MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Jiří Gregor

Holtova-Wintersova metoda s chybějícími pozorováními a její aktuárské aplikace

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: prof. RNDr. Tomáš Cipra, DrSc.
Studijní program: Matematika
Studijní obor: Finanční matematika

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V .... dne ....

Podpis autora

Rád bych poděkoval prof. RNDr. Tomáši Ciprovi, DrSc. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce.

Název práce: Holtova-Wintersova metoda s chybějícími pozorováními a její aktuárské aplikace

Autor: Jiří Gregor

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: prof. RNDr. Tomáš Cipra, DrSc., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Práce stručně popisuje problematiku časových řad a sezónnosti. Představuje různé přístupy k vyrovnání a predikci časových řad. Hlavní část práce je věnována Holtově - Wintersově metodě pro časové řady s chybějícími pozorováními a její následná aplikace na aktuárská data.

Klíčová slova: časové řada, vyrovnávání, Holtova - Wintersova metoda s chybějícími pozorováními

Title: Holt-Winters method with missing observations and its actuarial application

Author: Jiří Gregor

Department: Department of Probability and Mathematical Statistics

Supervisor: prof. RNDr. Tomáš Cipra, DrSc., Department of Probability and Mathematical Statistics

Abstract: This thesis describes problematics of time series and its seasonal component. It shows different approaches to smoothing and prediction of time series. The main part of thesis is devoted to Holt-Winters method with missing observations and its actuarial application.

Keywords: time serie, smoothing, Holt-Winters method with missing observations

Obsah

Úvod

V ekonomii a v dalších vědních oborech se často setkáváme s daty ve formě časové řady. Jedná se o posloupnost pozorování dané veličiny v určité frekvenci - např. měsíční. Příkladem časové řady může být počet pojistných událostí nahlášených pojišťovně v jednotlivých měsících.

Tato práce se zabývá extrapolací a vyhlazováním časové řady, kde vyhlazováním rozumíme potlačení náhodných fluktuací. Existuje mnoho způsobů, jak k danému problému přistupovat, například pomocí jednoduché metody eliminace trendu. Dalšími, více sofistikovanějšími možnostmi, jsou metody, kdy se snažíme popsat trend pomocí jednoduchých křivek. Při popisu klasických metod v tomto kontextu práce čerpala především z monografie Cipra (2013) a navazujících publikací, viz např. Koritarova (2014). Metodou, kterou se v této práci budeme více zabývat, je Holtova-Wintersova metoda aplikovaná na data s chybějícími pozorováními.

Řešení problému extrapolace a vyhlazování časové řady má mnoho aplikací jako je například předpověď vývoje ceny podkladového aktiva nebo lze pomocí sezónní složky časové řady zkoumat průběh poptávky po zboží a mnohé další.

1. Dekompozice jednorozměrných časových řad

Časové řady můžeme rozložit na několik složek, jimiž jsou: trend Tr_t, sezónní složka Sz_t, cyklická složka C_t a reziduální složka E_t. Klasická dekompozice pohlíží na trendovou, sezónní a cyklickou složku jako na deterministické funkce času, ale na reziduální složku jako náhodnou funkci času.

1.1 Trend

Zachycuje dlouhodobé změny v průměrné úrovni časové řady. Trendová složka je pojem do jisté míry vágní, např.: změny klimatu mohou být vnímány ekonomem jako trend dlouhodobý, naproti tomu takový klimatolog je může vnímat jen jako krátkodobý pohyb.

1.2 Sezónní složka

Popisuje periodické změny v časové řadě, které se odehrávají během jednoho období (např. roku) a každé období se opakují, například v lednu je větší spotřeba paliva.

1.3 Cyklická složka

Jedná se o fluktuace kolem trendu, v nichž se střídají fáze růstu s fázemi poklesu.

1.4 Reziduální složka

Reziduální složku časové řady dostaneme po odstranění cyklické, sezónní a trendové složky a je tvořena náhodnými pohyby v průběhu časové řady. Tyto náhodné pohyby nemají systematický charakter. Do reziduální složky řadíme i chyby v měření.

Předpokládáme, že reziduální složka je tzv. bílý šum. Bílý šum definujeme jako posloupnost $\{\epsilon_t\}$, pro kterou platí:

$$E(\epsilon_t) = 0, Var(\epsilon_t) > 0, cov(\epsilon_t, \epsilon_s) = 0 \text{ pro } s \neq t.$$

1.5 Aditivní a multiplikativní dekompozice

Dekompozici časové řady uvažujeme dvojí, aditivní nebo multiplikativní. Aditivní dekompozice časové řady je ve tvaru:

$$y_t = Tr_t + C_t + Sz_t + E_t,$$

kde jednotlivé složky rozkladu uvažujeme ve skutečných absolutních hodnotách a jsou měřeny v jednotkách řady y_t.

Multiplikativní dekompozice časové řady má tvar:

$$y_t = Tr_t \cdot C_t \cdot Sz_t \cdot E_t.$$

Při použití multiplikativního rozkladu je pouze trendová složka uvažována ve své absolutní hodnotě, a tedy měřena v jednotkách časové řady y_t. Ostatní složky jsou uvažovány v relativních hodnotách vůči trendu, a jsou tedy bezrozměrné. Například $Sz_1 = 1.20$ vyjadřuje, že hodnota řady, kterou zachycuje trendová a sezónní složka, je v prvním období 1.20 násobek Tr_1.

1.6 Vyrovnaná časová řada a předpověď v časové řadě

Předpokládejme pozorování y_1,...,y_n, pak klademe $\hat{y}_t = \hat{Tr}_t + \hat{C}_t + \hat{Sz}_t$ pro aditivní metodu, resp $\hat{y}_t = \hat{Tr}_t \cdot \hat{C}_t \cdot \hat{Sz}_t$ pro multiplikativní metodu, kde $\hat{Tr}_t$ je eliminovaná hodnota trendové složky v čase t a $\hat{Sz}_t$ je eliminovaná hodnota sezónní složky v čase t. Pokud t náleží $\{1,..,n\}$, pak se jedná o vyrovnání (vyhlazení) časové řady. Pokud $t > n$, pak se jedná o předpověď časové řady.

1.7 Metoda klouzavých průměrů

Jedná se o metodu, která patří mezi tzv. adaptivní přístupy k trendové složce, kde adaptivním přístupem obecně rozumíme přístup, který je schopen pracovat se systematickými složkami měnícími v průběhu času globální charakter. Nicméně předpokládáme, že v krátkých úsecích je možné vyrovnání pomocí nějaké matematické křivky. Konkrétně klouzavé průměry definujeme jako lineární kombinaci členů původní časové řady tak, že součet jejich vah se rovná 1. Tedy například:

$$1/4 \cdot (3y_{t-1} + y_{t-2}).$$

Konstrukce klouzavých průměrů vyrovnáním úseků řady polynomickými křivkami

Tato metoda předpokládá, že každá „rozumně se chovající" funkce jde aproximovat pomocí polynomu. Postupujeme následujícím způsobem:

1) Vyrovnáme vhodným polynomem $2k + 1$ prvních členů řady.

2) Použijeme hodnotu polynomu v bodě $k + 1$ jako vyrovnanou hodnotu $\hat{y}_{k+1}$ dané řady v tomto čase.

3) Pro získání vyrovnané hodnoty v bodě $t = k + 2$ opakujeme celý postup na pozorování o jedna posunuté, tedy $y_2, \ldots .., y_{2k+2}$. Koeficienty vyrovnávajícího polynomu odhadujeme metodou nejmenších čtverců.

1.8 Exponenciální vyrovnání

Pro celou kapitolu 1.8 budeme předpokládat, že řada je tvořena pouze trendem a aditivní reziduální složkou. Vyrovnaná hodnota je pro exponenciální vyrovnání zvláštním případem klouzavého průměru, kdy všechny dosud pozorované hodnoty vyrovnané řady vážíme do minulosti exponenciálními vahami. Vyrovnaná řada $\hat{y}_t$ tedy je minimalizací výrazu:

$$(y_t - \hat{y}_t)^2 + (y_{t-1} - \hat{y}_{t-1})^2 \cdot \beta + (y_{t-2} - \hat{y}_{t-2})^2 \cdot \beta^2 + \ldots,$$

kde β $(0 < \beta < 1)$ je předem zvolená diskontní konstanta.

Jednoduché exponenciální vyrovnání

Předpokládejme, že řada má lokálně konstantní trend $Tr_t = \beta_0$. Odhadněme β_0. Nechť $b_0(t)$ je odhad parametru β_0 provedený v čase t. Potom odhad $b_0(t)$ je jak odhadem trendu v čase t, ale i zároveň vyrovnanou hodnotou $\hat{y}_t$ uvažované řady. Tento odhad dostaneme minimalizací následujícího výrazu:

$$\sum_{j=0}^{\infty} (y_{t-j} - \beta_0)^2 \cdot \beta^j, \tag{1.8.1}$$

kde $\beta(0 < \beta < 1)$ je předem zvolená diskontní konstanta. Výraz (1.8.1) je ve tvaru nekonečného součtu, navzdory skutečnosti, že v praktických příkladech známe pouze konečný počet y_1,...,y_n. Toto prodloužení řady do minulosti ovšem zjednodušuje následný výpočet, neboť odpovídá limitním přechodům. Položme derivaci (1.8.1) podle β_0 rovnou nule, pak díky konvexitě minimalizované funkce dostáváme odhad $b_0(t)$ parametru β_0 v čase t ve tvaru:

$$\hat{y}_t = (1 - \beta) \cdot \sum_{j=0}^{\infty} \beta^j \cdot y_{t-j}. \tag{1.8.2}$$

Tedy vyrovnanou hodnotu časové řady v čase t dostaneme jako vážený průměr hodnot této řady do času t převáženými exponenciálně klesajícími váhami

$$1 - \beta, (1 - \beta) \cdot \beta, (1 - \beta) \cdot \beta^2, (1 - \beta) \cdot \beta^3, \ldots \quad .$$

Kde výraz (1.8.2) odpovídá rekurentnímu tvaru:

$$\hat{y}_t = \alpha \cdot y_t + (1 - \alpha) \cdot \hat{y}_{t-1}, \tag{1.8.3}$$

kde definujeme alfu jako: $\alpha = 1 - \beta(0 < \alpha < 1)$ a nazveme ji vyrovnávací konstanta. Díky tomu, že exponenciální vyrovnání funguje na základě vzorce (1.8.3), vidíme, že nepotřebujeme uchovávat velké množství dat, stačí nám pouze minulá vyrovnaná hodnota.

1.9 Dvojité exponenciální vyrovnání

Pro sekci (1.9) předpokládáme, že časová řada má trend takový, že je lokálně lineární, tedy:

$$Tr_{t-j} = \beta_0 + \beta_1 \cdot (-j).$$

Minimalizací výrazu

$$\sum_{j=0}^{\infty} [y_{t-j} - (\beta_0 + \beta_1 \cdot (-j))]^2 \cdot \beta^j, \tag{1.9.1}$$

kde β $(0 < \beta < 1)$ je předem zvolená diskontní konstanta, dostaneme odhady parametrů β_0 a β_1 v čase t, jež označujeme jako $b_0(t)$ a $b_1(t)$. Parciálně zderivujeme výraz (1.9.1) podle β_0 a β_1 a položíme jej roven 0:

$$\begin{aligned} \sum_{j=0}^{\infty} \beta^j \cdot y_{t-j} - \beta_0 \cdot \sum_{j=0}^{\infty} \beta^j + \beta_1 \cdot \sum_{j=0}^{\infty} j \cdot \beta^j &= 0 \\ \sum_{j=0}^{\infty} j \cdot \beta^j \cdot y_{t-j} - \beta_0 \cdot \sum_{j=0}^{\infty} j \cdot \beta^j + \beta_1 \cdot \sum_{j=0}^{\infty} j^2 \cdot \beta^j &= 0 \end{aligned} \tag{1.9.2}$$

Pro zjednodušení zápisu zavedeme tzv. *jednoduchou vyrovnávací statistiku*, jejíž tvar plyne z výše uvedené soustavy a z (1.8.2) a (1.8.3):

$$S_t = \alpha \cdot y_t + (1 - \alpha) \cdot S_{t-1}.$$

Podobně zavedeme tzv. *dvojitou vyrovnávací statistiku*:

$$S_t^{[2]} = (1 - \beta) \cdot \textstyle\sum_{j=0}^{\infty} \beta^j \cdot S_{t-j}.$$

A po úpravách dostáváme vzorec pro předpověď hodnoty $y_{t+\tau}$, jež je provedená v čase t:

$$\hat{y}_{t+\tau}(t) = (2 + \tfrac{\alpha \cdot \tau}{1-\alpha}) \cdot S_t - (1 + \tfrac{\alpha \cdot \tau}{1-\alpha}) \cdot S_t^{[2]}.$$

Speciálně nechť $\tau = 0$, pak dostáváme vyrovnanou hodnotu uvažované řady ve tvaru:

$$\hat{y}_t = 2S_t - S_t^{[2]}.$$

1.10 Holtova metoda

Holtovu metodu je vhodné používat na časové řady, ve kterých se nevyskytuje sezónní složka. Navíc předpokládáme, že trendová složka je lokálně odhadnutelná lineární funkcí, jejíž směrnice se může v čase měnit. Holtova metoda je zobecněním dvojitého exponenciálního vyrovnání, navzdory tomu, že tato metoda byla nejdříve navržena jako samostatný postup. V Holtově metodě uplatňujeme dvě vyrovnávací konstanty $0 < \alpha,\gamma < 1$, kde α využíváme pro vyrovnání úrovně L_t a γ pro vyrovnání směrnice T_t téže řady:

$$L_t = \alpha \cdot y_t + (1-\alpha)(L_{t-1} + T_{t-1}).$$

Konvexní kombinací poslední pozorované hodnoty v čase t a odhadu této hodnoty v čase $t-1$ dostáváme vyhlazení úrovně trendu:

$$T_t = \gamma \cdot (L_t - L_{t-1}) + (1-\gamma) \cdot T_{t-1}.$$

Pro vyrovnávání časové řady platí:

$$\hat{y}_t = L_t.$$

Pro predikování časové řady platí:

$$\hat{y}_{t+\tau}(t) = L_t + T_t \cdot \tau, (\tau > 0).$$

Aby byl postup funkční je potřeba znát iniciační hodnoty. Takovými vhodnými počátečními hodnotami jsou například:

$$L_0 = y_1 \text{ a } T_0 = y_2 - y_1.$$

Jak bylo zmíněno v úvodu částí (1.10), dvojité exponenciální vyrovnání s vyrovnávací konstantou alfa je speciálním případem Holtovy metody s vyrovnávacími konstantami α_{Holt} a γ_{Holt} ve tvaru:

$$\alpha_{Holt} = \alpha \cdot (2-\alpha), \gamma_{Holt} = \tfrac{\alpha}{2-\alpha}.$$

2. Holtova-Wintersova metoda

2.1 Sezónnost v časové řadě

Pro celou práci předpokládáme, že sezónnost je pravidelná, jinak by náš přístup nedával smysl. Holtova–Wintersova metoda se mimo jiné také zabývá eliminací sezónní složky (1.2). Pro analýzu sezónnosti separujeme tzv. sezónní faktory $I_1, I_2, \ldots, I_s$, kde s je délka sezóny, tedy $s = 12$ odpovídá měsíčním pozorováním. Sezónní faktory modelují sezónní složku Sz_t, proto je potřebujeme pro následné sezónní očištění.

Jednotky, ve kterých měříme sezónní faktory, závisí na tom, zda je dekompozice aditivní, anebo multiplikativní. Navíc platí, že trendová a sezónní složka nejsou navzájem určeny jednoznačně, navýšením jedné dochází ke snížené druhé. Proto zavádíme tzv. normalizační pravidlo pro centrování sezónních faktorů, toto pravidlo ovšem také záleží na způsobu dekompozice.

2.1.1 Sezónnost a aditivní dekompozice

Mějme $y_t = Tr_t + Sz_t + E_t$. Potom se I_t měří ve stejných jednotkách jako příslušná časová řada y_t, tedy pokud 6. sezónní faktor I_6 prodeje paliva benzinové pumpy je velikosti 20 000 Kč, tak pro šesté období se sezónnost projevuje nárůstem prodeje o 20 000 Kč nad celoročním průměrem.

Normalizační pravidlo při aditivní dekompozici je následovné: Nechť k je počet sezón a $i \geq 0$. Potom, aby normalizační pravidlo platilo, musí být splněno:

$$I_{(1+12i)} + I_{(2+12i)} + \ldots . + I_{(12+12i)} = 0.$$

2.1.2 Sezónnost a multiplikativní dekompozice

Mějme $y_t = Tr_t \cdot Sz_t \cdot E_t$. Potom je I_t bezrozměrná veličina, tedy pokud 6. sezónní faktor I_6 je 0,9 prodeje paliva benzinové pumpy, tak to znamená, že se sezónnost projevuje v šestém období poklesem o 10% pod celoročním průměrem.

Normalizační pravidlo pro multiplikativní dekompozici je následovné: Nechť k je počet sezón a $i \geq 0$. Potom, aby normalizační pravidlo platilo, musí být splněno:

$$I_{(1+12i)} \cdot I_{(2+12i)} \cdot \ldots . \cdot I_{(12+12i)} = 1.$$

2.2 Holtova – Wintersova metoda

Tato metoda je rozšířením Holtovy metody (1.10), protože dokáže „zachytit” nejen trend, ale i sezónnost dané časové řady. Na rozdíl od (1.10) používáme 3 vyrovnávací konstanty:

α – pro úroveň $L_t, 0 < \alpha < 1$,

γ – pro směrnici trendu $T_t, 0 < \gamma < 1$,

δ – pro sezónní složku $Sz_t, 0 < \delta < 1$.

Obdobně jako u (1.10) existuje aditivní i multiplikativní verze, kde aditivní verze je více vhodná pro časové řady s neměnnou sezónností a multiplikativní je lepší pro časové řady, u které se sezónní složka mění v závislosti na úrovni dané řady.

2.2.1 Aditivní Holtova–Wintersova metoda

Předpokládejme, že trend je lokálně lineární, pak popíšeme daný model pomocí rekurentních vzorců. Nechť s je počet pozorování v jedné sezóně, např.: pro $s = 12$ máme k dispozici pozorování s měsíční frekvencí.
Pro úroveň řady L_t platí:

$$L_t = \alpha(y_t - Sz_{t-s}) + (1-\alpha)(L_{t-1} + T_{t-1}),$$

kde $Sz_{(t-s)}$ je odhad sezónní složky z minulé sezóny, $L_{(t-1)}$ je odhad úrovně v čase $t-1, T_{(t-1)}$ je odhad trendu v čase $t-1$ a y_t je aktuální pozorovaná hodnota.

Pro trend T_t platí:

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1},$$

kde, $L_t - L_{(t-1)}$ je přibližný odhad směrnice trendu v čase t a $T_{(t-1)}$ je odhad trendu v období $t-1$.
Pro sezónní složku platí:

$$Sz_t = \delta \cdot (y_t - L_t) + (1-\delta) \cdot Sz_{t-s},$$

kde $y_t - L_t$ je odhad aktuální sezónní složky a $Sz_{(t-s)}$ je odhad sezónní složky v čase $t-s$, tedy pokud uvažujeme, že $s = 12$ a $t = 18$, tak jde o sezónní složku pozorování z června minulého roku.
Pro vyrovnanou hodnotu platí:

$$\hat{y}_t = L_t + Sz_t,$$

tedy je rovna součtu sezónní složky a úrovně v čase t.
Pro předpověď budoucího období v čase $t+\tau$, kde $\tau \in \{1,...,s\}$ platí:

$$\hat{y}_{t+\tau} = L_t + T_t \cdot \tau + Sz_{t+\tau-s}.$$

Pro předpovědi časově vzdálenější, v čase $t+\tau$, kde $\tau \in \{s+1,...,2s\}$ platí:

$$\hat{y}_{t+\tau} = L_t + T_t \cdot \tau + Sz_{t+\tau-2s},$$

podobně pro $\tau \geq 2s$.

Inicializace aditivní verze Holtovy – Wintersovy metody

Volba doporučovaná v literatuře je následující:

$$\alpha = \delta = 0.4 \quad \text{a} \quad \gamma = 0.1. \tag{2.1}$$

Dále musíme zvolit počáteční hodnoty L_0,T_0,Sz_{-s+1},...,Sz_0, použijeme regresní přístup k sezónnosti. Na základě odhadů regresního modelu b_0,b_1,a_2,a_3,...,a_s dostáváme následující inicializaci:

$$L_0 = b_0, \quad T_0 = b_1, \quad Sz_{-s+1} = 0, \quad Sz_{-s+2} = a_2, ..., Sz_0 = a_s. \tag{2.2}$$

2.2.2 Multiplikativní Holtova-Wintersova metoda

Vztahy pro multiplikativní verzi metody jsou podobné vztahům z podkapitoly 2.2.1, ale rozdíly a součty jsou nahrazeny podíly a součiny. Dostáváme následující vzorce:

$$L_t = \alpha \cdot (y_t/Sz_{t-s}) + (1-\alpha)(L_{t-1} + T_{t-1}),$$

$$T_t = \gamma \cdot (L_t - L_{t-1}) + (1-\gamma) \cdot T_{t-1},$$

$$Sz_t = \delta \cdot (y_t/L_t) + (1-\delta) \cdot Sz_{t-s},$$

$$\hat{y}_t = L_t \cdot Sz_t,$$

$$\hat{y}_{t+\tau} = (L_t + T_t \cdot \tau) \cdot Sz_{t+\tau-s} \text{ pro } \tau = 1,...,s,$$

$$\hat{y}_{t+\tau} = (L_t + T_t \cdot \tau) \cdot Sz_{t+\tau-2s} \text{ pro } \tau = s+1,...,2s.$$

A podobně pro více časově vzdálené odhady, nicméně je nutné podotknout, že čím vzdálenější odhady jsou, tím méně se stávají přesnými.

Inicializace multiplikativní verze Holtovy – Wintersovy metody

Pro volbu vyrovnávacích konstant můžeme použít volbu (2.1). Pro volbu $L_0, T_0, Sz_{-s+1}, \dots, Sz_0$ nejde použít regresní přístup k sezónnosti. Můžeme ovšem použít následující vzorce:

$$T_0 = \frac{\bar{y}_m - \bar{y}_1}{(m-1)s}, \quad L_0 = \bar{y_1} - \frac{s+1}{2} T_0, \quad Sz_{j-s} = \frac{1}{m} \sum_{i=0}^{m-1} \frac{y_{j+s \cdot i}}{\bar{y}_{i+1} - (\frac{s+1}{2} - j) T_0}, j = 1,...,s, \tag{2.3}$$

kde $\bar{y}_i$ je aritmetický průměr pozorování přes *i-tou* sezónu a m je celkový počet těchto sezón.

3. Holtova-Wintersova metoda s chybějícími pozorováními

Holtova-Wintersova metoda s chybějícím pozorováním je rozšířením Holtovy-Wintersovy metody z 2. kapitoly. Speciálně umožňuje analyzovat časové řady, které mají sezónní složku nenulovou a pro některé časy t neznáme hodnoty této časové řady. Toto rozšíření je praktické, protože v praxi se objevují problémy, kde neznáme všechna pozorování pro dané časy. V této části práce používáme výsledky podle Cipra, Trujillo a Rubio (1995).

Mějme pozorování y_{t_1},y_{t_2},...,y_{t_n} v nepravidelných časech $t_1 < t_2 < ... < t_n$. Nechť p je délka sezóny. Dále předpokládejme, že pro každé z p období existuje alespoň jedno pozorování. Tedy pokud $p = 12$, předpokládáme, že máme informaci alespoň o jednom lednovém pozorování, únorovém pozorování,...,prosincovém pozorování. Navíc předpokládejme o těchto pozorováních, že se objevují „dostatečně brzy“. Například při $p = 12$ je potřeba pro správnou inicializaci algoritmu, aby během maximálně prvních tří let bylo známo alespoň jedno lednové pozorování, alespoň jedno únorové pozorování,..., alespoň jedno prosincové pozorování.
Tyto předpoklady nám ulehčují inicializaci algoritmu, zajišťují nám, že můžeme odhadovat sezónní složky pro jednotlivá období. Definujeme L_t jako odhadovanou úroveň časové řady, T_t jako trend časové řady a Sz_t jako sezónní složku časové řady v čase t. Dále definujeme t_n^* jako max $(t_{n-1}$,t_{n-2},...), pro které platí, že t_n^* je ze stejného sezónního období jako t_n.

3.1 Aditivní Holtova-Wintersova metoda s chybějícími pozorováními

Podobně jako v kapitole (2.2.1) zavedeme následující vztahy. Pro úroveň časové řady použijeme vztah:

$$L_{t_n} = V_{t_n}(y_{t_n} - Sz_{t_n^*}) + (1 - V_{t_n})[L_{t_{n-1}} + (t_n - t_{n-1})T_{t_{n-1}}]. \qquad (3.1)$$

Vidíme, že se tento vztah podobá vztahu pro úroveň časové řady při použití aditivní Holtovy-Wintersovy metody, ale se dvěma rozdíly. Místo Sz_{t-s} používáme $Sz_{t_n^*}$, jelikož musíme vzít v úvahu, že údaj Sz_{t-s} nemáme k dispozici (příslušné pozorování nám chybí), tedy vezmeme poslední údaj o sezónnosti, který máme k dispozici. Druhým rozdílem je, že nepoužíváme přímo α, ale V_{t_n}, které definujeme následujícím způsobem:

$$V_{t_n} = \frac{V_{t_{n-1}}}{b_{t_n} + V_{t_{n-1}}}, \quad b_{t_n} = (1 - \alpha)^{(t_n - t_{n-1})}. \qquad (3.2)$$

Jedná se o rekurentní verzi vyrovnávací konstanty používané v Holtově metodě s chybějícími pozorováními místo konstanty α, viz Cipra, Trujillo a Rubio (1995):

$$V_{t_n} = \frac{1}{\sum_{i=1}^{n}(1-\alpha)^{t_n-t_i}}$$

(podobně pro další vyrovnávací konstanty).

Pro trend časové řady používáme vztah:

$$T_{t_n} = U_{t_n}\frac{L_{t_n} - L_{t_{n-1}}}{t_n - t_{n-1}} + (1 - U_{t_n})T_{t_{n-1}}. \tag{3.3}$$

Narozdíl od vzorce pro trend časové řady z (2.2.1) zde místo γ vystupuje U_{t_n}, které definujeme podobně jako V_{t_n}:

$$U_{t_n} = \frac{U_{t_{n-1}}}{d_{t_n} + U_{t_{n-1}}}, \quad d_{t_n} = (1-\gamma)^{(t_n - t_{n-1})}. \tag{3.4}$$

Pro sezónní složku časové řady používáme vztah:

$$Sz_{t_n} = W_{t_n} \cdot (y_{t_n} - L_{t_n}) + (1 - W_{t_n})Sz_{t_n^*}, \tag{3.5}$$

kde W_{t_n} definujeme jako:

$$W_{t_n} = \frac{W_{t_n^*}}{f_{t_n} + W_{t_n^*}}, \quad f_{t_n} = (1-\delta)^{(t_n - t_n^*)/p}. \tag{3.6}$$

Konstanty α, γ a δ jsou standardní vyrovnávací koeficienty pro Holtovu-Wintersovu metodu viz. vzorec (2.1). Rekurzivní vzorce (3.1), (3.2), (3.3) a (3.4) vycházejí z rozšířené Holtovy metody - Wright(1986).

Vzorce (3.5) a (3.6) popisující sezónní složku časové řady se dají vysvětlit tak, že odhadovaný sezónní faktor Sz_{t_n} bereme jako kombinaci aktuálního pozorování v čase n upraveného o aktuální odhad trendu L_t s posledním dostupným odhadem sezónní složky pro stejné sezónní období. Demonstrujeme na příkladu. Předpokládejme takovouto časovou řadu: $p = 12$, první data jsou k dispozici z únoru 2015, máme k dispozici data ze všech měsíců až do únoru 2017 včetně, s výjimkou únoru roku 2016. Potom pro $Sz_{Únor2017}$ platí:

$$Sz_{Únor2017} = W_{Únor2017} \cdot (y_{Únor2017} - L_{Únor2017}) + (1 - W_{Únor2017})Sz_{Únor2015},$$
$$W_{Únor2017} = \frac{W_{Únor2015}}{(1-\delta)^2 + W_{Únor2015}}.$$

Výše zmíněné vztahy nám nyní pomohou s výpočtem vyrovnaných hodnot a odhadů. Předpověď pro data časové řady časově vzdálené o m kroků v čase n je vyjádřena následujícím vztahem:

$$\hat{y}_{t_n+m}(t_n) = L_{t_n} + mT_n + Sz_{(t_n+m)^*} \quad \text{pro} \quad m = 1,2,... \quad , \tag{3.7}$$

kde $(t_n + m)^*$ definujeme obdobně jako t_n^* jen pro čas $t_n + m$.

Vyrovnané hodnoty časové řady definujeme jako:

$$\hat{y}_{t_n} = L_{t_n} + Sz_{t_n}. \tag{3.8}$$

Inicializace aditivní Holtovy-Wintersovy metody s chybějícími pozorováními

Pro fungování naší metody potřebujeme následující startovací proměnné: L_0, T_0,Sz_i,V_0,U_0 a W_i , kde $i \in (-p+1,-p+2,...,0)$. Pro jejich volbu se budeme řídit doporučením z Chatfield a Yar (1988) pro klasickou Holtovu-Wintersovu metodu. Nejdříve zavedeme potřebné proměnné:
Pro k_0 platí, že sezóna s pořadovým číslem k_0 je první sezóna obsahující alespoň jedno pozorování.
Pro k_1 platí, že prvních k_1 sezón obsahuje alespoň jedno pozorování pro každé období, a navíc je nejmenší ze všech k, pro které toto platí. Dále:
$\overline{y}_k$ je aritmetický průměr pozorování z k. sezóny.
A nyní už samotné vztahy pro inicializaci (jejich odůvodnění plyne z pragmatických úvah, viz Cipra, Trujillo a Rubio (1995)):

$$T_0 = \frac{\overline{y}_{k_1} - \overline{y}_{k_0}}{(k_1 - k_0)p}, \tag{3.9}$$

$$L_0 = \overline{y}_{k_0} - (k_0 p - \frac{p-1}{2})T_0, \tag{3.10}$$

$$Sz_t = \frac{1}{k(i)} \sum_k Sz_{i+kp} \quad \text{pro} \quad i \in (-p+1,-p+2,...,0), \tag{3.11}$$

$$Sz_{i+kp} = y_{i+kp} - [\overline{y}_k + (i + \frac{p-1}{2})T_0]. \tag{3.12}$$

Suma ze vzorce (3.11) je přes všechna k taková, že pozorování y_{i+kp} $(k = 1,...,k_1)$ jsou přítomna v prvních k_1 sezónách a $k(i)$ je počet sčítanců v této sumě.
Následující volby jsou jistým zjednodušením tradičního využití metody, nicméně se ukazuje, že toto zjednodušení funguje dobře:

$$V_0 = \alpha, \quad U_0 = \gamma, \quad W_{-p+1} = W_{-p+2} = ... = W_0 = \delta. \tag{3.13}$$

Multiplikativní Holtova-Wintersova metoda s chybějícími pozorováními

Podobně jako v sekci (2.2.2) dostaneme multiplikativní verzi úpravou některých vztahů. Místo vzorce (3.1) používáme:

$$L_{t_n} = V_{t_n} \frac{y_{t_n}}{Sz_{t_n^*}} + (1 - V_{t_n})[L_{t_{n-1}} + (t_n - t_{n-1})T_{t_{n-1}}].$$

Místo vzorce (3.5) používáme:

$$Sz_{t_n} = W_{t_n} \frac{y_{t_n}}{L_{t_n}} + (1 - W_{t_n}) Sz_{t_n^*}.$$

Místo vzorce (3.7) používáme:

$$\hat{y}_{t_n+m}(t_n) = (L_{t_n} + mT_{t_n}) Sz_{(t_n+m)^*}.$$

Místo vzorce (3.8) používáme:

$$\hat{y}_{t_n} = L_{t_n} Sz_{t_n}.$$

Místo vzorce (3.12) při inicializaci používáme:

$$Sz_{i+kp} = \frac{y_{i+kp}}{\overline{y}_k + (i + \frac{p-1}{2})T_0}.$$

4. Aplikace Holtovy-Wintersovy metody na aktuárská data

V této poslední kapitole bakalářské práce se zaměříme na reálnou aplikaci Holtovy-Wintersovy metody s chybějícími pozorováními. Použijeme jak aditivní, tak multiplikativní verzi metody. Příslušné výpočty jsme prováděli v programu psaném v jazyce Pascal viz. příloha (A.1). Našimi vstupními daty jsou výše škod nahlášené pojišťovně v jednotlivých letech. U každého škodního úhrnu je známo, v jakém roce vznikl a kdy byl nahlášen pojišťovně. Tato data se přirozeně vyskytují v podobě tzv. Run-off trojúhelnících.

4.1 Vstupní data

Data, se kterými budeme pracovat v této kapitole, jsou v obrázku 4.1. podle Hendrych a Cipra (2020) (motor bodily injury class of insurance business in one Australian state, viz také Taylor a Ashe (1983)):

Acc.	Development year									
year	0	1	2	3	4	5	6	7	8	9
0	357,848	766,940	610,542	482,940	527,326	574,398	146,342	139,950	227,229	67,948
1	352,118	884,021	933,894	1,183,289	445,745	320,996	527,804	266,172	425,046	
2	590,507	1,001,799	926,219	1,016,654	750,816	146,923	495,992	280,405		
3	310,608	1,108,250	776,189	1,562,400	272,482	352,053	206,286			
4	443,160	693,190	991,983	769,488	504,851	470,639				
5	396,132	937,085	847,498	805,037	705,960					
6	440,832	847,631	1,131,398	1,063,269						
7	359,480	1,061,648	1,443,370							
8	376,686	986,608								
9	344,014									

Tabulka 4.1:Vstupní data ve formě run-off trojúhelníku.

Nechť run-off trojújelník z (4.1) je A. Například číslo na pozici $A_{(0,2)}$ v run-off trojúhelníku, tedy 610542 znamená, že úhrn škod, které se staly v čase 0 a byly nahlášeny v čase 2, je právě 610542. Naším úkolem je pomocí Holtovy-Wintersovy metody s chybějícími pozorováními vyhladit daná data a předpovědět data chybějící. Tedy na run-off trojúhelník (4.1) budeme nahlížet jako na časovou řadu, jejíž členy vypadají následujícím způsobem:

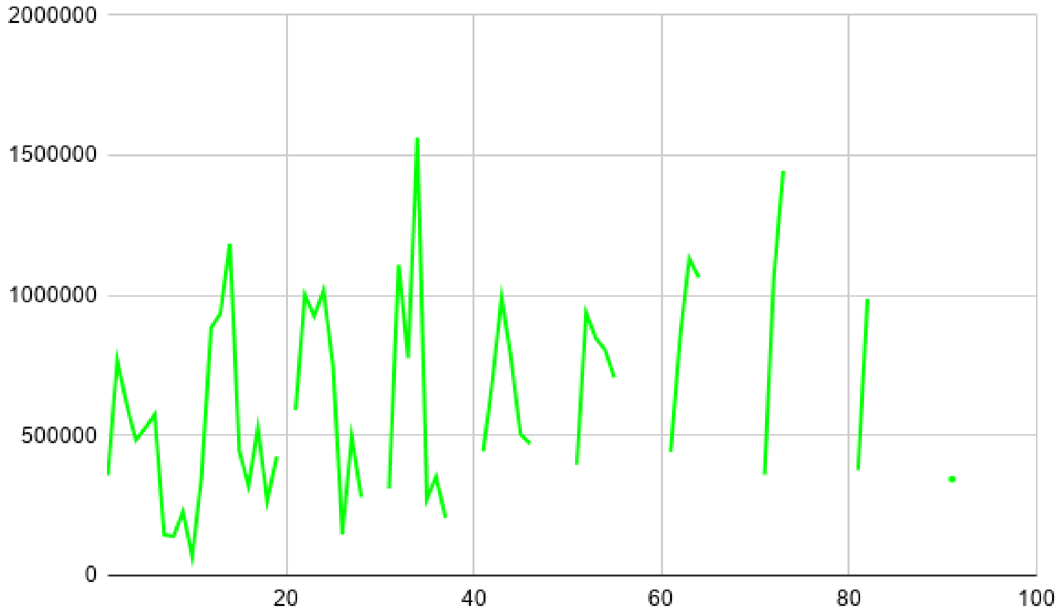
$$y_1 = A_{(0,0)}, y_2 = A_{(0,1)}, \ldots\ldots, y_{10} = A_{(0,9)}, y_{11} = A_{(1,0)},$$
$$y_{12} = A_{(1,1)}, \ldots\ldots, y_{99} = A_{(9,8)}, y_{100} = A_{(9,9)}.$$

Tedy časová řada, se kterou budeme dále pracovat, vypadá takto:

1	357848		51	396132
2	766940		52	937085
3	610542		53	847498
4	482940		54	805037
5	527326		55	705960
6	574398		56	
7	146342		57	
8	139950		58	
9	227229		59	
10	67948		60	
11	352118		61	440832
12	884021		62	847631
13	933894		63	1131398
14	1183289		64	1063269
15	445745		65	
16	320996		66	
17	527804		67	
18	266172		68	
19	425046		69	
20			70	
21	590507		71	359480
22	1001799		72	1061648
23	926219		73	1443370
24	1016654		74	
25	750816		75	
26	146923		76	
27	495992		77	
28	280405		78	
29			79	
30			80	
31	310608		81	376686
32	1108250		82	986608
33	776189		83	
34	1562400		84	
35	272482		85	
36	352053		86	
37	206286		87	
38			88	
39			89	
40			90	
41	443160		91	344014
42	693190		92	
43	991983		93	
44	769488		94	
45	504851		95	
46	470639		96	
47			97	
48			98	
49			99	
50			100	

Tabulka 4.2: Vývojový trojúhelník pro odhad technických rezerv v neživotním pojištění uspořádaný do jedné časové řady po jednotlivých řádcích a s chybějícími pozorováními (motor bodily injury class of insurance business in one Australian state, viz Taylor a Ashe (1983))

Můžeme ji znázornit následujícím grafem:



Obrázek 4.1: Vstupní data ve formě časové řady (viz tabulka 4.2).

Inicializace aditivní verze algoritmu

Nejdříve provedeme inicializaci daného algoritmu. Použijeme postup z kapitoly (3.2), přesněji vztahy (3.9)–(3.12):

$$T_0 = 0, \quad L_0 = 390146.3,$$

$$Sz_{-9} = -32298.3, Sz_{-8} = 376793.7, Sz_{-7} = 220395.7, Sz_{-6} = 92793.7,$$
$$Sz_{-5} = 137179.7, Sz_{-4} = 184251.7, Sz_{-3} = -243804.3, Sz_{-2} = -250196.3,$$
$$Sz_{-1} = -162917.3, Sz_0 = -322198.3,$$

$$\mathrm{V}_0 = 0.4, \quad U_0 = 0.1, \quad W_{-9} = W_{-8} = ... = W_0 = 0.4.$$

Inicializace multiplikativní verze algoritmu

Inicializace probíhá stejně jako u aditivní metody, kromě sezonních složek (jsou uvedeny zaokrouhlené na 3 desetinná místa, i když ve vlastním výpočtu nezaokrouhlujeme):

$$Sz_{-9} = 0.917, Sz_{-8} = 1.966, Sz_{-7} = 1.565, Sz_{-6} = 1.238,$$
$$Sz_{-5} = 1.352, Sz_{-4} = 1.472, Sz_{-3} = 0.375, Sz_{-2} = 0.359,$$
$$Sz_{-1} = 0.582, Sz_0 = 0.174.$$

4.2 Aplikace metody

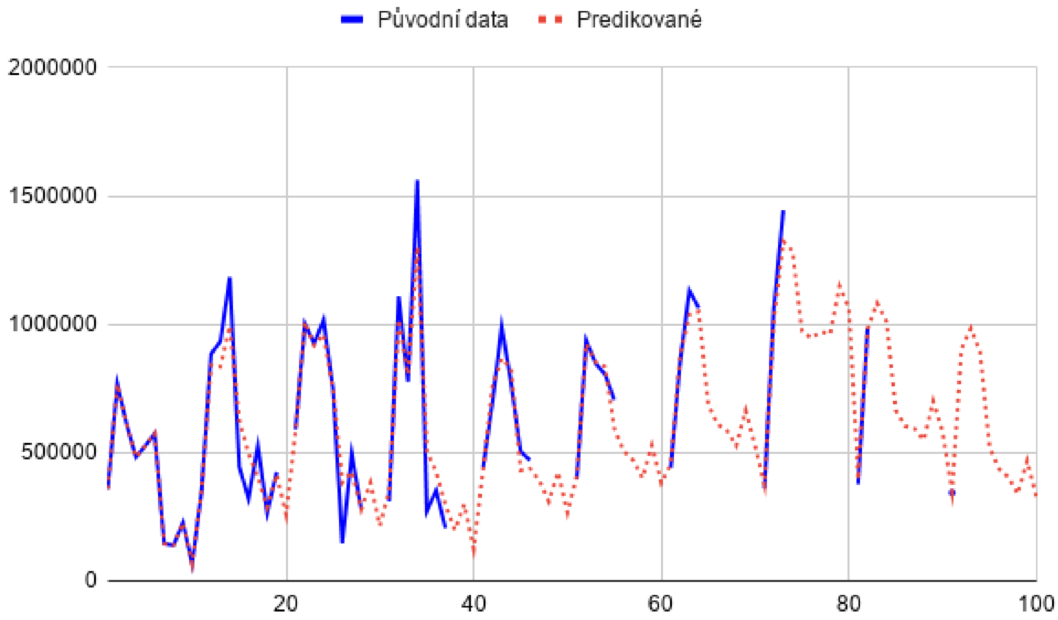
Nyní aplikujeme obě varianty Holtovy-Wintersovy metody s chybějícími pozorováními a výsledek uvedeme v následující tabulce:

	Původní data	Aditivní	Multiplikativní
1	357848	357,848.00	357,848.00
2	766940	766,940.00	766,940.00
3	610542	610,542.00	610,542.00
4	482940	482,940.00	482,940.00
5	527326	527,326.00	527,326.00
6	574398	574,398.00	574,398.00
7	146342	146,342.00	146,342.00
8	139950	139,950.00	139,950.00
9	227229	227,229.00	227,229.00
10	67948	67,948.00	67,948.00
11	352118	354,180.80	354,180.80
12	884021	840,964.21	839,926.60
13	933894	835,442.12	831,243.03
14	1183289	994,076.79	978,770.21
15	445745	626,858.80	629,992.88
16	320996	497,421.61	507,400.52
17	527804	403,953.07	395,551.65
18	266172	287,546.63	281,062.60
19	425046	415,040.38	449,085.45
20		258,377.91	144,071.83
21	590507	580,754.22	644,930.69
22	1001799	1,008,800.86	1,181,755.51
23	926219	918,443.91	986,278.40
24	1016654	961,072.81	966,042.01
25	750816	746,657.17	747,879.06
26	146923	383,347.33	375,487.15
27	495992	426,848.54	393,189.02
28	280405	278,518.04	264,724.72
29		383,276.65	421,552.76
30		215,142.65	133,296.68
31	310608	351,952.84	387,076.08
32	1108250	1,019,527.11	1,061,100.11
33	776189	804,027.55	825,181.19
34	1562400	1,302,731.32	1,300,204.92
35	272482	504,151.98	487,683.99
36	352053	426,280.58	394,435.60
37	206286	295,055.83	236,995.92
38		199,106.17	155,771.05
39		298,964.52	238,008.67
40		125,930.26	72,213.95
41	443160	434,926.84	423,926.51
42	693190	758,228.26	779,206.44
43	991983	866,942.14	890,384.77
44	769488	826,692.92	810,321.76
45	504851	431,541.51	475,451.04
46	470639	438,954.32	448,097.03
47		391,471.60	297,738.92
48		310,099.80	200,184.19
49		424,536.01	313,005.88
50		266,079.61	97,223.88

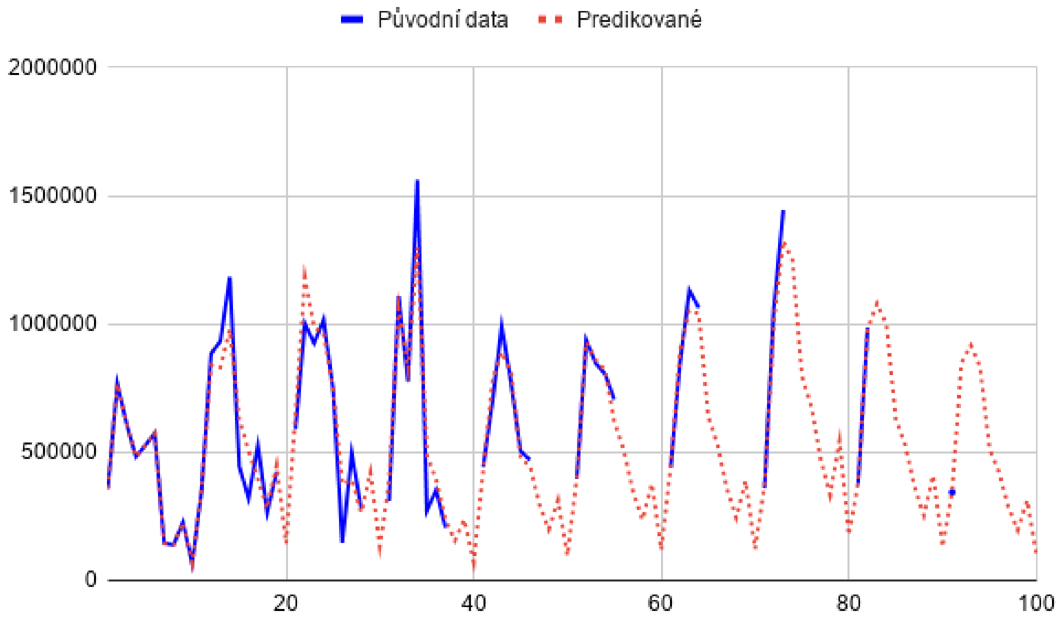
51	396132	410,364.18	400,405.38
52	937085	925,045.75	929,869.84
53	847498	852,537.14	860,933.13
54	805037	835,407.41	819,371.75
55	705960	593,899.51	619,282.19
56		506,010.93	510,608.50
57		470,831.21	345,946.57
58		401,762.39	237,055.73
59		528,501.59	377,589.09
60		382,348.18	119,424.59
61	440832	454,054.98	447,125.55
62	847631	874,345.36	895,401.10
63	1131398	1,038,178.12	1,054,008.62
64	1063269	1,049,480.25	1,040,925.33
65		688,013.66	641,907.57
66		610,428.13	528,351.72
67		585,551.46	357,377.60
68		526,785.70	244,501.76
69		663,827.95	388,859.18
70		527,977.59	122,810.23
71	359480	375,253.07	366,365.20
72	1061648	1,018,063.81	1,008,295.24
73	1443370	1,322,446.04	1,324,233.42
74		1,286,591.44	1,246,158.86
75		974,676.86	797,239.36
76		946,643.34	678,890.33
77		971,318.67	473,915.02
78		962,104.92	333,891.42
79		1,148,699.17	545,781.81
80		1,062,400.82	176,849.43
81	376686	396,201.72	385,100.55
82	986608	981,591.45	978,950.78
83		1,082,832.81	1,079,776.36
84		1,012,385.31	999,113.93
85		665,877.83	629,549.30
86		603,251.42	528,765.79
87		593,333.85	364,523.41
88		549,527.20	253,898.38
89		701,528.56	410,689.53
90		580,637.31	131,795.11
91	344014	348,956.44	346,395.48
92		905,253.99	849,640.77
93		984,125.18	916,769.73
94		891,307.53	831,095.19
95		522,429.90	513,765.37
96		437,433.32	423,866.97
97		405,145.60	287,345.19
98		338,968.79	197,009.69
99		468,599.99	313,970.16
100		325,338.59	99,353.86

Tabulka 4.3: Predikované hodnoty ve vývojovém trojúhelníku z tabulky (4.2) s využitím aditivní a multiplikativní Holtovy-Wintersovy metody s chybějícími pozorováními

A zobrazíme naše výsledky:



Obrázek 4.2: Aditivní Holtova-Wintersova metoda s chybějícími pozorováními



Obrázek 4.3: Multiplikativní Holtova-Wintersova metoda s chybějícími pozorováními

4.3 Porovnání multiplikativní a aditivní verze algoritmu

V této podkapitole porovnáme, která křivka prokládá data lépe. K tomuto účelu definujeme *střední čtvercovou chybu předpovědi (MSE)* jako:

$$MSE = \sum_{j=1}^{n} (y_{t_j} - \hat{y}_{t_j}(t_{j-1}))^2/n, \tag{4.1}$$

kde y_{t_j} je skutečná hodnota časové řady v čase t_j a $\hat{y}_{t_j}(t_{j-1})$ je odhad y_{t_j} provedený v čase t_{j-1}. Pro jednoduchost časový index nyní probíhá po řádcích jen přes známé hodnoty z daného vývojového trojúhelníku (viz také obr. 4.4 a 4.5). Pro lepší představu o velikosti dané odchylky definujeme *RMSE - odmocninu střední čtvercové odchylky*, která se měří ve stejných jednotkách jako pozorované hodnoty:

$$RMSE = (MSE)^{(1/2)}. \tag{4.2}$$

Důležité je, že MSE (případně $RMSE$) můžeme počítat pouze, pokud je v daném čase známa hodnota časové řady. Tedy dostáváme následující tabulku:

Čas	Aditivní $\hat{y}_{t_j}(t_{j-1})$	Multiplikativní $\hat{y}_{t_j}(t_{j-1})$	Reálná naměřená hodnota
1	357,848.00	357,848.00	357,848.00
2	766,940.00	766,940.00	766,940.00
3	610,542.00	610,542.00	610,542.00
4	482,940.00	482,940.00	482,940.00
5	527,326.00	527,326.00	527,326.00
6	574,398.00	574,398.00	574,398.00
7	146,342.00	146,342.00	146,342.00
8	139,950.00	139,950.00	139,950.00
9	227,229.00	227,229.00	227,229.00
10	67,948.00	67,948.00	67,948.00
11	357,848.00	357,848.00	352,118.00
12	764,418.80	761,536.56	884,021.00
13	660,416.57	648,752.42	933,894.00
14	657,699.53	615,181.26	1,183,289.00
15	948,838.88	957,544.67	445,745.00
16	811,067.14	838,786.34	320,996.00
17	183,773.65	160,436.35	527,804.00
18	325,545.97	307,534.79	266,172.00
19	397,252.60	491,822.26	425,046.00
20	556,191.67	781,997.76	590,507.00
21	1,023,705.40	1,564,820.82	1,001,799.00
22	903,168.23	1,104,276.73	926,219.00
23	856,421.40	870,746.88	1,016,654.00
24	739,011.05	742,479.42	750,816.00
25	812,090.39	789,976.18	146,923.00
26	302,465.80	208,255.91	495,992.00
27	275,139.70	236,651.37	280,405.00
28	507,986.72	675,663.75	310,608.00
29	799,966.53	944,419.34	1,108,250.00
30	862,824.06	928,655.30	776,189.00
31	794,611.71	787,141.63	1,562,400.00
32	939,387.47	891,981.30	272,482.00
33	562,577.37	472,258.59	352,053.00
34	455,916.51	292,645.68	206,286.00
35	386,310.84	310,354.66	443,160.00
36	938,641.78	1,017,812.58	693,190.00
37	589,637.20	665,068.89	991,983.00
38	941,425.98	892,219.98	769,488.00
39	291,950.32	419,469.52	504,851.00
40	380,328.35	406,387.75	470,639.00
41	546,526.83	441,289.87	396,132.00
42	888,851.75	908,178.66	937,085.00
43	864,071.06	891,684.34	847,498.00
44	897,265.17	848,568.45	805,037.00
45	378,751.92	452,867.40	705,960.00
46	665,610.26	547,816.48	440,832.00
47	959,085.93	1,046,932.55	847,631.00
48	820,530.21	873,321.46	1,131,398.00
49	1,021,128.39	994,983.21	1,063,269.00
50	812,727.40	557,329.95	359,480.00
51	874,962.40	833,120.37	1,061,648.00
52	1,036,630.53	1,042,642.58	1,443,370.00
53	1,386,187.26	811,950.47	376,686.00
54	964,761.21	953,261.23	986,608.00
55	856,820.37	591,105.97	344,014.00

Tabulka 4.4: Reálné hodnoty časové řady v čase t a odhady jednotlivých metod pro hodnotu časové řady v čase t učiněné v čase $t-1$

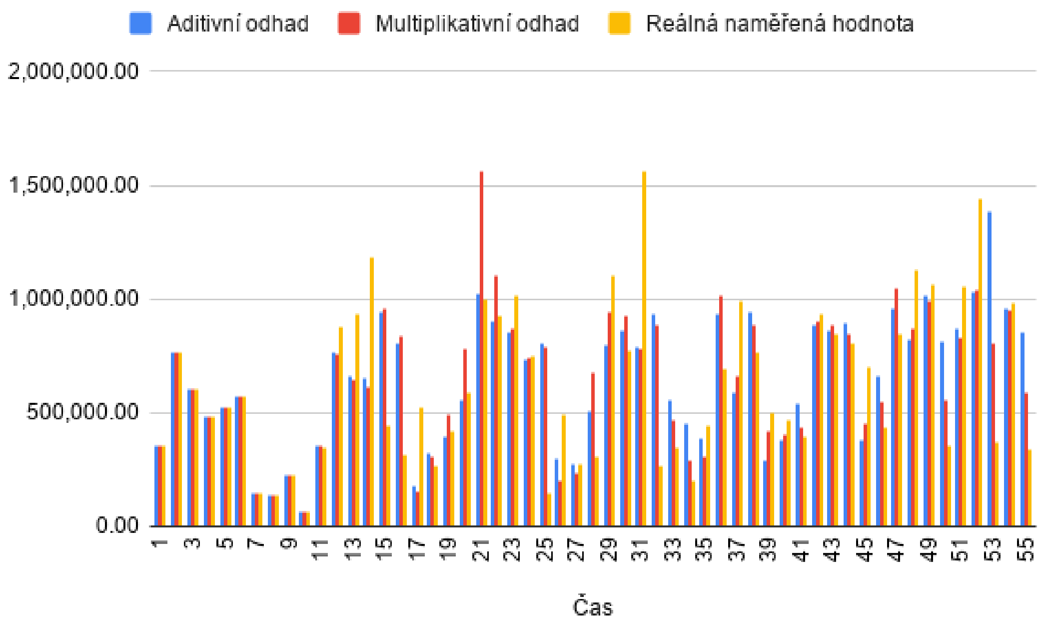
Nyní spočítáme MSE a $RMSE$ z dat uvedených v tabulce (4.4) a dostáváme následující výsledky:

$$MSE \text{ pro Aditivní metodu: } 92190023765$$
$$RMSE \text{ pro Aditivní metodu: } 303628$$
$$MSE \text{ pro Multiplikativní metodu: } 75244213475$$
$$RMSE \text{ pro Multiplikativní metodu: } 274307$$

Tedy vidíme, že z pohledu $RMSE$ (MSE) je multiplikativní metoda přesnější s menší RMSE.

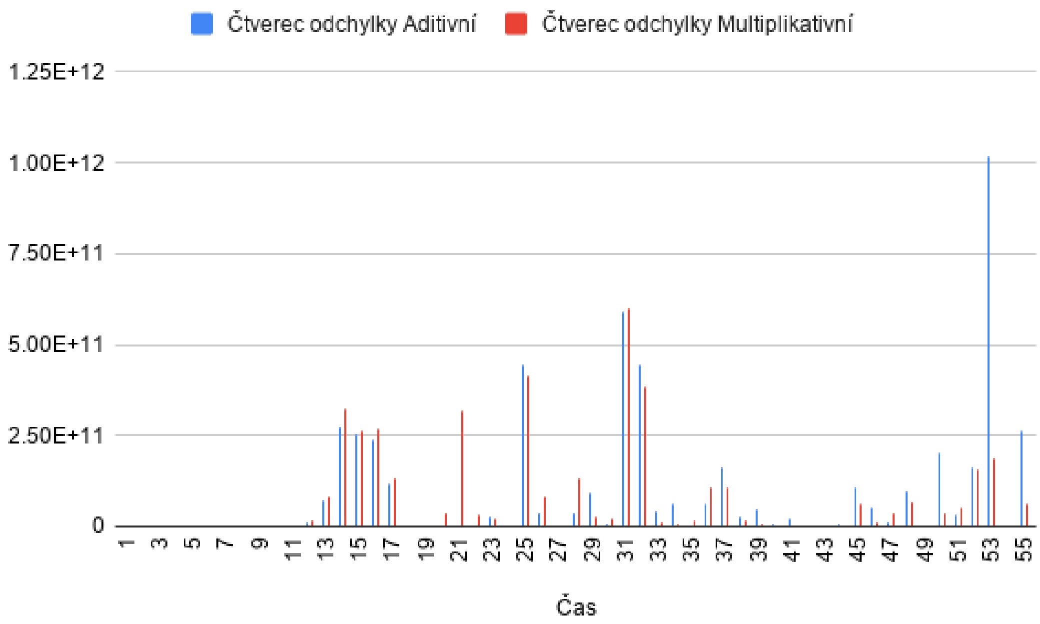
Grafické znázornění odhadů

Nejdříve znázorníme data z tabulky (4.4):



Obrázek 4.4: Zobrazení tabulky (4.4) – Reálné hodnoty časové řady v čase t a odhady jednotlivých metod pro hodnotu časové řady v čase t učiněné v čase $t-1$.

A nyní znázorníme jednotlivé čtverce odchylek obou metod:



Obrázek 4.5: Čtverce odchylek odhadů učiněných v čase $t-1$ jednotlivých metod od reálných dat časové řady.

Z grafů můžeme vyčíst, že hlavní příčinou toho, že aditivní metoda hůře odhadla reálná data, je chování křivek v čase 53, jak je vidět v obrázku (4.5), jelikož zde došlo k velkému poklesu a aditivní metoda na rychlé změny reaguje hůže než metoda multiplikativní.

Závěr

Předmětem této práce bylo představení různých přístupů k vyrovnání a predikcím časových řad. Ukázali jsme základní principy exponenciálního vyrovnání, jak námi probírané metody fungují, jejich předpoklady a možnosti převedení do rekurentních formulí, které usnadňují algoritmizaci. Následně jsme se zaměřili na časové řady, u nichž se vyskytuje sezónní složka, a popsali jsme Holtovu-Wintersovu metodu, která dokáže s řadami se sezónní složkou pracovat. Holtovu-Wintersovu metodu jsme potom rozšířili na Holtovu-Wintersovu metodu s chybějícími pozorováními, která za určitých podmínek nevyžaduje úplnost původních dat. Ukázali jsme dvě varianty této metody, a to aditivní a multiplikativní verzi. V poslední části této práce jsme Holtovu-Wintersovu metodu s chybějícími pozorováními aplikovali na reálná data a následně jsme porovnávali, která z verzí Holtovy-Wintersovy metody dokázala lépe předpovídat chování časové řady. Multiplikativní metoda se ukázala jako spolehlivější pro naše reálná data a z grafických výstupů můžeme usoudit, že Holtova-Wintersova metoda s chybějícími pozorováními dokáže, navzdory své jednoduchosti, poměrně dobře odhadovat chování časové řady.

Seznam použité literatury

[1] Hendrych, R., Cipra, T.,: *Applying state space models to stochastic claims reserving.* Astin Bulletin 51(1), 2020, 267-301.
[2] Cipra, T.: *Finanční ekonometrie.* Ekopress (2. vyd.), Praha, 2013.
[3] Koritarová L.: *Holtova-Wintersova metoda pro sezónní vyrovnání.* Bakalářská práce, MFF UK 2014.
[4] Cipra, T., Trujillo, J., Rubio, A.: *Holt-Winters Method with missing observations.* Management Science 41(1), 1995, 174-178.
[5] Taylor, G. C. and Ashe, F. R.: *Second moments of estimates of outstanding claims.* Journal of Econometrics 23, 1983, 37-61.

Seznam obrázků

Seznam tabulek

A. Přílohy

A.1 Zdrojový kód v jazyce Pascal

```
I_n:array [-11..1000] of real;
W:array[-11..1000] of real;
U:array[0..1000] of real;
p:integer;
V:array[0..1000] of real;
smooth:real;
ocekavana:real;
multi:integer;
zarazka:integer;
minule:integer;
UsporadaneI:array [-11..1000] of real;
 UsporadaneW :array[-11..1000] of real;
procedure Multiplikativni02;
var
pocitadlo:integer;
zarazka2:integer;
posun:integer;
pauza:integer;
 zarazka:integer;
 hvezdickaI:real;
 hvezdickaW:real;
 hvezdickaP:real;
 counter:integer;
 interpol:real;

 begin
   {nejdříve si zkusím vypsat všechny důležité hodnoty}
   zarazka2:=0;
   zarazka:=0;
   pocitadlo:=0;
   posun:=0;
   pauza:=1;
   while zarazka2<>42 do
         begin
         pocitadlo:=pocitadlo+1;
         if Data[pocitadlo]<>31111 then
            begin;
              posun:=posun+1;
              V[posun]:=V[posun-1]/((1-alfa)**(pauza)+V[posun-1]);
              {najdu odpovídající I_n}
              zarazka:=0;
              counter:=0;
              while zarazka<>42 do
```

```
                  begin
                    counter:=counter+1;
                    if Data[pocitadlo-counter*p]<>31111 then
                       begin
                       hvezdickaI:=UsporadaneI[pocitadlo-counter*p];
                       hvezdickaW:=UsporadaneW[pocitadlo-counter*p];
                       hvezdickaP:=pocitadlo-counter*p;
                       zarazka:=42;
                       end;
                  end;
             if multi=1 then S[posun]:=V[posun]*Data[pocitadlo]/hvezdickaI+▌
(1-V[posun])*(S[posun-1]+pauza*T[posun-1])
             else S[posun]:= V[posun]*(Data[pocitadlo]-hvezdickaI)+
(1-V[posun])*(S[posun-1]+pauza*T[posun-1]);
                  {dopočítej další neznámé}
                  U[posun]:=U[posun-1]/((1-gama)**(pauza)+U[posun-1]);▌
                  T[posun]:=U[posun]*(S[posun]-S[posun-1])/(pauza)+
(1-U[posun])*T[posun-1];
                  W[posun]:=hvezdickaW/((1-delta)**((pocitadlo-hvezdickaP)/p)▌
+hvezdickaW);
                 if multi=1 then I_n[posun]:=W[posun]*Data[pocitadlo]/S[posun]▌
+(1-W[posun])*hvezdickaI
                 else I_n[posun]:=W[posun]*(Data[pocitadlo]-S[posun])
+(1-W[posun])*hvezdickaI;
                  {ještě spočítám interpolated}
             if multi=1 then interpol:=(S[posun-1]+pauza*T[posun-1])*hvezdickaI▌
             else interpol:=S[posun-1]+pauza*T[posun-1]+hvezdickaI;
                  {vypíšu}
                  write(interpol:2:2);
                  write('   ');
                  if multi=1 then writeln(S[posun]*I_n[posun]:2:2)
                  else writeln(S[posun]+I_n[posun]);
                  UsporadaneI[pocitadlo]:=I_n[posun];
                  UsporadaneW[pocitadlo]:=W[posun];
                  pauza:=1;
                end;
             if Data[pocitadlo]=31111 then
                begin
                zarazka:=0;
                counter:=0;
                pauza:=pauza+1;
                {vypíšu alespoň interpolated}
                {najdu hvezdicku I}
                while zarazka<>42 do
                      begin
                       counter:=counter+1;
                       if Data[pocitadlo-p*counter]<>31111 then
                          begin
```

```
                       hvezdickaI:=UsporadaneI[pocitadlo-counter*p];
                       zarazka:=42;
                      end;
                  end;
            if multi=1 then interpol:=(S[posun]+(pauza-1)*T[posun])
*hvezdickaI
            else interpol:=S[posun]+(pauza-1)*T[posun]+hvezdickaI;
            writeln(interpol:2:2)
            end;
         if pocitadlo=pocet then zarazka2:=42;

         end;
  end;

 procedure Inicializace(D:array of real);
 var
 a,b:integer;
 K_0:integer;
 zarazka:integer;
 K_1:integer;
 posuvnik:integer;
 pocitadlo:integer;
 K_0_prumer:real;
 K_1_prumer:real;
 pocitadloK_0:integer;
 pocitadloK_1:integer;
 soucet:real;
 hodnoty:array[1..1000]of real;
 pomocny:real;
 counter:integer;
 prumer_k:real;
 prazdne:integer;
 dodelavka:integer;
 begin
   {nejdřív zvolím hodnoty k_1 a u  k_0 předpokládámě že se rovná 1}
   K_0:=1;
   zarazka:=0;
   K_1:=0;
   counter:=0;
   soucet:=0;


 for i:=1 to p do  {podívám se kde se poprvé vyskytne každá sezona
 a vezmu maximum}
    begin
     posuvnik:=i;
     pocitadlo:=0;
      while zarazka<>42 do
```

```
            begin
              if Data[posuvnik]<>31111 then
                 begin
                   pocitadlo:=pocitadlo+1;
                   if K_1<pocitadlo then K_1:=pocitadlo;
                   zarazka:=42;
                 end
               else
                  begin
                    posuvnik:=posuvnik+p;
                    pocitadlo:=pocitadlo+1;
                  end;

            end;
      zarazka:=0;
    end;
   {Nyní je načase spočítat hodnoty v čase 0 - nejdříve průměry}
   K_0_prumer:=0;
   K_1_prumer:=0;
   pocitadloK_0:=0;
   pocitadloK_1:=0;
   for i:=1 to p do
      begin
        if Data[i]<>31111 then  K_0_prumer:= K_0_prumer + Data[i]
        else pocitadloK_0:=pocitadloK_0+1;
        if Data[(K_1-1)*p+i]<>31111 then  K_1_prumer:= K_1_prumer
+ Data[(K_1-1)*p+i]
        else pocitadloK_1:=pocitadloK_1+1;
      end;
   {Teď dopočítám okamžitě T0 a S0}
   K_0_prumer:=K_0_prumer/(p-pocitadloK_0);
   K_1_prumer:=K_1_prumer/(p-pocitadloK_1);
   if  K_1-K_0=0 then dodelavka:=1
   else dodelavka:=0;
   T[0]:=(K_1_prumer-K_0_prumer)/(( K_1-K_0)*p+dodelavka);
   S[0]:=K_0_prumer-(K_0*p-(p-1)/2)*T[0];
   {Nyní potřebuju několik hodnot In}
   for i:=1 to p do
       begin
         {nejdříve zjistím kolik je dat o tomto ,,měsíci" v K_1 sesonach}
         pocitadlo:=0;
         prazdne:=0;
         for a:=1 to 1000 do    {cele hodnoty nastavim na prazdné = 31111}
             begin
              hodnoty[a]:=31111;
             end;
         zarazka:=0;
         posuvnik:=0;
```

```
         soucet:=0;
         {nactu prislusna data}
         for b:=1 to K_1 do
             begin
               if Data[i+p*(b-1)]<>31111 then
                  begin
                  pocitadlo:=pocitadlo+1;        {pridam o 1}
                  hodnoty[pocitadlo]:= Data[i+p*(b-1)];  {poznamenám
si dané číslo}
                  end
               else
                   begin
                     pocitadlo:=pocitadlo+1;
                     prazdne:=prazdne+1;
                   end;
             end;
         {nyní spočítám l_i+kp viz článek}
         for b:=1 to (pocitadlo - prazdne) do
             begin
               counter:=0;
               pomocny:=0;
               {zjistím si první nenulové pole v hodnotach}
                while zarazka<>42 do
                      begin
                        posuvnik:=posuvnik+1;
                        if hodnoty[posuvnik]<>31111 then
                           begin
                           zarazka:=42;
                           end;
                      end;
                zarazka:=0;

                {jeste potrebuji prumer k te sesony}
                 for a:=1 to p do
                     begin
                      if Data[(posuvnik-1)*p+a]<>31111 then
                         begin
                         pomocny:=pomocny+ Data[(posuvnik-1)*p+a];
                         counter:=counter+1;
                         end;
                     end;
         {vypocitam konkretni l_i+kp}
         prumer_k:=pomocny/counter;
       if multi=1 then  soucet:=soucet+
(hodnoty[posuvnik])/(prumer_k + (-p+i+(p-1)/2)*T[0])
       else soucet:=soucet+hodnoty[posuvnik]
-(prumer_k+(-p+i+(p-1)/2)*T[0]);
          end;
```

```
         I_n[-p+i]:=soucet/(pocitadlo-prazdne);
         UsporadaneI[-p+i]:=soucet/(pocitadlo-prazdne);
       end;
{Tímhle jsem spočítal všechny I_n, ufff}
{je načase spočítat W_i}
for i:=1 to p do
    begin
     W[-p+i]:=1-(1-delta); {předpokládám že Q=1}
     UsporadaneW[-p+i]:=1-(1-delta);
    end;
V[0]:=1-(1-alfa);
U[0]:=1-(1-gama);
end;

begin
writeln('Predpoklady: prvni sesona ma alespon jedno pozorovani co nechybi.
 Casy jednotlivych pozorovani jsou integery.');
writeln('Pokud pozorovani chybi napiste prosim hodnotu 31111, s tim jde
tichy predpoklad, ze 31111 neni hodnota zadneho pozorovani');
write('hodnota alfa:');    {načtu parametry}
read(alfa);
write('hodnota gama:');
read(gama);
write('hodnota delta:');
read(delta);
write('Kolik hodnot (vcetne chybejicich pozorovani):');        {načtu data}
read(pocet);
write('delka sezony:');
read(p);
write('pro multiplikativni napiste 1, jinak 0:');
read(multi);
writeln('zacnete psat data, jednotliva data oddelujte entrem');
for i:=1 to 1000 do
   begin
     Data[i]:=31111;   {31111 bude značit že daný údaj není}
   end;
for i:=1 to pocet do
    begin
     read(Data[i]);
    end;
Inicializace(Data);     {Inicializuji potřebná data}
{a teď to spustíme}
 Multiplikativni02;

readln;
readln;
end.
```