzov.

Dát. súbor	τ	m	Dát. súbor	τ	m
AUD	126	8	CHF	150	9
EUR	127	9	JPY	144	8
GBP	140	12	NOK	106	12
NZD	134	10	SEK	130	10
CAD	127	10	SGD	108	10

Tabuľka 2.2: Časový posun τ a dimenzia vnorenia m pre výmenné kurzy

normalizovanú logaritmickú strednú cenu y :

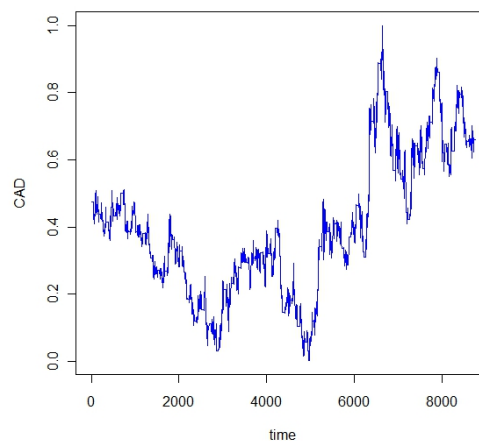
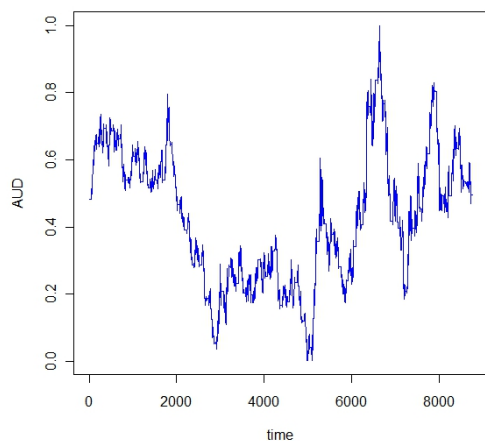
$$y = \frac{y_m - \min(y_m)}{\max(y_m) - \min(y_m)} + \delta \quad (2.13)$$

Hodnota δ (0,001) je potrebná, aby sme sa vyhli deleniu nulou pri výmene z jednej meny do druhej. Výmenné kurzy AUD/USD, EUR/USD, GBP/USD a NZD/USD sme invertovali a použili rovnakú transformáciu popísanú vyššie. Každý dátový súbor obsahuje 7860 bodov, t.j. 24 bodov korešponduje jednému dňu, a teda napr. 15. februáru zodpovedajú body 1083 až 1106.

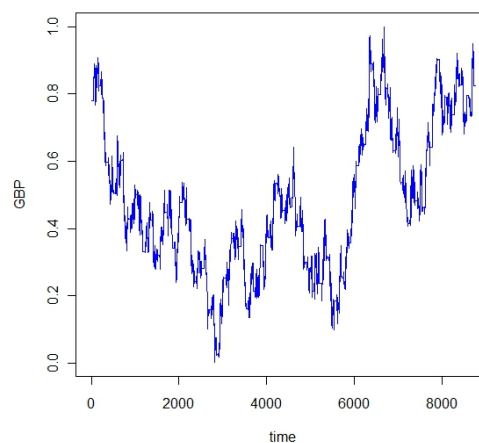
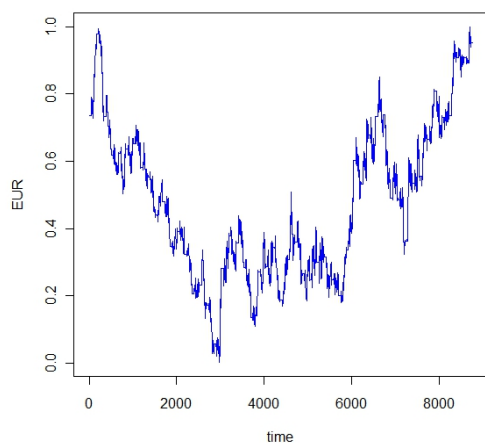
2.3.2 Vnárание výmenných kurzov

Na praktické aplikácie týchto metód sme používali program R [15] a softvérové balíčky *tseriesChaos* [16] a *RTisean* [17], ktoré sú implementáciou softvérového balíčka TISEAN [18] do prostredia programu R.

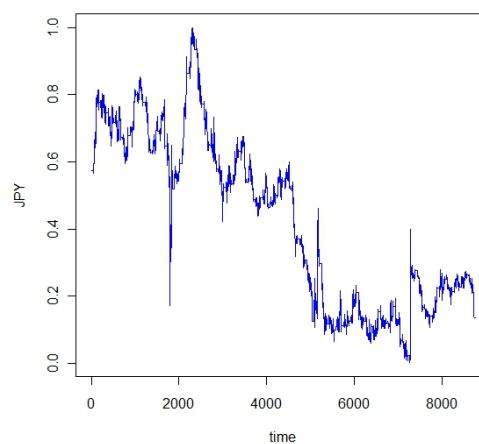
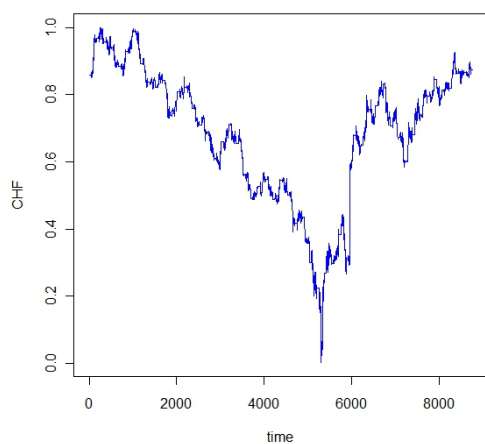
Na nájdenie vhodného časového posunu sme využili prvé minimum MI. Použili sme funkciu *mutual* z balíčka *tseriesChaos*. Výsledky sú zhrnuté v tabuľke 2.2. Priemerný lag 129.2, čo zodpovedá približne jednému týždňu (5 dní) pozorovaní. Tento časový posun sa zhoduje s typickým lagom, ktorý sa získa pri analýze denných dát, takže z pohľadu dynamických systémov, použitie dát s vyššou frekvenciou znamená, že sme dáta zaznamenávali vo vyššej frekvencii než bolo nutné. Ďalej dimenziu vnorenia sme vypočítali metódou FNN použitím funkcie *false.nearest* opäť z balíčka *tseriesChaos*. Za dobrú dimenziu sme považovali takú, pri ktorej podiel



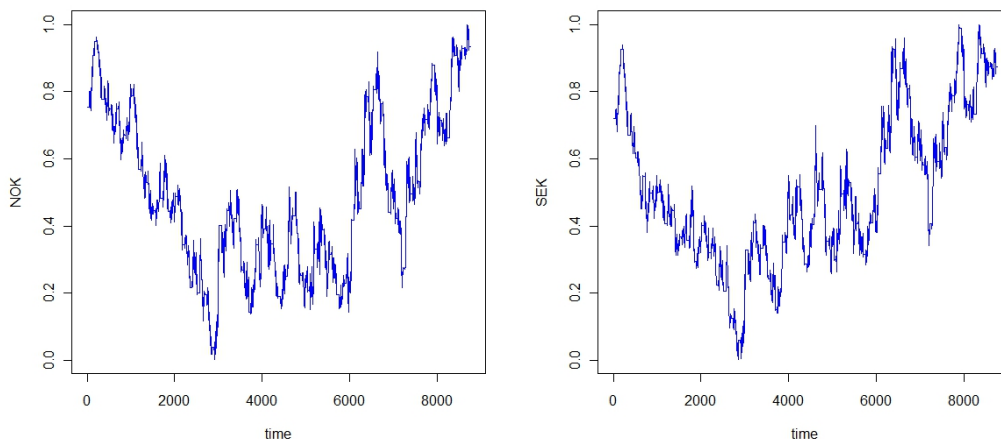
(a) čas. rad výmenného kurzu USD/AUD (b) čas. rad výmenného kurzu USD/CAD



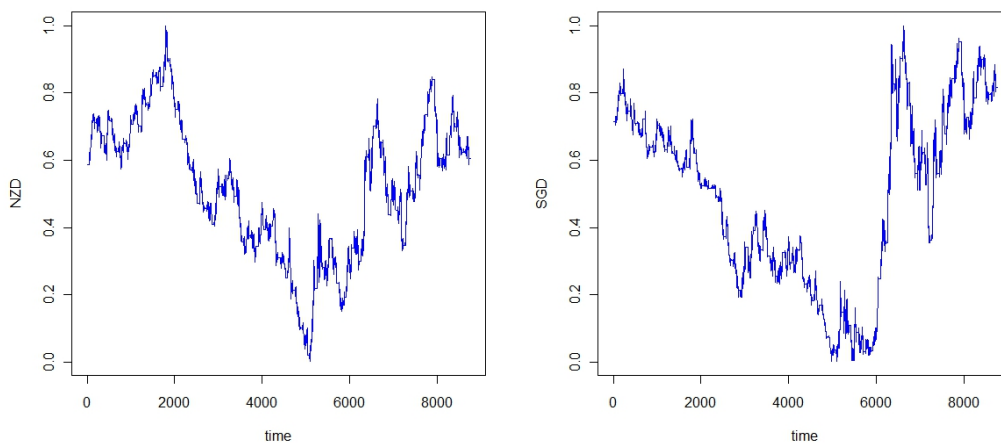
(c) čas. rad výmenného kurzu USD/EUR (d) čas. rad výmenného kurzu USD/GBP



(e) čas. rad výmenného kurzu USD/CHF (f) čas. rad výmenného kurzu USD/JPY



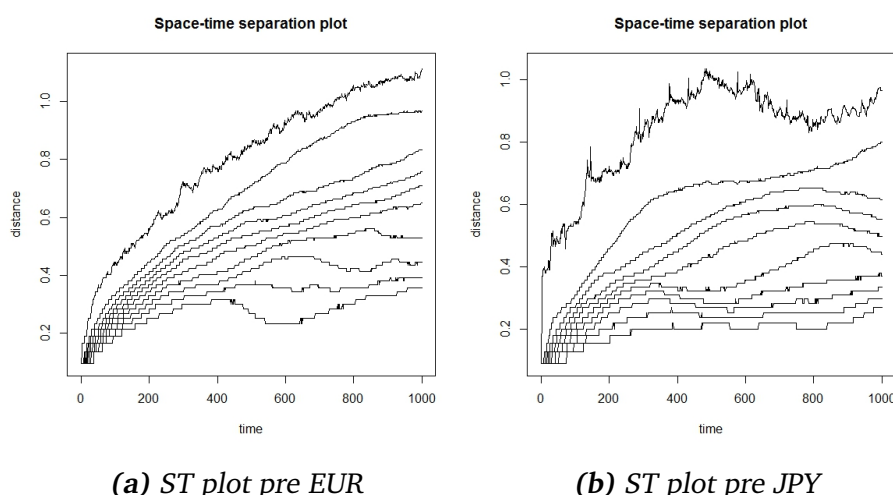
(a) čas. rad výmenného kurzu USD/NOK (b) čas. rad výmenného kurzu USD/SEK



(c) čas. rad výmenného kurzu USD/NZD (d) čas. rad výmenného kurzu USD/SGD

falošných najbližších susedov zo všetkých susedov klesol pod jedno percento (1%-ný podiel najbližších falošných susedov sa často považuje za vhodnú hranicu [19]). Vidíme, že všetky dátové súbory majú vysokú dimenzionalitu, $m \geq 8$. Tieto hodnoty sú v súlade aj s podobnou analýzou vykonanou Strozzi [20].

Ďalej sme na naše dáta použili priestorovo-časový separačný graf (*space-time separation plot*) [11]. Na obrázkoch 2.5a, 2.5b môžeme vidieť výsledky pre niektoré meny. Horizontálna os reprezentuje separačný čas a vertikálna os logaritmus pri základe 2 separácie v priestore. Aplikovali sme funkciu *stplot* z balíka *tseriesChaos*, ktorá kreslí pravdepodobnosť, že dva body v rekonštruovanom priestore majú menšiu vzdialenosť ako ε vo funkcii dvoch premenných ε a času Δt medzi týmito bodmi ako izočiaru na hladinách 10%, 20%,...,100%.



Obr. 2.5: Space-time separation plot

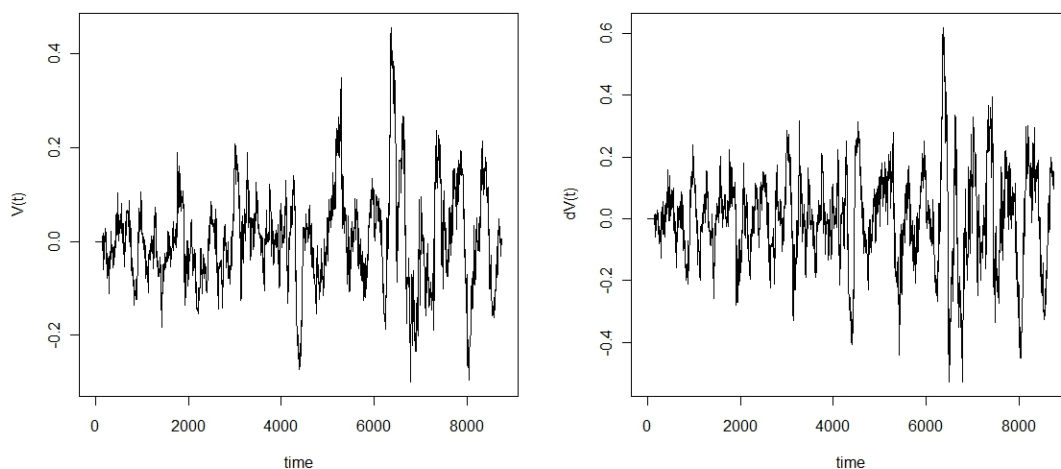
Mena	%gain	τ^{opt}	m^{opt}	Σg^{opt}
AUD	96,62	29	7	18,50
EUR	92,43	50	7	30,70
GBP	95,34	67	1	24,86
NZD	90,69	95	15	13,80
CAD	96,55	139	1	27,79
CHF	85,19	51	8	31,10
JPY	93,20	118	1	42,19
NOK	95,64	200	11	19,16
SEK	93,87	115	3	37,24
SGD	96,28	116	11	25,63

Tabuľka 2.3: %gain vyjadruje počet koľkokrát bol čistý výnos kladný pre všetky možné kombinácie parametrov (t.j. τ medzi 2 a 200 a m medzi 1 a 15). τ^{opt} a m^{opt} sú optimálne hodnoty parametrov τ a m , pre ktoré bola dosiahnutá najvyššia hodnota gain-loss funkcie, Σg^{opt} .

Môžeme vidieť, že krivky nesaturujú. To značí, že naše dáta nie sú stacionárne.

Ďalej sme použili Stroziovej obchodnú stratégiu na tieto výmenné kurzy. Počet transakcií sme žiadnym spôsobom nelimitovali. Keď $\Delta V(t)$ kleslo, tak sme naše aktíva zmenili v čase $t + 1$ na $menu_2$, v opačnom prípade na $menu_1$. Testovali sme gain-loss funkciu, rovnica (2.10), pre hodnoty časového posunu medzi 2 a 200 a dimenzie vnorenia medzi 1 a 15.

V tabuľke 2.3 sú výsledky pre každý výmenný kurz. Hodnoty v prvom stĺpci vyjadrujú, v koľkých percentách prípadov bol výnos kladný zo všetkých možných kombinácií parametrov (τ od 2 po 200 a m medzi 1 a 15). Priemerne sme kladný



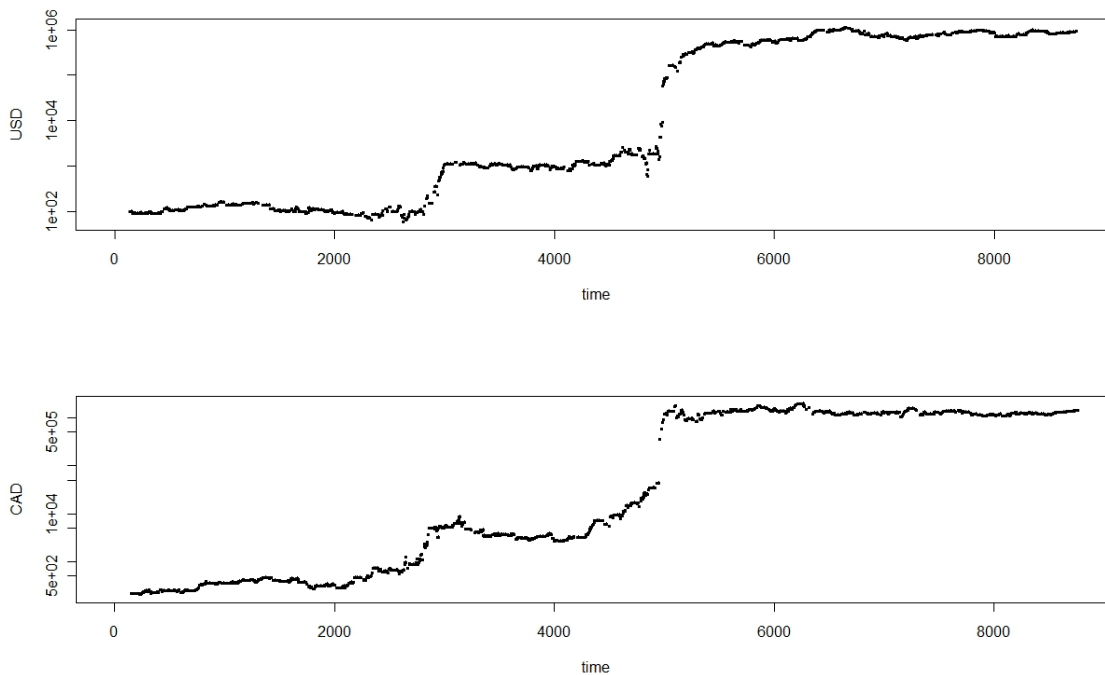
Obr. 2.6: Objem stavového priestoru pre normalizovaný výmenný kurz USD/CAD, parametre: $\Delta t = 139$, $m = 1$

výnos získali v 94 %. V ďalších stĺpcoch sú uvedené optimálne hodnoty časového posunu τ a dimenzie vnorenia m . V poslednom stĺpci je uvedená optimálna hodnota gain-loss funkcie. Vzhľadom na fakt, že sme skúmané časové rady normalizovali a pokrývajú celý interval od 0,001 po 1,001, tieto vysoké výnosy nemožno považovať ako reprezentatívne hodnoty pre reálne časové rady.

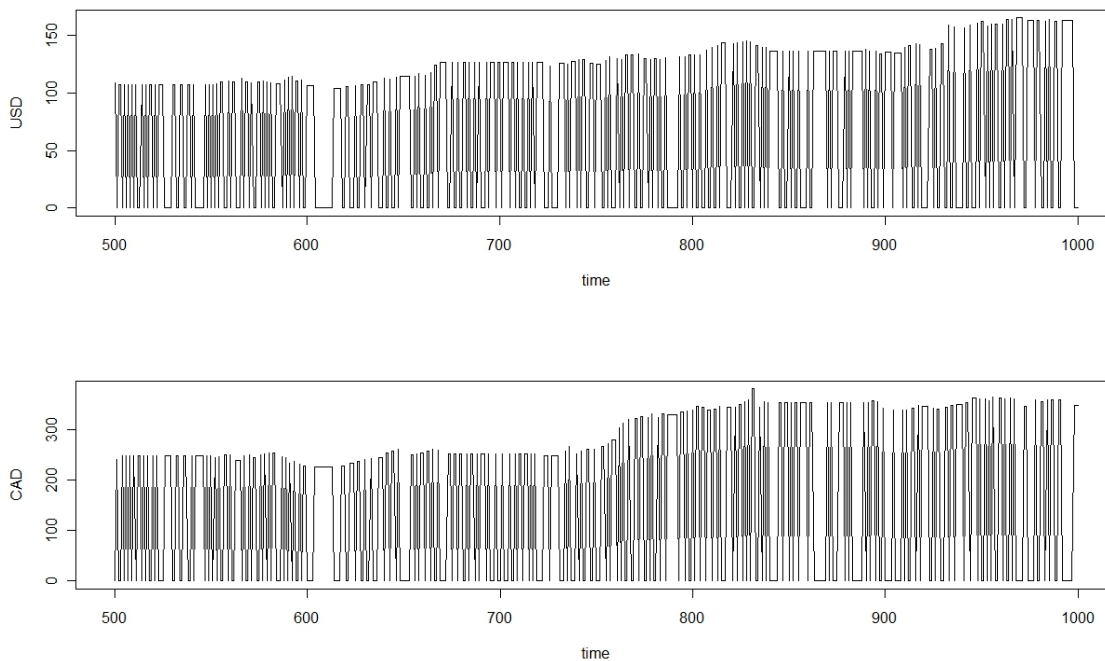
Obrázky (2.6) až (2.10) ukazujú dva výsledky, ktoré korešpondujú časovým radom USD/CAD a USD/JPY, pričom sme použili optimálne parametre nájdené pomocou Strozziovej stratégie. Hoci oba časové rady majú rôzny priebeh, v oboch prípadoch sa dosiahol kladný čistý zisk. Je zrejmé, že s takto definovanou stratégiou, nemáme žiadne obmedzenie na počet transakcií, viď obr. (2.8). Ak by sa v takto definovanej obchodnej stratégii uvažovali transakčné náklady, bolo by obtiažne dosiahnuť zisk.

Na porovnanie predpovedateľnosti týchto výmenných kurzov a náhodnej prechádzky sme vygenerovali 10 časových radov náhodných prechádzok. Tieto náhodné prechádzky sme normalizovali rovnakým spôsobom ako výmenné kurzy. S týmito náhodnými časovými radmi sme sledovali tú istú obchodnú stratégiu popísanú vyššie. Výsledky sú zaznamenané v tab. (2.4).

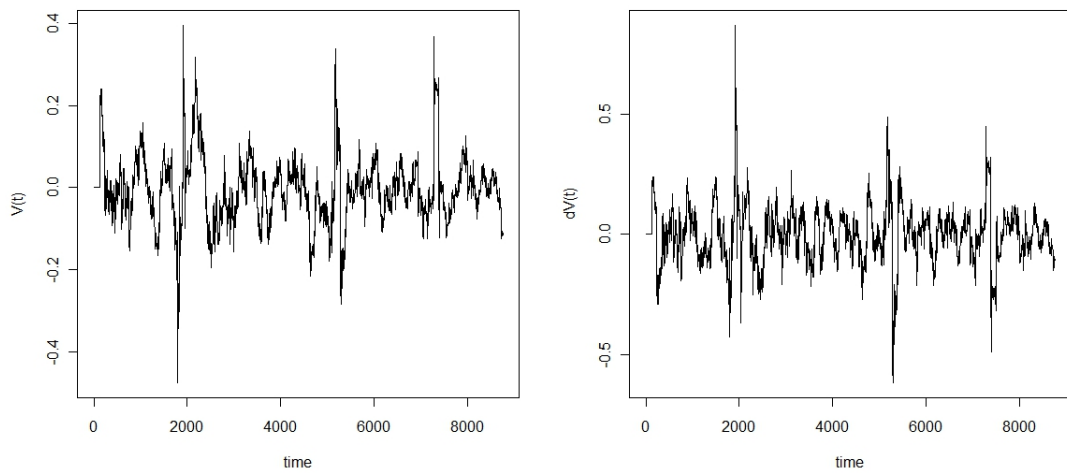
Na to, aby sme dokázali rozlíšiť dané súbory časových radov (t.j. súbor výmenných kurzov a súbor náhodných prechádzok), testovali sme nulovú hypotézu, že medián našich finančných dát sa rovná mediánu časových radov náhodných prechádzok pri použití optimálnych parametrov. Na prijatie alebo zamietnutie tejto nulovej



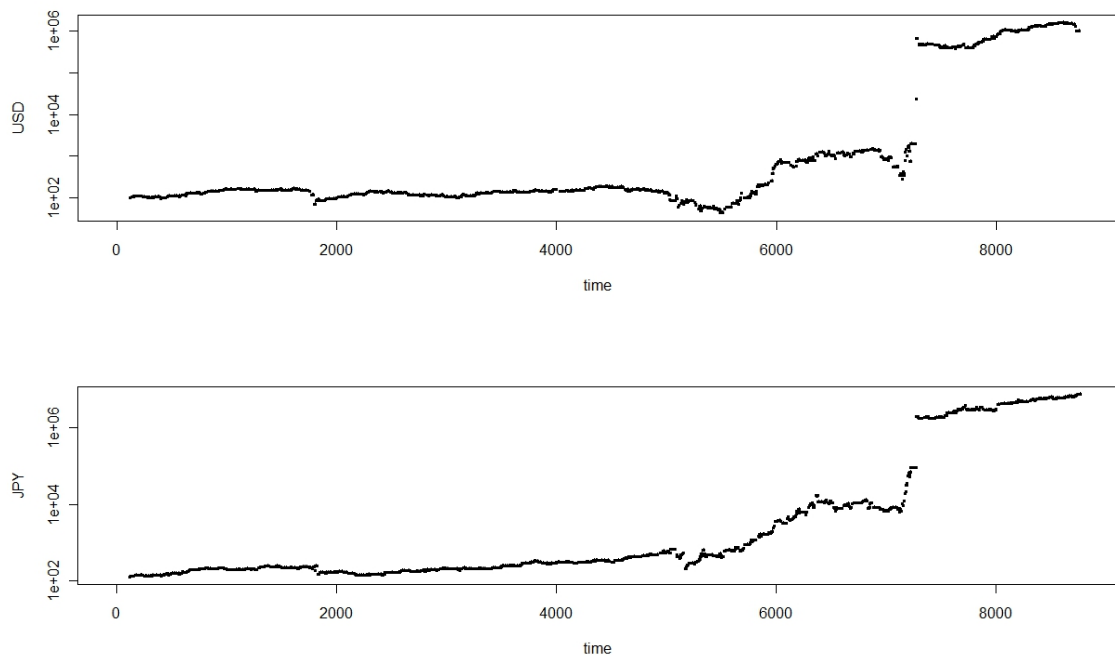
Obr. 2.7: Výmeny medzi CAD a USD v logaritmickej mierke sledujúc vyššie definovanú stratégiu a začínajúc so 100 USD, parametre: $\Delta t = 139$, $m = 1$.



Obr. 2.8: Detail výmen medzi CAD a USD medzi časmi 500 a 1000.



Obr. 2.9: Objem stavového priestoru pre normalizovaný výmenný kurz USD/JPY, parametre: $\Delta t = 118, m = 1$



Obr. 2.10: Výmeny medzi JPY a USD v logaritmickej mierke sledujúc vyššie definovanú stratégiu a začínajúc so 100 USD, parametre: $\Delta t = 118, m = 1$.

Mena	%gain	τ^{opt}	m^{opt}	Σg^{opt}
random1	97,05	20	4	92,30
random2	90,92	198	4	69,34
random3	95,41	87	4	115,52
random4	98,39	123	10	38,52
random5	94,97	142	4	21,92
random6	77,29	17	10	104,74
random7	87,97	83	15	71,32
random8	97,59	73	8	46,71
random9	81,51	148	14	87,37
random10	94,04	141	3	55,25

Tabuľka 2.4: %gain vyjadruje počet koľkokrát bol čistý výnos kladný pre všetky možné kombinácie parametrov (t.j. τ medzi 2 a 200 a m medzi 1 a 15). τ^{opt} a m^{opt} sú optimálne hodnoty parametrov τ a m , pre ktoré bola dosiahnutá najvyššia hodnota gain-loss funkcie, Σg^{opt} .

hypotézy sme použili neparametrický znamienkový test na hodnoty %gain a Σg^{opt} . Danú nulovú hypotézu zamietame na hladine významnosti 5%, ak $|\frac{n^*}{n} - \frac{1}{2}| > \frac{1}{\sqrt{n}}$, kde n^* je počet pozorovaní menších ako medián náhodných prechádzok a n je celkový počet pozorovaní.

Pre %gain danú nulovú hypotézu na hladine významnosti 5% nezamietame, pretože $|\frac{5}{10} - \frac{1}{2}| = 0 < \frac{1}{\sqrt{10}} \approx 0,32$. Pre Σg^{opt} získavame $|\frac{10}{10} - \frac{1}{2}| = 0,5 > \frac{1}{\sqrt{10}}$, teda v tomto prípade nulovú hypotézu zamietame.

Vnárание ekonomických časových radov

3.1 Metóda najbližších susedov

Metóda najbližších susedov (*nearest neighbor method, NN*) je jednou z neparametrických metód využívaná prevažne na nelineárne predikcie. Inšpiráciou pre túto metódu boli predikcie nelineárnych dynamických systémov [21] a snaží sa predpovedať daný časový rad tým, že analyzuje, ako sa vyvíjal v podobných situáciách v minulosti. Takže stačí zobrať posledný dostupný údaj časového radu a vyhľadať mu k najviac podobných, ktoré sa nazývajú najbližší susedia. Ďalej sa zistí, ktoré hodnoty nasledovali po týchto najbližších susedoch. Na základe tejto informácie sa usúdi, na akú hodnotu sa časový rad dostane.

Nech $x_t, t = 1, \dots, n, n \in \mathbb{N}$ je časový rad. Na to, aby sme v ňom určili určité vzory vývinu, správania, vytvoríme pomocou neho vektory vnorenia \mathbf{x}_t

$$\mathbf{x}_t = (x_{t-(m-1)\tau}, \dots, x_{t-\tau}, x_t) \quad t = 1 + \tau(m-1), \dots, n \quad (3.1)$$

kde m je dimenzia vnorenia a τ je časový posun. Predikcia je určená analyzovaním vývinu vektorov okolo posledného dostupného vektora

$$\mathbf{x}_n = (x_{n-(m-1)\tau}, \dots, x_{n-\tau}, x_n) \quad (3.2)$$

To znamená, že nájdeme k vektorov $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots, \mathbf{x}_{i_k} \in \mathbb{R}^m$, ktoré majú najmenšiu vzdialenosť od vektora \mathbf{x}_n . V literatúre môžeme nájsť rôzne odporúčania, ktorú normu použiť. Casdagli [22] odporúča použiť *max* normu kvôli jej výpočtovej jednoduchosti. Iní (napr. Yakowitz [23]) sú zástancami Euklidovskej normy. My sme sa

držali posledného odporúčania, pretože sa často používa pri predpovedaní finančných časových radov. Formálne, k najbližších susedov k poslednému dostupnému vektoru \mathbf{x}_n budú také vektory, ktoré minimalizujú funkciu

$$d_i = \|\mathbf{x}_{i_k} - \mathbf{x}_n\| = \left(\sum_{l=1}^m (x_{l,i_k} - x_{l,n})^2 \right)^{1/2} \quad (3.3)$$

Najjednoduchším prípadom tejto metódy je prípad $k = 1$. Nech najbližším susedom vektora \mathbf{x}_n je vektor \mathbf{x}_K . Potom predikcia pozorovania x_n bude výlučne nasledovné pozorovanie bodu x_K , t.j.

$$\hat{x}_{n+1} = x_{K+1} \quad (3.4)$$

Táto metóda je veľmi intuitívna a jednoduchá. Avšak túto metódu sme nepoužili na predikciu, pretože pre krátke časové rady a/alebo rady so šumom tento spôsob predikcie má obmedzenú predikčnú silu. Zlepšením je vziať jednoducho priemer k najbližších susedov

$$\hat{x}_{n+1} = \frac{x_{i_1+1} + x_{i_2+1} + \dots + x_{i_k+1}}{k} \quad (3.5)$$

alebo môžeme uvažovať vážený priemer, pričom jednotlivé váhy sú určené vzhľadom na vzdialenosť najbližšieho suseda k bodu x_n

$$\hat{x}_{n+1} = w_1 x_{i_1+1} + w_2 x_{i_2+1} + \dots + w_k x_{i_k+1} \quad (3.6)$$

kde

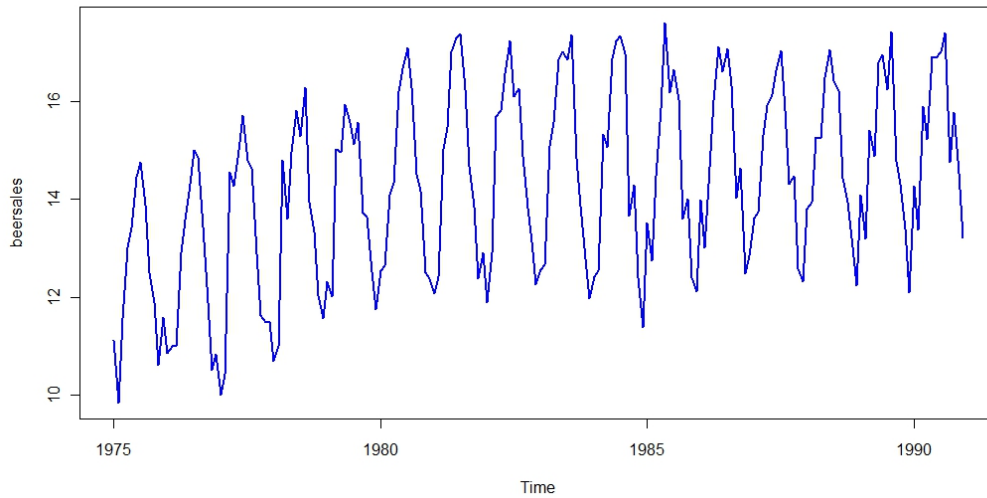
$$w_l = \frac{\langle \mathbf{x}_{i_l}, \mathbf{x}_n \rangle}{\sum_{j=1}^k \langle \mathbf{x}_{i_j}, \mathbf{x}_n \rangle} \quad (3.7)$$

Tento model ilustrujme na jednoduchom časovom rade *beersales*, ktorý sa nachádza v balíčku *TSA* [24] programu R. V tomto časovom rade je zaznamenaný mesačný predaj piva v miliónoch barelov od januára 1975 do decembra 1990 v USA, t.j. spolu 192 údajov, obr. (3.1).

Jedným z kľúčových konceptov analýzy časových radov je rozklad daného časového radu x_t na trend T_t , sezónny komponent S_t a zostatok e_t .

$$x_t = T_t + S_t + e_t \quad (3.8)$$

Jednou z možností ako vypočítať trend časového radu je použiť neparametrickú regresnú techniku tzv. "loess"regresiu. Tento rozklad sme vypočítali pomocou funkcie *stl()*.



Obr. 3.1: Mesačný predaj piva v mil. bareloch 1/1975-12/1990

Odstránili sme teda trend a ďalej sme pracovali iba s časovým radom, ktorý bol zložený z komponentov *seasonal* a *remainder*. Tento časový rad sme rozdelili na tréningovú časť ($t = 1, \dots, 156$) a testovaciu časť ($t = 157, \dots, 192$). Na tréningovej časti sme hľadali optimálne parametre pre danú metódu predpovedania a na testovacej časti sme zisťovali jej presnosť.

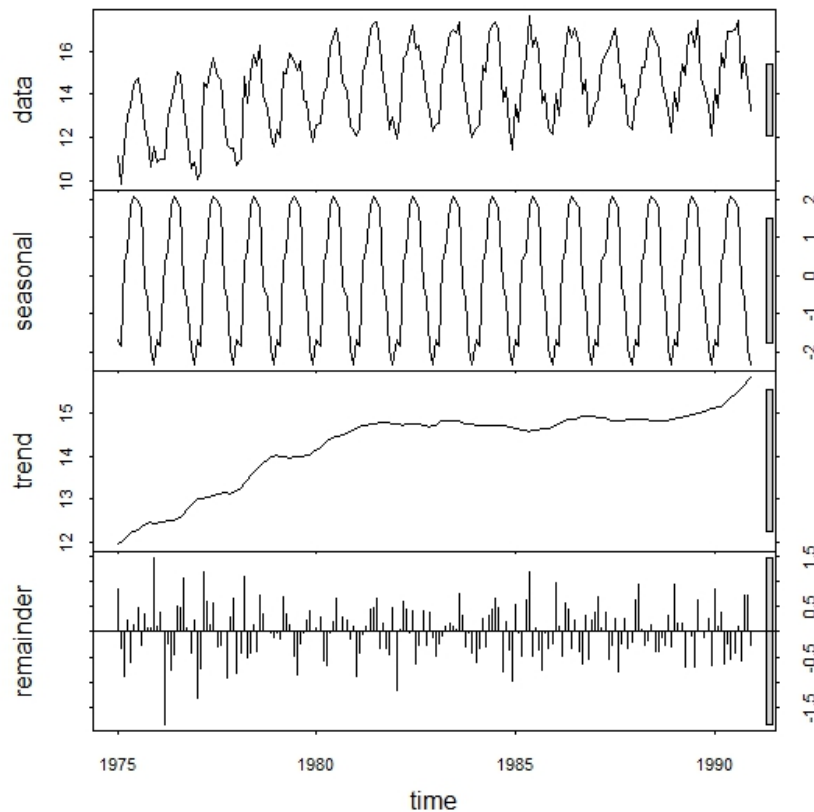
Presnosť predpovede možno kvantifikovať viacerými veličinami, konkrétne sú to chyby predikcie (*prediction error*). Často používanou je *root mean squared prediction error* (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2} \quad (3.9)$$

kde \hat{x}_i je predikcia a x_i je skutočná hodnota danej premennej. Ďalšie sú stredná absolútna chyba (*mean absolute error*) $\frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i|$ alebo logaritmická chyba (*logarithmic error*) $\frac{1}{n} \sum_{i=1}^n \ln |\hat{x}_i - x_i|$. My sme na porovnanie rôznych metód predikcie použili RMSE.

Aby sme mohli využiť NN metódu potrebujeme ďalej zistiť hodnoty m , τ a k . Na určenie m a τ použijeme metódy popísané vyššie. Avšak na stanovenie k neexistuje všeobecné pravidlo. V literatúre sa uvádza odporúčanie $k \geq 2(m + 1)$ [22].

Optimálny parameter časového posunu sme hľadali na tréningovej časti časového radu pomocou funkcie *mutual* a ako optimálna hodnota vyšla $\tau = 2$. Na nájdenie optimálnej dimenzie vnorenia sme použili funkciu *false.nearest*, $m = 4$. Obe tieto funkcie sú súčasťou balíka *tseriesChaos*.



Obr. 3.2: Rozklad čas. radu beersales pomocou $stl()$

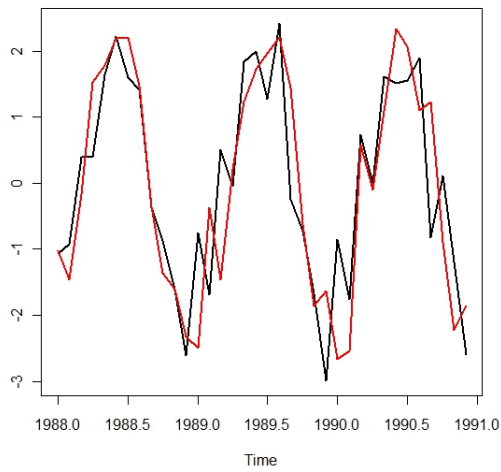
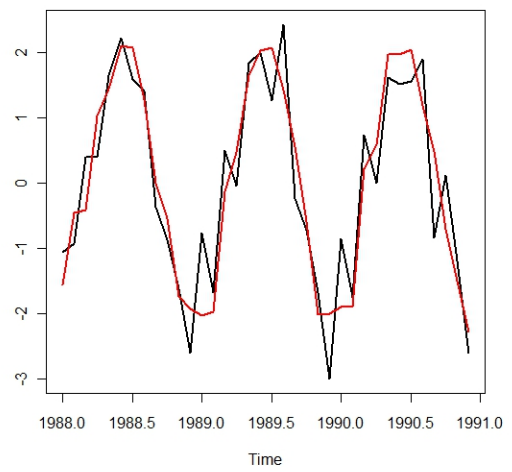
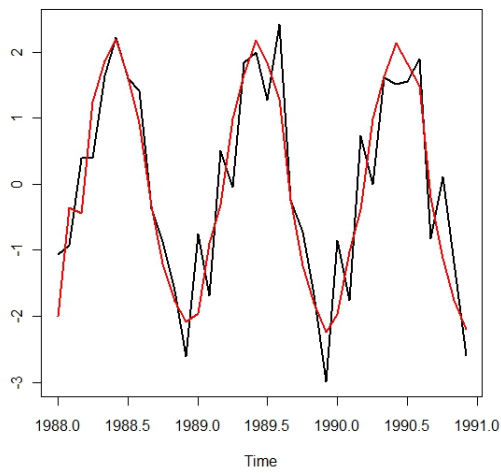
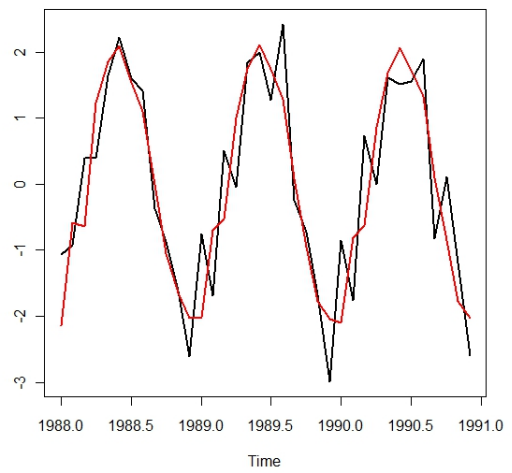
Tieto hodnoty sa použili na vytvorenie vektorov vnorení

$$\mathbf{x}_i = (x_{i-6}, x_{i-4}, x_{i-2}, x_i) \quad i = 7, \dots, 156 \quad (3.10)$$

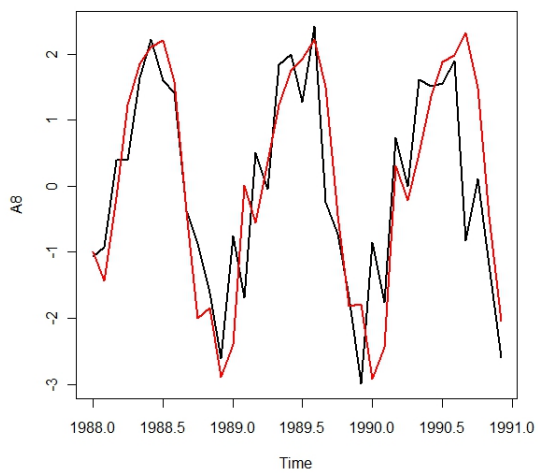
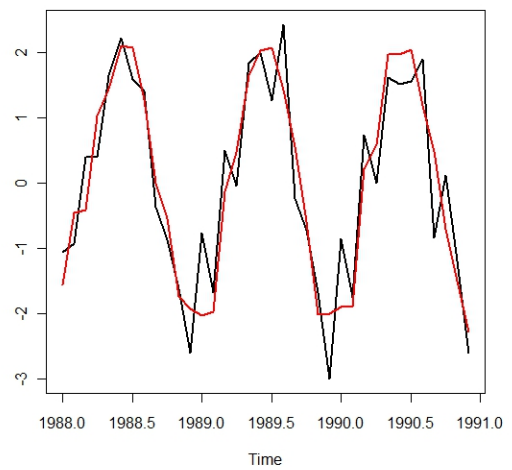
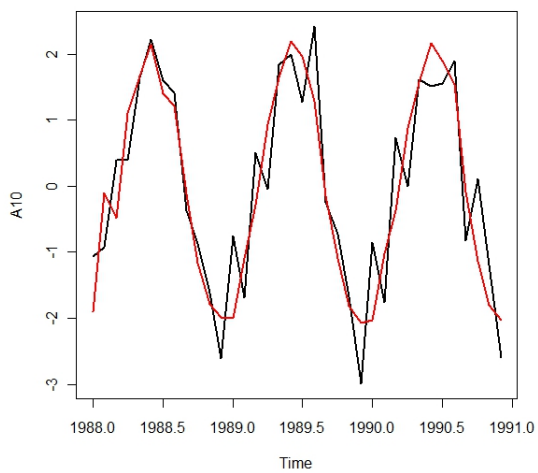
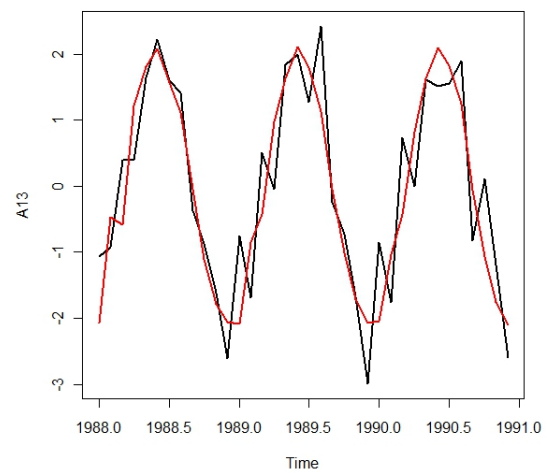
Najskôr sme robili predikcie pomocou aritmetického priemeru najbližších susedov. Pre posledný dostupný vektor \mathbf{x}_{156} sa našlo k vektorov, ktoré mali od neho najmenšiu vzdialenosť a predikcia $\hat{\mathbf{x}}_{157}$ pre vektor \mathbf{x}_{157} sa vypočítala ako priemer po nich nasledujúcich pozorovaní. Potom sa vyhľadalo k najbližších susedov pre vektor $\hat{\mathbf{x}}_{157}$ a opäť priemer ich nasledujúcich meraní bol predikcia $\hat{\mathbf{x}}_{158}$ atď. Presnosť týchto predikcií pre rôzne k môžeme vidieť na obr (3.3) a RMSE v tab. (3.1).

Ďalej sme na vypočítanie predikcií použili vážený priemer NN. Pozorovanie nasledujúce po NN s najmenšou vzdialenosťou malo najvyššie váhy a naopak, pozorovanie idúce po k -tom NN dostalo váhy najmenšie. Výsledky pre rôzne k možno vidieť na obr. (3.4) a v tab. (3.1).

Najmenšie RMSE spomedzi $k = 1, \dots, 15$ sa dosiahlo pri $k = 4$ v oboch prípadoch. Najmenej presná predikcia sa získala použitím dvoch susedov v oboch varian-

**(a)** $k = 2$ **(b)** $k = 4$ **(c)** $k = 10$ **(d)** $k = 15$

Obr. 3.3: Čiernou farbou je zakreslený pôvodný čas. rad, červenou predikcia vypočítaná pomocou priemeru NN pre rôzne k .

**(a)** $k = 2$ **(b)** $k = 4$ **(c)** $k = 10$ **(d)** $k = 15$

Obr. 3.4: Čiernou farbou je zakreslený pôvodný čas. rad, červenou predikcia vypočítaná pomocou váženého priemeru NN pre rôzne k .

kNN		wkNN	
k	RMSE	k	RMSE
2	0,90585	2	0,97843
4	0,61633	4	0,63822
6	0,71314	6	0,69271
8	0,68724	8	0,67757
10	0,68090	10	0,68680
12	0,76086	12	0,69951
15	0,72853	15	0,70865

Tabuľka 3.1: RMSE

toch metódy NN.

3.2 ARMA model

Náš časový rad *beersales* očistený od trendu sme modelovali pomocou ARMA modelu. Autoregressive Moving Average (ARMA) model pozostáva z dvoch častí, z autoregresnej (AR) časti a z časti kľzavých priemerov (moving average (MA)). Zvyčajne sa označuje ako $ARMA(p,q)$, kde p je rád AR časti a q je rád MA časti.

Definícia $ARMA(p,q)$ modelu

$$x_t = \delta + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \dots + \alpha_p x_{t-p} + u_t - \beta_1 u_{t-1} - \beta_2 u_{t-2} - \dots - \beta_q u_{t-q} \quad (3.11)$$

pričom u_t je biely šum a $\alpha_p \neq 0$, $\beta_q \neq 0$. Využitím lag operátora získame

$$(1 - \alpha_1 L - \alpha_2 L^2 - \dots - \alpha_p L^p)x_t = \delta + (1 - \beta_1 L - \beta_2 L^2 - \dots - \beta_q L^q)u_t \quad (3.12)$$

alebo

$$\alpha(L)x_t = \delta + \beta(L)u_t \quad (3.13)$$

pričom uvedené polynómy nemôžu mať spoločný koreň. $ARMA(p,q)$ proces je stacionárny, ak je splnená podmienka stacionarity AR časti, t.j. korene $\alpha(L)$ ležia mimo jednotkového kruhu.

Nestacionárne časové rady možno previesť na stacionárne diferencovaním. Na ich modelovanie možno použiť $ARIMA(p,d,q)$ modely (Autoregressive Integrated Moving Average)

$$\alpha(L)(1 - L^d)x_t = \delta + \beta(L)\varepsilon_t \quad (3.14)$$

Na modelovanie sezónnych dát existujú špeciálne modely: sezónne $ARIMA$ modely, $ARIMA(p,d,q)(P,D,Q)_s$, kde P je počet sezónnych AR členov, D sa dáta sezónne

diferencujú, Q počet sezónnych MA členov a s je perióda dát. Sezónna ARIMA má tvar:

$$A(L^s)\alpha(L)(1 - L^s)^D(1 - L)^d x_t = \delta + B(L^s)\beta(L)\varepsilon_t \quad (3.15)$$

kde $A(z)$ je polynóm rádu P a $B(z)$ je polynóm rádu Q a žiaden z nich neobsahuje koreň vo vnútri jednotkového koreňa.

Na určenie rádu ARIMA modelu Box a Jenkins [25] využívali grafy autokorelačnej (ACF) a parciálnej autokorelačnej funkcie (PACF) časového radu. Identifikácia ARIMA modelu pomocou grafov ACF a PACF je však zložitá a vyžaduje veľa skúseností v modelovaní časových radov.

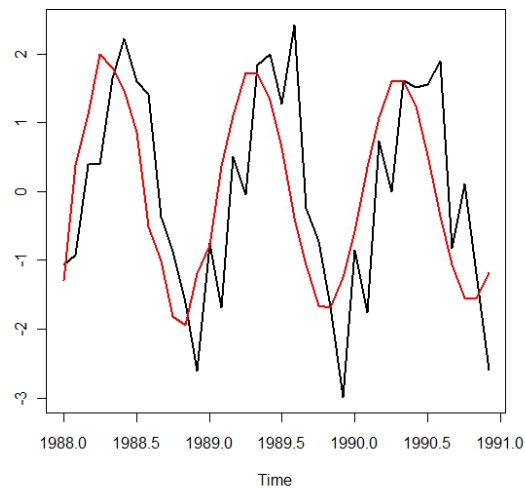
Ďalej možno na určenie správneho rádu ARIMA modelu použiť Akaikeho informačné kritérium $AIC = 2k - 2\ln(L)$, kde k je počet parametrov modelu a L je maximalizovaná hodnota funkcie vierohodnosti pre odhadnutý model. Ak pre danú množinu dát máme viacero modelových kandidátov, vyberieme ten model, ktorý má AIC minimálne.

Na odhad parametrov modelu sa používajú komplikované iteračné procedúry [26]. V súčasnosti je ich odhad v praxi pomerne jednoduchý, pretože sa vykonáva pomocou počítačových programov.

Na nájdenie vhodného ARMA modelu sme použili funkciu *auto.arima* z balíka *forecast* [27]. Algoritmus, ktorým táto funkcia hľadá najlepší model spomedzi všetkých možných možno nájsť v [28]. Ako najvhodnejší model vyšiel ARIMA $(5, 0, 5)(1, 0, 0)_{12}$ s hodnotou Akaikeho informačného kritéria 2359,66. Všetky korene polynómov $\alpha(L)$ a $\beta(L)$ sú v absolútnej hodnote väčšie ako 1, teda tento proces je stacionárny a invertovateľný. Na výpočet predikcie sme využili funkciu *forecast.Arima* opäť z balíka *forecast*. Výsledok možno vidieť na obr. (3.5) a RMSE v tomto prípade vyšlo 1,205982.

3.3 Štrukturálny prístup

Podrobný popis tohto prístupu možno nájsť v [29] a [30]. Tu uvedieme iba stručný prehľad. Základným modelom v štrukturálnom prístupe je tzv. *local level model*. V tomto modeli sa komponent *level* môže meniť v čase. Tento komponent možno chápať ako priesečník regresnej priamky s osou y v klasickom regresnom modeli. Významný rozdiel medzi nimi je ten, že priesečník v regresii je fixný, zatiaľ čo *level* v štrukturálnom prístupe sa časom meniť môže.



Obr. 3.5: Čiernou farbou je zakreslený pôvodný čas. rad, červenou predikcia vypočítaná pomocou $ARIMA(5, 0, 5)(1, 0, 0)_{12}$ modelu

Local level model je definovaný

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \quad i.i.d. \\ \mu_{t+1} &= \mu_t + \xi_t, & \xi_t &\sim N(0, \sigma_\xi^2) \quad i.i.d. \end{aligned} \quad (3.16)$$

pre $t = 1, \dots, n$, kde μ_t je nepozorovaný *level* v čase t , ε_t je porucha pozorovania v čase t a ξ_t sa nazýva *levelová* porucha v čase t . Poruchy pozorovania aj *levelové* poruchy sú seriálne aj vzájomne nezávislé a normálne rozdelené so strednou hodnotou 0 a varianciou σ_ε^2 , resp. σ_ξ^2 .

V prípade, keď sú všetky poruchy $\xi_t = 0$ pre $t = 1, \dots, n$, model sa označuje ako deterministický a v tomto prípade sa *level* nemení v čase. V opačnom prípade, keď sa *level* časom mení, model je stochastický. V štrukturálnom prístupe neznámymi parametrami sú variancie porúch pozorovania σ_ε^2 a *levelu* σ_ξ^2 .

Pre deterministický model sa rovnica (3.16) zredukuje na tvar

$$y_t = \mu_1 + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad i.i.d. \quad (3.17)$$

pre $t = 1, \dots, n$. Teda v tomto špeciálnom prípade všetko závisí od hodnoty μ_1 , t.j. hodnoty *levela* v čase $t = 1$. Keď sa raz určí jej hodnota, zostane konštantná pre všetky ostatné časy $t = 2, \dots, n$.

Ďalším modelom v štrukturálnom prístupe je lokálny lineárny trend model, ktorý

sa získa pridaním komponentu trend/sklon ν_t do *local level* modelu

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \quad i.i.d. \\ \mu_{t+1} &= \mu_t + \nu_t + \xi_t, & \xi_t &\sim N(0, \sigma_\xi^2) \quad i.i.d. \\ \nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim N(0, \sigma_\zeta^2) \quad i.i.d. \end{aligned} \quad (3.18)$$

pre $t = 1, \dots, n$. Tento model obsahuje 2 stavové rovnice: jednu pre modelovanie *levelu* a druhú pre trend. Trend určuje uhol medzi trendovou priamkou a osou x , ktorý sa v čase môže meniť. V literatúre sa tiež označuje ako drift.

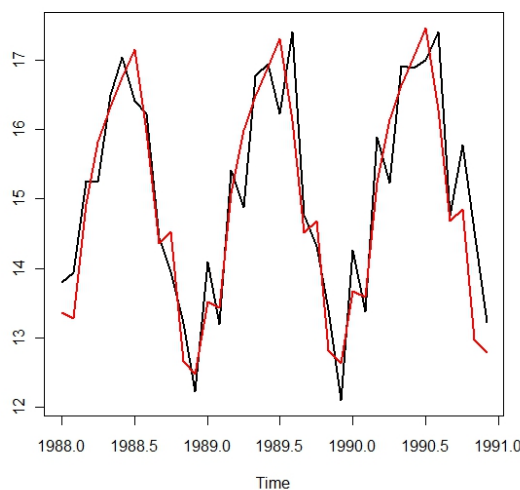
V štruktúrálnom prístupe možno sezónny efekt modelovať pridaním sezónneho komponentu buď do *local level* modelu alebo do trend modelu. V prípade štvrťročných dát má *local level* model so sezónnym komponentom tvar

$$\begin{aligned} y_t &= \mu_t + \gamma_t + \varepsilon_t & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \quad i.i.d. \\ \mu_{t+1} &= \mu_t + \xi_t & \xi_t &\sim N(0, \sigma_\xi^2) \quad i.i.d. \\ \gamma_{1,t+1} &= -\gamma_{1,t} - \gamma_{2,t} - \gamma_{3,t} + \omega_t & \omega_t &\sim N(0, \sigma_\omega^2) \quad i.i.d. \\ \gamma_{2,t+1} &= \gamma_{1,t} \\ \gamma_{3,t+1} &= \gamma_{2,t} \end{aligned} \quad (3.19)$$

pre $t = 1, \dots, n$, kde $\gamma_t = \gamma_{1,t}$ označuje sezónny komponent. Vďaka poruchám ω_t sa môže sezónny komponent meniť v čase. Počiatočné hodnoty μ_1 , $\gamma_{1,1}$, $\gamma_{2,1}$ a $\gamma_{3,1}$ sú fixné a neznáme koeficienty.

Na rozdiel od komponentov *level* a sklon, z ktorých každý v modeli potrebuje jednu stavovú rovnicu, sezónny komponent vo všeobecnosti vyžaduje v modeli ($s - 1$) stavových rovníc, kde s je perióda sezónnosti. Teda pre štvrťročné dáta ($s = 4$) sú potrebné 3 rovnice, ako vidno v rov. (3.19). Štvrtá a piata rovnica v (3.19) sú identity a možno ich interpretovať nasledovne: definujme $\gamma_{i,t}$ ako i -ty kvartál t časových období. Potom štvrtá rovnica v (3.19) nám hovorí, že kvartál v ďalšom období $t + 1$ je nasledujúci kvartál od aktuálneho obdobia t . Keďže je toto je fakt, nemožno pridať poruchy do týchto rovníc.

Na nájdenie vhodného modelu sme použili funkciu *StructTS* programu R. Získali sme nasledovný odhad variancií: $\sigma_\varepsilon^2 = 0,00$, $\sigma_\xi^2 = 0,260$, $\sigma_\zeta^2 = 0,0001209$ a $\sigma_\omega^2 =$



Obr. 3.6: Čiernou farbou je zakreslený pôvodný čas. rad, červenou predikcia vypočítaná pomocou štrukturálneho prístupu

0, 228. Teda naše dáta možno modelovať pomocou:

$$\begin{aligned}
 y_t &= \mu_t + \gamma_t \\
 \mu_{t+1} &= \mu_t + \nu_t + \xi_t, & \xi_t &\sim N(0, 0, 260) i.i.d. \\
 \nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim N(0, 0, 0001209) i.i.d. \\
 \gamma_{1,t+1} &= -\gamma_{1,t} - \gamma_{2,t} - \dots - \gamma_{11,t} + \omega_t, & \omega_t &\sim N(0, 0, 228) i.i.d. \\
 \gamma_{2,t+1} &= \gamma_{1,t} \\
 &\vdots \\
 \gamma_{11,t+1} &= \gamma_{10,t}
 \end{aligned} \tag{3.20}$$

Na predikciu sme použili funkciu *forecast*. Vypočítaná predikcia je zobrazená na obr. (3.6). RMSE v tomto prípade vyšlo 0,6393829.

3.4 Zhrnutie

Na predchádzajúcich stranách sme predikovali časový rad *beersales* pomocou metódy najbližších susedov, ARMA modelu a štrukturálneho prístupu. Aby bolo možné porovnať presnosť týchto metód, pre každú sme vypočítali RMSE. Na základe získaných RMSE pre metódu NN počítanú pomocou aritmetického priemeru a váženého priemeru, sa možno domnievať, že obe metódy majú podobnú presnosť. Pri použití iba dvoch najbližších susedov boli tieto spôsoby predikcie najmenej presné. Naopak, najpresnejšie výsledky sa dosiahli pri použití 4 najbližších susedov.

Zo všetkých spomínaných predikčných metód ARMA vyšla ako najmenej presná. Na druhej strane najpresnejšiu predikciu (t.j. najnižšie RMSE) sme získali pri použití metódy NN. Avšak rozdiel medzi RMSE získaným pomocou metódy NN a štruktúrneho prístupu je rádovo v stotínach.

Videli sme, že aj jednoduchý princíp predikcie založený na hľadaní podobných stavov v minulosti a spriemerovania po nich nasledujúcich stavov, môže viesť k porovnateľne presným výsledkom ako matematicky a výpočtovo zložité ARMA a štruktúrne modely.

Záver

V diplomovej práci sme popísali Takensovú vetu z teórie dynamických systémov. Táto matematická veta je základom pre vnáranie časových radov a rekonštrukciu stavového priestoru. Podľa tejto matematickej vety platí, ak $m \geq 2D + 1$, kde m je dimenzia vnorenia a D je dimenzia originálneho systému, ktorý je predmetom skúmania, tak zrekonštruovaný a originálny stavové priestory sú difeomorfné. To znamená, že stačí mať súbor skalárnych meraní, z ktorých je možné zostaviť nový stavový priestor topologicky identický s originálnym (ale neznámym) priestorom. Uviedli sme metódy na odhad parametrov vnorenia, lagu τ a dimenzie vnorenia m . Lag možno odhadnúť pomocou funkcie vzájomnej informácie, dimenzia vnorenia sa odhaduje použitím metódy falošných najbližších susedov. Následne sme tieto metódy ilustrovali na finančných časových radoch, na výmenných kurzoch medzi americkým dolárom a desiatimi inými menami v roku 2011. Získané výsledky vyšli v súlade s obdobnou analýzou vykonanou v [20].

Teóriu vnorenia sme využili v obchodnej stratégii navrhnujej F. Stozziovou. Ilustrovali sme, že pri tejto stratégii je možné získať čistý zisk (pri nulových transakčných nákladoch). Základom tejto stratégie bol odhad objemu stavového priestoru ako determinant medzi blízkymi bodmi v stavovom priestore. Princíp obchodovania s výmennými kurzami podľa tejto stratégie bol meniť naše aktíva z jednej meny do druhej podľa toho, ako sa menil objem stavového priestoru. Videli sme, že pri všetkých výmenných kurzoch sa pri istých parametroch vnorenia dosiahol čistý zisk. Na porovnanie predpovedateľnosti týchto výmenných kurzov s náhodnou prechádzkou sme vygenerovali 10 časových radov náhodných prechádzok, na ktorých sme tiež použili túto obchodnú stratégiu. Testovali sme nulovú hypotézu, že medián výmenných kurzov sa rovná mediánu náhodných prechádzok pri použití optimálnych parametrov vnorenia. Znamienkový test danú hypotézu pre %gain nezamietol, pre $\sum g^{opt}$ zamietol. To znamená, že nemožno tvrdiť, že naše finančné časové rady majú rovnaké správanie ako náhodná prechádzka.

V záverečnej časti sme porovnali predikčnú presnosť metódy najbližších suse-

dov, ARMA modelu a modelu založeného na štrukturálnom prístupe. Presnosť sme porovnávali na základe hodnoty RMSE (*root mean square error*). Pomocou metódy najbližších susedov sme najpresnejšie výsledky dosiahli použitím 4 najbližších susedov. Získaná predikcia pomocou ARMA modelu vyšla vzhľadom na RMSE najmenej presne. Avšak rozdiely v RMSE medzi jednotlivými spôsobmi predikcie boli malé, rádovo desatiny až stotiny. Všetky tri metódy predikcie mali v našom prípade podobnú presnosť. Na základe týchto výsledkov môžeme konštatovať nasledovné: jednoduchá idea predpovedania, ktorá spočíva v detekcii stavov v minulosti (stavov blízkych aktuálnemu stavu) a následnom spriemerovaní ich nasledujúcich stavov, môže mať obdobnú presnosť ako pomerne zložité ARMA modelovanie a štrukturálny prístup.

Literatúra

- [1] KENNEL, M.B., BROWN, R. a ABARBANEL, H.D.I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*. 1992, **45**, 3403-3411.
- [2] HEGGER, R. a KANTZ, H. Improved false nearest neighbour method to detect determinism in time series data. *Physical Review E*. 1999, **60**.
- [3] BUZUG, Th. and PFISTER, G. Comparison of algorithms calculating optimal parameters for delay time coordinates. *Physica D*. 1992, **58**.
- [4] ROSENSTEIN, M.T., COLLINS, J.J. a DE LUCA, C.J. Reconstruction expansion as a geometry-based framework for choosing proper delay times. *Physica D*. 1994, **73**.
- [5] CASDAGLI, M., EUBANK, S., FARMER, J.D. a GIBSON, J. State space reconstruction in the presence of noise. *Physica D*. 1991, **51**.
- [6] FRASER, A. and SWINNEY, H. Independent coordinates for strange attractors from mutual information. *Physical Review A*. 1986, **33**.
- [7] WHITNEY, H. Differentiable manifolds. *Ann. Math.* 1936, **37**.
- [8] KANTZ, H. a SCHREIBER, T. *Nonlinear time series analysis*. Cambridge University Press, 1997.
- [9] TAKENS, F. Detecting strange attractors in turbulence. In: *Lecture Notes in Mathematics* vol. 898. New York: Springer, 1981.
- [10] SAUER, T., YORKE, J.A a CASDAGLI, M. Embedology. *J. Stat. Phys.* 1991, **65**.
- [11] PROVENZALE, A., SMITH, L.A., VIO, R. a MURANTE, G. Distinguishing between low-dimensional dynamics and randomness in measured time series. *Physica D*. 1992, **58**.

- [12] STROZZI, F. a ZALDIVAR, J.M. Nonlinear forecasting in high-frequency financial time series. *Physica A*. 2005, **353** 463-479.
- [13] OHIRA, T., et al. Predictability of currency market exchange. *Physica A*. 2002, **308**, 368-374.
- [14] www.dukascopy.com/swiss/english/marketwatch/historical/
- [15] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2011, ISBN 3-900051-07-0, <http://www.R-project.org/>
- [16] Di NARZO, A. a Di NARZO, F. tseriesChaos: Analysis of nonlinear time series. R package version 0.1-11., 2010, <http://CRAN.R-project.org/package=tseriesChaos>
- [17] Di NARZO, A. a Di NARZO, F. RTisean: R interface to Tisean algorithms. R package version 3.0.13., 2011, R documentation and examples by Gianluca Gazzola. <http://CRAN.R-project.org/package=RTisean>
- [18] HEGGER, R., KANTZ, H. a SCHREIBER, T. Practical implementation of nonlinear time series methods: The TISEAN package. *CHAOS*. 1999 **9**, 413-. <http://www.mpipks-dresden.mpg.de/tisean>
- [19] KUGIUMTZIS, D. State space reconstruction parameters in the analysis of chaotic time series – the role of the time window length. *Physica D*. 1996, **95**.
- [20] STROZZI, F., ZALDIVAR, J.M. a ZBILUT, J.P. Application of nonlinear time series analysis techniques to high-frequency currency exchange data. *Physica A*. 2002, **312**.
- [21] FARMER, D. a SIDOROWICH, J. Predicting chaotic time series. *Physical Review Letters*. 1987, **59**.
- [22] CASDAGLI, M. Nonlinear prediction of chaotic time series. *Physica D*. 1989, **35**.
- [23] YAKOWITZ, S. Nearest-neighbor methods for time series analysis. *Journal of Time Series Analysis*. 1987, **8**.
- [24] KUNG-SIK CHAN. TSA: Time Series Analysis. R package version 0.98. 2010, <http://CRAN.R-project.org/package=TSA>

- [25] BOX, G.E.P. a JENKINS, G.M. Time series analysis: forecasting and control. San Francisco: Holden Day, 1976.
- [26] CHATFIELD, C. The analysis of time series, an introduction. 6. vyd. New York: Chapman & Hall/CRC, 2004.
- [27] HYNDMAN, R.J. forecast: Forecasting functions for time series. R package version 3.10. 2011, <http://CRAN.R-project.org/package=forecast>
- [28] HYNDMAN, R.J. a KHANDAKAR, Y. Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*. 2008, **27**.
- [29] COMMANDEUR, J.J.F. a KOOPMAN, S.J. *An Introduction to State Space Time Series Analysis*. New York: Oxford University Pres, 2007.
- [30] DURBIN, J. a KOOPMAN, S.J. *Time Series Analysis by State Space Methods*. New York: Oxford University Pres, 2001.
- [31] ALVAREZ-DIAZ, M. Exchange Rates Forecasting: Local or Global Method? *Applied Financial Economics Letters*. 2008, **40**.

Príloha

```
data <- read.table(file="C:\\diplom_data.txt" , header=TRUE)
attach(data)
### uprava dat
AUD_ask <- 1/AUDUSD_ask
AUD_bid <- 1/AUDUSD_bid
EUR_ask <- 1/EURUSD_ask
EUR_bid <- 1/EURUSD_bid
GBP_ask <- 1/GBPUSD_ask
GBP_bid <- 1/GBPUSD_bid
NZD_ask <- 1/NZDUSD_ask
NZD_bid <- 1/NZDUSD_bid
### logaritmicna stredna cena y_m
AUD_m <- (log(AUD_ask)+log(AUD_bid))/2
EUR_m <- (log(EUR_ask)+log(EUR_bid))/2
GBP_m <- (log(GBP_ask)+log(GBP_bid))/2
NZD_m <- (log(NZD_ask)+log(NZD_bid))/2
CAD_m <- (log(USDCAD_ask)+log(USDCAD_bid))/2
CHF_m <- (log(USDCHF_ask)+log(USDCHF_bid))/2
JPY_m <- (log(USDJPY_ask)+log(USDJPY_bid))/2
NOK_m <- (log(USDNOK_ask)+log(USDNOK_bid))/2
SEK_m <- (log(USDSEK_ask)+log(USDSEK_bid))/2
SGD_m <- (log(USDSGD_ask)+log(USDSGD_bid))/2
### normalizovana log. str. cena y
AUD <- (AUD_m - min(AUD_m))/(max(AUD_m) - min(AUD_m))+0.001
EUR <- (EUR_m - min(EUR_m))/(max(EUR_m) - min(EUR_m))+0.001
GBP <- (GBP_m - min(GBP_m))/(max(GBP_m) - min(GBP_m))+0.001
NZD <- (NZD_m - min(NZD_m))/(max(NZD_m) - min(NZD_m))+0.001
CAD <- (CAD_m - min(CAD_m))/(max(CAD_m) - min(CAD_m))+0.001
CHF <- (CHF_m - min(CHF_m))/(max(CHF_m) - min(CHF_m))+0.001
```



```
JPY <- (JPY_m - min(JPY_m))/(max(JPY_m) - min(JPY_m))+0.001
NOK <- (NOK_m - min(NOK_m))/(max(NOK_m) - min(NOK_m))+0.001
SEK <- (SEK_m - min(SEK_m))/(max(SEK_m) - min(SEK_m))+0.001
SGD <- (SGD_m - min(SGD_m))/(max(SGD_m) - min(SGD_m))+0.001
n<-length(AUD) # 8760
library(odesolve)
library(tseriesChaos)
ami_AUD<-mutual(AUD, partitions = 16, lag.max =150, plot=TRUE) # 126
ami_AUD
ami_EUR<-mutual(EUR, partitions = 16, lag.max =150, plot=TRUE) # 127
ami_EUR
ami_GBP<-mutual(GBP, partitions = 16, lag.max =150, plot=TRUE) # 140
ami_GBP
ami_NZD<-mutual(NZD, partitions = 16, lag.max =150, plot=TRUE) # 134
ami_NZD
ami_CAD<-mutual(CAD, partitions = 16, lag.max =150, plot=TRUE) # 127
ami_CAD
ami_CHF<-mutual(CHF, partitions = 16, lag.max =200, plot=TRUE) # 150
ami_CHF
ami_JPY<-mutual(JPY, partitions = 16, lag.max =150, plot=TRUE) # 144
ami_JPY
ami_NOK<-mutual(NOK, partitions = 16, lag.max =150, plot=TRUE) # 106
ami_NOK
ami_SEK<-mutual(SEK, partitions = 16, lag.max =150, plot=TRUE) # 130
ami_SEK
ami_SGD<-mutual(SGD, partitions = 16, lag.max =150, plot=TRUE) # 108
ami_SGD
fn_AUD<-false.nearest(AUD, m=15, d=126, t=0, rt=10, eps=sd(AUD)/10)
plot.false.nearest(fn_AUD)
print.false.nearest(fn_AUD)
fn_EUR<-false.nearest(EUR, m=15, d=127, t=0, rt=10, eps=sd(EUR)/10)
plot.false.nearest(fn_EUR)
print.false.nearest(fn_EUR)
fn_GBP<-false.nearest(GBP, m=15, d=140, t=0, rt=10, eps=sd(GBP)/10)
plot.false.nearest(fn_GBP)
print.false.nearest(fn_GBP)
```

```
fn_NZD<-false.nearest(NZD, m=15, d=134, t=0, rt=10, eps=sd(NZD)/10)
plot.false.nearest(fn_NZD)
print.false.nearest(fn_NZD)
fn_CAD<-false.nearest(CAD, m=15, d=127, t=0, rt=10, eps=sd(CAD)/10)
plot.false.nearest(fn_CAD)
print.false.nearest(fn_CAD)
fn_CHF<-false.nearest(CHF, m=15, d=150, t=0, rt=10, eps=sd(CHF)/10)
plot.false.nearest(fn_CHF)
print.false.nearest(fn_CHF)
fn_JPY<-false.nearest(JPY, m=15, d=144, t=0, rt=10, eps=sd(JPY)/10)
plot.false.nearest(fn_JPY)
print.false.nearest(fn_JPY)
fn_NOK<-false.nearest(NOK, m=15, d=106, t=0, rt=10, eps=sd(NOK)/10)
plot.false.nearest(fn_NOK)
print.false.nearest(fn_NOK)
fn_SEK<-false.nearest(SEK, m=15, d=130, t=0, rt=10, eps=sd(SEK)/10)
plot.false.nearest(fn_SEK)
print.false.nearest(fn_SEK)
fn_SGD<-false.nearest(SGD, m=15, d=108, t=0, rt=10, eps=sd(SGD)/10)
plot.false.nearest(fn_SGD)
print.false.nearest(fn_SGD)

stplot(AUD, m=8, d=126, idt=1, mdt=1000)
stplot(EUR, m=9, d=127, idt=1, mdt=1000)
stplot(GBP, m=12, d=140, idt=1, mdt=1000)
stplot(NZD, m=10, d=134, idt=1, mdt=1000)
stplot(CAD, m=10, d=127, idt=1, mdt=1000)
stplot(CHF, m=9, d=150, idt=1, mdt=1000)
stplot(JPY, m=8, d=144, idt=1, mdt=1000)
stplot(NOK, m=12, d=106, idt=1, mdt=1000)
stplot(SEK, m=10, d=130, idt=1, mdt=1000)
stplot(SGD, m=10, d=108, idt=1, mdt=1000)
#####
### strategia Strozzi
d<-1 # dimenzia vnorenia
detV<-function(tau, d, t, x) # t>tau*d, vypocita V(t)=det(diag_matica)
```

```

{ x[t]-x[t-tau] } # pre konkretne t
Vol<-function(tau,d,x)
{VOLUME<-sapply((tau*d+1):n, function(i) detV(tau, d, i, x) )
  space_volume<-c(rep(0,tau*d),VOLUME)
  return(space_volume)}
dVol<-function(tau, d, x)
{dVOLUME<-sapply((d*tau+1):n, function(i) x[i]-x[i-tau])
  dspace_volume<-c(rep(0,d*tau),dVOLUME)
  return(dspace_volume)}
0<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760)
d0<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760)
for(tau in 2:200) 0[,tau-1]<-Vol(tau,d,AUD) # matica space volume 0,
#kazdy stlpec pre ine tau, pre AUD
for(tau in 2:200) d0[,tau-1]<-dVol(tau,d,0[,tau-1])
# matica zmeny space volume d0 -|-
##### strategia
C1<-matrix(c(rep(0,199*8761)), ncol=199, nrow=8761)
# mnozstvo penazi v USD
C2<-matrix(c(rep(0,199*8761)), ncol=199, nrow=8761)
# mnozstvo penazi v inej mene
for (tau in 2:200) { C1[(d*tau+1),tau-1]<-100}

for (tau in 2:200){
for (i in (d*tau+1):8760)
  {if(d0[i,tau-1]<d0[i-1,tau-1] && C1[i,tau-1]!=0){C1[i+1,tau-1]=0
# C2[i+1,tau-1]=C1[i,tau-1]/AUD[i]}
  if(d0[i,tau-1]<d0[i-1,tau-1] && C1[i,tau-1]==0){C1[i+1,tau-1]=C1[i,tau-1]
# C2[i+1,tau-1]=C2[i,tau-1]}
  if(d0[i,tau-1]>d0[i-1,tau-1] && C1[i,tau-1]!=0){C1[i+1,tau-1]=C1[i,tau-1]
# C2[i+1,tau-1]=C2[i,tau-1]}
  if(d0[i,tau-1]>d0[i-1,tau-1] && C1[i,tau-1]==0){
# C1[i+1,tau-1]=C2[i,tau-1]*AUD[i]
# C2[i+1,tau-1]=0 }
  if(d0[i,tau-1]==d0[i-1,tau-1]) {C1[i+1,tau-1]=C1[i,tau-1]
# C2[i+1,tau-1]=C2[i,tau-1]}

```

```

}}
##### zisk/strata
asset<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760) #### assets
for (tau in 2:200) {for(i in 1:8760){
    if (C1[i,tau-1]!=0) asset[i,tau-1]=C1[i,tau-1]
    else asset[i,tau-1]=C2[i,tau-1]*AUD[i]}}

gain<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760)
for (tau in 2:200) {for (i in (d*tau+1+1):8760){
    gain[i,tau-1]=(asset[i,tau-1]-asset[i-1,tau-1])/asset[i-1,tau-1] }}

TotalGain<-c(rep(0,199))
for (i in 1:199) TotalGain[i]<-sum(gain[,i], na.rm=TRUE)
maxim<-max(TotalGain)
maxim
min(TotalGain)
pocetK<-0 # pocet pripadov ked TotalGain je > 0
for (i in 1:199) { if (TotalGain[i]>0) pocetK<-pocetK+1
    else pocetK<-pocetK }
pocetK
opt_tau<-0
for (i in 2:200) { if (TotalGain[i-1]==maxim) opt_tau<-i}
opt_tau
##### pre inu menu, treba v predchadz. "AUD" vymenit
#####
##### dimenzia >= 2 #####
detVol<-function(tau, d, t, x) # t>tau*d,vypocita V(t)=det(diag_matica)
{zlozka1<-x[t]-x[t-tau] # pre konkretne t
  zlozka2<-sapply(2:d, function(i) x[t-((i-1)*tau)]-x[t-(i*tau)])
  vektor<-c(zlozka1,zlozka2)
  prod(vektor)}
#####
Vol<-function(tau,d,x)
{VOLUME<-sapply((tau*d+1):n, function(i) detVol(tau, d, i, x) )
  space_volume<-c(rep(0,tau*d),VOLUME)
  return(space_volume)}

```

```
#####
dVol<-function(tau, d, x)
{dVOLUME<-sapply((d*tau+1):n, function(i) x[i]-x[i-tau] )
  dspace_volume<-c(rep(0,d*tau),dVOLUME)
  return(dspace_volume)}
#####
d<-2      # dimenzia vnorenia
0<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760)  #
d0<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760)
for(tau in 2:200) 0[,tau-1]<-Vol(tau,d,AUD) # matica space volume 0,
                                     #kazdy stlpec pre ine tau
for(tau in 2:200) d0[,tau-1]<-dVol(tau,d,0[,tau-1])
                                     # matica zmeny space volume d0 -||-
##### strategia
C1<-matrix(c(rep(0,199*8761)),ncol=199,nrow=8761)# mnozstvo penazi v USD
C2<-matrix(c(rep(0,199*8761)),ncol=199,nrow=8761)# mnoz.penazi v inej mene
for (tau in 2:200) { C1[(d*tau+1),tau-1]<-100}

for (tau in 2:200){
for (i in (d*tau+1):8760)
  {if(d0[i,tau-1]<d0[i-1,tau-1] && C1[i,tau-1]!=0){C1[i+1,tau-1]=0
                                     C2[i+1,tau-1]=C1[i,tau-1]/AUD[i]}
  if(d0[i,tau-1]<d0[i-1,tau-1] && C1[i,tau-1]==0){C1[i+1,tau-1]=C1[i,tau-1]
                                     C2[i+1,tau-1]=C2[i,tau-1]}
  if(d0[i,tau-1]>d0[i-1,tau-1] && C1[i,tau-1]!=0){C1[i+1,tau-1]=C1[i,tau-1]
                                     C2[i+1,tau-1]=C2[i,tau-1]}
  if(d0[i,tau-1]>d0[i-1,tau-1] && C1[i,tau-1]==0)
                                     {C1[i+1,tau-1]=C2[i,tau-1]*AUD[i]
                                     C2[i+1,tau-1]=0 }
  if(d0[i,tau-1]==d0[i-1,tau-1])      {C1[i+1,tau-1]=C1[i,tau-1]
                                     C2[i+1,tau-1]=C2[i,tau-1]}
}}

##### zisk/strata
asset<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760) ##### assets
for (tau in 2:200) {for(i in 1:8760){
```

```
        if (C1[i,tau-1]!=0) asset[i,tau-1]=C1[i,tau-1]
        else asset[i,tau-1]=C2[i,tau-1]*AUD[i]}}
gain<-matrix(c(rep(0,199*8760)), ncol=199, nrow=8760)
for (tau in 2:200) {for (i in (d*tau+1+1):8760){
    gain[i,tau-1]=(asset[i,tau-1]-asset[i-1,tau-1])/asset[i-1,tau-1]}}
TotalGain<-c(rep(0,199))
for (i in 1:199) TotalGain[i]<-sum(gain[,i], na.rm=TRUE)
maxim<-max(TotalGain)
maxim
min(TotalGain)
pocetK<-0 # pocet pripadov ked TotalGain je > 0
for (i in 1:199) { if (TotalGain[i]>0) pocetK<-pocetK+1
    else pocetK<-pocetK }
pocetK
opt_tau<-0
for (i in 2:200) { if (TotalGain[i-1]==maxim) opt_tau<-i}
opt_tau
##### pre ine d, treba zmenit d
###RANDOM WALKS
N<-8760
index<-c(1:8760)
n<-8760
set.seed(10)
u1<-rnorm(N)
y1<-cumsum(u1)+40
set.seed(12)
u2<-rnorm(N)
y2<-cumsum(u2)+80
set.seed(15)
u3<-rnorm(N)
y3<-cumsum(u3)+80
set.seed(17)
u4<-rnorm(N)
y4<-cumsum(u4)+100
set.seed(21)
u5<-rnorm(N)
```

```
y5<-cumsum(u5)+50
set.seed(26)
u6<-rnorm(N)
y6<-cumsum(u6)+100
set.seed(28)
u7<-rnorm(N)
y7<-cumsum(u7)+100
set.seed(30)
u8<-rnorm(N)
y8<-cumsum(u8)+70
set.seed(54)
u9<-rnorm(N)
y9<-cumsum(u9)+70
set.seed(61)
u10<-rnorm(N)
y10<-cumsum(u10)+120
y1_m <- log(y1)
r1 <- (y1_m - min(y1_m))/(max(y1_m) - min(y1_m))+0.001
y2_m <- log(y2)
r2 <- (y2_m - min(y2_m))/(max(y2_m) - min(y2_m))+0.001
y3_m <- log(y3)
r3 <- (y3_m - min(y3_m))/(max(y3_m) - min(y3_m))+0.001
y4_m <- log(y4)
r4 <- (y4_m - min(y4_m))/(max(y4_m) - min(y4_m))+0.001
y5_m <- log(y5)
r5 <- (y5_m - min(y5_m))/(max(y5_m) - min(y5_m))+0.001
y6_m <- log(y6)
r6 <- (y6_m - min(y6_m))/(max(y6_m) - min(y6_m))+0.001
y7_m <- log(y7)
r7 <- (y7_m - min(y7_m))/(max(y7_m) - min(y7_m))+0.001
y8_m <- log(y8)
r8 <- (y8_m - min(y8_m))/(max(y8_m) - min(y8_m))+0.001
y9_m <- log(y9)
r9 <- (y9_m - min(y9_m))/(max(y9_m) - min(y9_m))+0.001
y10_m <- log(y10)
r10 <- (y10_m - min(y10_m))/(max(y10_m) - min(y10_m))+0.001
```

```
#####  
# na nah. prechádzkach r1, r2, ..., r10 sme použili tu istu stratégiu  
# ako na vym.kurzoch AUD, EUR, ..., SGD  
##### NN metoda #####  
library(TSA)  
library(fracdiff)  
library(forecast)  
###data a ich uprava  
data(beersales) # 192 udajov  
X<-stl(beersales,"periodic") # stl rozklad:seasonal, trend, remainder  
Y<-X$time.series  
piva<-Y[,1]+Y[,3]  
train<-ts(piva, start = 1, end = 156)  
### najdenie opt. parametrov vnorenia  
mut<-mutual(train, partitions = 16, lag.max = 20, plot=TRUE) # tau=2  
print(mut)  
ff<-false.nearest(train, m=10, d=2,t=0) # dimenzia m=4  
print(ff)  
plot(ff)  
#funkcia embedovanie robi to iste ako embedSeries  
#s tym rozdielom, za embedding vektory su inak cislovane,  
#pri embedSeries: 1,2,3,...,n-tau  
#pri embedovanie: tau+1, tau+2,...,n  
#teda vracia maticu, ktorej riadky su embed.vektory  
embedovanie<-function(x, m, tau, series.name=NULL){  
  if (is.null(series.name))  
    series.name <- deparseText(substitute(x))  
  N<-length(x)  
  if(N<=(m-1)*tau) return("chyba, malo pozorovani")  
  embed.matrix<-matrix(c(rep(0,m*(N-(m-1)*tau))),ncol=m,nrow=N-(m-1)*tau)  
  for(i in 1:(N-(m-1)*tau)){  
    for(j in 1:m){embed.matrix[i,j]<-x[i+(j-1)*tau]}}  
  dim.names <- character(m)  
  for (i in seq(length = m)) {  
    dim.names[i]<-switch(as.integer(i>m-1)+1,paste(series.name,  
      "[t-",(m-i)*tau,"]",sep = ""),
```



```

        paste(series.name, "[t]", sep = ""))}
dimnames(embed.matrix)<-list(paste("t=",seq((m-1)*tau+1,N,by=1),sep=""),
        dim.names)

return(embed.matrix)}
###
# funkcia knnForecast vypocitava predpoved na 1 krok dopredu pomocou
# k najblizsich susedov a ich arit. priemeru
# x...vektor dat
# m...embed.dimenzia
# tau...lag/time delay
# k...pocet hladanych susedov
# norm...metric/norma
# F...pocet krokov predpovede
knnForecast<-function(x, m, tau, k, norm, F)
{l<-length(x)
  if(l<=(m-1)*tau) return("chyba, zla dimenzia/lag")
  for (j in 1:F){
    x_embed<-embedovanie(x, m, tau, series.name=NULL)
    susedia<-findNeighbors(x_embed,n.neighbor=k,metric=norm,max.distance =0,
        olag=1, sort.distances=TRUE)
    N<-length(susedia$original)
    vektor<-NULL
    for (i in 1:k){ vektor<-c(vektor,susedia$neighbor[N-(k-i)])}
    sucet<-0
    for (i in 1:k) sucet<-sucet+x_embed[1+vektor[i],m]
    priemer<-sucet/k
    x<-c(x,priemer)}
  return(x)}
### funkcia wknnForecast vypocita predpoved na 1 krok dopredu
### pomocou k najblizsich susedov a ich vazeneho priemeru
### blizsi sused->vyssie vahy
# x...vektor dat
# m...embed.dimenzia
# tau...time delay
# k...pocet hladanych najblizsich susedov
# norm...norma...metric v "findNeighbor"...1,2,Inf

```

```

# F...kolko krokov dopredu sa rata predpoved
# method...norma...method v "dist"
wknnForecast<-function(x, m, tau, k, norm, F, method)
{l<-length(x)
  if(l<=(m-1)*tau) return("chyba")
  for (j in 1:F){
    E<-embedovanie(x, m, tau, series.name=NULL)
    R<-nrow(E)
    susedia<-findNeighbors(E, n.neighbor=k, metric=norm, max.distance = 0,
                          olag=1, sort.distances=TRUE)
    N<-length(susedia$original)
    vektor<-NULL
    for (i in 1:k){ vektor<-c(vektor,susedia$neighbor[N-(k-i)])}
    D<-E[R,]
    for(i in 1:k){ D<-rbind(D,E[vektor[i],])}
    d<-dist(D,method = method)
    M<-as.matrix(d)
    W<-sum(M[,1])
    w<-NULL
    for(i in 1:k){w<-c(w,M[i+1,1])}
    w<-w[k:1]/W
    P<-NULL
    for(i in 1:k){P<-rbind(P,E[vektor[i]+1,])}
    predpoved<-sum(w*P[,m])
    x<-c(x,predpoved)}
  return(x)}

frc1<-knnForecast(train,4,2,6,2,36) # k=6
frc1_ts<-ts(frc1[157:192],start=c(1988,1),frequency=12)
povodne<-ts(piva[157:192],start=c(1988,1),frequency=12)
A1<-cbind(povodne, frc1_ts)
plot(A1,plot.type="single", col=c("black","red"),lwd=2, ylab="")
rmse1<-sqrt(mean((frc1_ts-povodne)^2))
rmse1 # 0.7131374
##### ARMA model #####
# na najdenie opt. ARIMA modelu pouzijeme finkciu auto.arima
train<-ts(piva, start = 1, end = 156, frequency=12)

```

```
fpiva<-auto.arima(train,d=0,D=0,max.p=5,max.q=5,max.P=5,max.Q=5,max.order=6,
start.p=1, start.q=1, start.P=1, start.Q=1,
stationary=TRUE, ic="aic", stepwise=TRUE, trace=FALSE,
approximation=(length(train)>100 | frequency(train)>12), xreg=NULL,
test="adf", seasonal.test="ch",
allowdrift=TRUE, lambda=NULL)
fpiva # ARIMA(5,0,5)(1,0,0)[12], 2359.66
ar_korene<-polyroot(c(1, 0.4021, -1.0569, -0.2833, 0.9310, 0.3812))
abs(ar_korene) # ok, vsetky su >1 => stacionarita
ma_korene<-polyroot(c(1, 0.1636, -1.4134, -0.2556, 0.6127,-0.0416))
abs(ma_korene) # ok, vsetky su >1 => invertovatelnost
arma_f<-forecast(fpiva, h=36)
arma_f_ts<-ts(arma_f$mean,start=c(1988,1), frequency=12 )
povodne<-ts(piva[157:192], start=c(1988,1), frequency=12)
B<-cbind(povodne,arma_f_ts)
plot(B ,plot.type="single", col=c("black","red"), lwd=2, ylab="")
rmse_arma<-sqrt(mean((arma_f_ts-povodne)^2))
rmse_arma # 1.205982
##### Strukturalny pristup #####
require(graphics)
train1<-ts(beersales[1:156], frequency=12, start=c(1975,1))
fit<-StructTS(train1, type = "BSM", init=c(0.9, 0.1, 0.7, 0.8))
fit #level      slope      seas      epsilon
#0.2602949  0.0001209  0.2284347  0.0000000
Struct_forecast<-forecast(fit, h=36) # predikcia
# kreslenie obrazkov, povodnych dat a predikcie
Struct_forecast_ts<-ts(Struct_forecast$mean,start=c(1988,1),frequency=12)
povodne<-ts(beersales[157:192], start=c(1988,1), frequency=12)
B<-cbind(povodne,Struct_forecast_ts)
plot(B ,plot.type="single", col=c("black","red"), lwd=2, ylab="")
rmse_Struct<-sqrt(mean((Struct_forecast_ts-povodne)^2))
rmse_Struct # 0.6393829
```