

Jihočeská univerzita

Ekonomická fakulta

BAKALÁŘSKÁ PRÁCE

Michael Kaška

**Technická analýza akciových a měnových trhů a hodnocení její
spolehlivosti**

Vedoucí bakalářské práce: Mrkvička Tomáš, doc. RNDr., Ph. D.

Studijní program: Aplikovaná matematika

Studijní obor: Finanční a pojistná matematika

Třebíč 2020

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Fakulta ekonomická

Akademický rok: 2015/2016

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michael KAŠKA**
Osobní číslo: **E14331**
Studijní program: **B1103 Aplikovaná matematika**
Studijní obor: **Finanční a pojistná matematika**
Název tématu: **Technická analýza akciových a měnových trhů a hodnocení její spolehlivosti**
Zadávající katedra: **Katedra aplikované matematiky a informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je prozkoumání různých technických analýz a jejich srovnání z hlediska dlouhodobé spolehlivosti predikce trendů akciových a měnových trhů. Porovnání bude provedeno základními statistickými metodami, jako je korelační koeficient či regresní analýza.

Metodický postup:

1. Studium literatury - 1. semestr.
2. Sběr dat - 2. semestr.
3. Vyhodnocení informací - 2. semestr.
4. Zpracování výsledků - 3. semestr.

Rozsah grafických prací: **dle potřeby**

Rozsah pracovní zprávy: **40 - 50 stran**

Forma zpracování bakalářské práce: **tištěná**

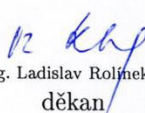
Seznam odborné literatury:

1. **Anděl, J. (1998).** *Statistické metody.* **Praha: Matfyzpress.**
2. **Cipra, T. (2005).** *Praktický průvodce finanční a pojistnou matematikou.* **Praha: Ekopress, s.r.o.**
3. **Ehrman, Douglas S. (2006).** *The Handbook of pairs trading.* **Hoboken, New Jersey: John Wiley & Sons, Inc.**
4. **Lien, K. (2013).** *Forex - ziskové intradenní a swingové obchodní strategie.* **Praha: FXstreet.cz.**
5. **Williams, Larry R. (2007).** *Jak jsem vydělal milion dolarů za rok obchodováním komodit.* **Praha: Centrum finančního vzdělávání, s.r.o.**
6. **Zima, P., & Brown, L. R. (1996).** *Schaum's Outline of Theory and Problems of Mathematics of Finance.* **New York: McGraw-Hill Companies, Inc.**

Vedoucí bakalářské práce: **doc. RNDr. Tomáš Mrkvička, Ph.D.**
Katedra aplikované matematiky a informatiky

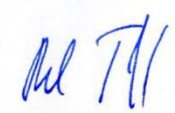
Datum zadání bakalářské práce: **15. ledna 2016**

Termín odevzdání bakalářské práce: **15. dubna 2017**


doc. Ing. Ladislav Rolínek, Ph.D.

děkan

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
EKONOMICKÁ FAKULTA
Studentská 13 (26)
370 05 České Budějovice


prof. RNDr. Pavel Tlustý, CSc.
vedoucí katedry

V Českých Budějovicích dne 8. února 2016

Prohlášení,

Prohlašuji, že svou práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě/v úpravě vzniklé vypuštěním vyznačených částí archivovaných Ekonomickou fakultou - elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Třebíči, dne 24. dubna 2020

.....

Michael Kaška

Poděkování

Děkuji vedoucímu bakalářské práce doc. RNDr. Tomášovi Mrkvičkovi, Ph. D za hodnotné připomínky a rady, které výrazně posunuly kvalitu práce.

Děkuji své rodině za výjimečnou trpělivost, kterou se mnou měla po dobu, kterou jsem bakalářské práci věnoval.

1 OBSAH

1	Obsah	1
2	Úvod	3
3	Literární rešerše	5
3.1	Charakteristika akcií, cizích měn a komodit	5
3.1.1	Akcie	5
3.1.2	Měny	6
3.1.3	Komodity	7
3.2	Fundamentální analýza	7
3.3	Technická analýza	7
3.3.1	Grafy	8
3.3.2	Technické Indikátory	9
3.4	Statistické metody	14
3.4.1	Popisná statistika	14
3.4.2	Náhodná veličina	15
3.4.3	Příklad použití základní popisné statistiky na vzorovém souboru závěrečných měsíčních cen indexu SP500 let 2008-2010	17
3.4.4	Testování normality	18
3.4.5	Testy střední hodnoty	18
3.5	Investiční instrument, portfolio	20
3.5.2	Měření výkonnosti portfolia	21
3.5.3	Dvoudimenzionální metody měření portfolia	22
4	Praktická část	24
4.1	Úvod	24
4.2	Stanovení portfolia a definice strategie	24
4.2.1	Akciové portfolio	25
4.2.2	Portfolio měnových spekulací	25

4.3	Definování strategie pomocí technické analýzy.....	26
4.4	Technické prostředky	29
4.5	Příklady vygenerovaných pozic	31
4.5.1	Příklad pozice na instrumentu AIG.....	31
4.5.2	Příklad pozice na instrumentu DKKUSD=X	32
4.6	Testy portfolií	34
4.6.1	Akcie	34
4.6.2	FX Spekulace.....	39
4.7	Hodnocení výkonnosti portfolií.....	44
5	Závěr	46
6	Literatura	48
7	Internetové zdroje	48
8	Rejstříky	49
8.1	Seznam obrázků	49
8.2	Seznam grafů.....	49
8.3	Seznam tabulek.....	50
9	Přílohy	51
9.1	Import dat z yahoo.finance.com do lokální databáze v MySQL.....	51
9.2	Optimalizace strategie na historických datech, vygenerování Out of sample dat.....	53
9.3	Kontrola konzistence stažených dat	72

2 ÚVOD

S postupující globalizací výroby, obchodu, digitalizací informací a v neposlední řadě spolu s rozvojem komunikace, došlo v posledních padesáti letech k prudkému rozvoji mezinárodního obchodu a finančnictví. Svět opustil fixní měnové kurzy, nové technologie umožnily rozvoj tradičních nástrojů finančních trhů i vznik nových. Původně exkluzivní instrumenty obchodované největšími bankami a obchodními společnostmi byly díky technologickým změnám zpřístupněny masové nabídce pro koncové neboli retailové zájemce. Makléřské společnosti zprostředkovávají přístup k finančním trhům, po splnění minimálních podmínek, prakticky komukoliv ze zájemců. V novém odvětví byla zavedena regulace jejímž cílem je ochránit zákazníky před podvodným jednáním nabízejících společností. Regulace je však díky globálnímu charakteru nabídky velmi omezená a zabraňuje jen nejhorším excesům. Oblíbeným marketingovým mottem společností zprostředkovávajících pro veřejnost obchodování finančních instrumentů, je příslib dosažení finanční nezávislosti s vynaložením minimálního úsilí. Propagovanou metodou k dosažení kýženého cíle finanční nezávislosti, je většinou dodržování různých strategií, podporovaných širokou nabídkou prezentovaných analýz a způsobů jejich aplikace. Marketingová komunikace makléřských společností vyvolává dojem, že po účasti na kurzu obchodování již nestojí nic v cestě k dosažení výjimečných zisků.

Jedním z nejvíce popularizovaných přístupů, která má obchodníkům pomoci dosáhnout vytoužených zisků je technická analýza trhů. Mnoho začínajících spekulantů se vlivem marketingových kampaní makléřských společností vrhá bez hlubších znalostí na trh a ve většině případů zde ztrácejí své úspory. Z povinně zveřejňovaných varování makléřů podléhajících regulaci vyplývá, že na trzích ztrácí 70-80 % retailových spekulantů. Část ze zbývajících 20-30 % je pasivní, nevyvíjí aktivitu a zbytek spekulantů je v různé míře ziskových.

Je skutečně možné za použití metod technické analýzy dosáhnout dlouhodobý zisk?

Pojmem technická analýza zahrnuje mnoho přístupů k vyhodnocování trhu. Často jsou odhady analytiků díky prvku subjektivního vyhodnocování vzájemně protichůdné. Pro minimalizaci subjektivního prvku v aplikaci technické analýzy a s přihlédnutím

k doporučenému rozsahu této práce jsem zvolit za cíl otestovat přínos části technické analýzy využívající technických indikátorů.

Změny, které umožnily zpřístupnění finančních trhů široké veřejnosti, podmínily také rozvoj možností statistické analýzy a modelování. Metody dříve vyhrazené pro majitele supervýkonných serverů a nákladného specializovaného software může dnes aplikovat za pomoci open source software i majitel výkonnějšího domácího počítače.

Zpřístupněné jsou datové zdroje o pohybech cen různých instrumentů, které dnes lze ve standardizovaných formátech stáhnout do vlastního software a dále zpracovávat.

V první části práce stručně charakterizuji akcie, komodity a měny. V další části specifikuji indikátory použité v praktické části. Stručně popíšu statistické metody, které použiji při testování hypotéz o úspěšnosti technické analýzy. V praktické části vyberu několik instrumentů z nejběžnějších oblastí, kde jsou realizovány spekulace retailu. Popíšu způsob, jakým jsem získal data o cenových pohybech instrumentů z otevřených zdrojů. V následujících krocích přiblížím způsob aplikace technické analýzy na vybrané instrumenty a jaké jsem získal výsledky. Statisticky otestuji a porovnam výsledky technické analýzy a příjmu závěr.

V závěru se pokusím odpovědět na otázku, zda pomohla technická analýza řídit ziskové portfolia nebo ne.

3 LITERÁRNÍ REŠERŠE

3.1 CHARAKTERISTIKA AKCIÍ, CIZÍCH MĚN A KOMODIT

Nejběžnějšími způsoby, jak realizovat na finančním trhu zisk v souvislosti s finančními instrumenty, je zisk z prodeje a zisk z držení aktiva. Zisk z prodeje je dán rozdílem mezi prodejní cenou a cenou nákupní, dále snížený o transakční náklady související s pořízením a prodejem případně o náklady souvisejícím s držením CP. Příkladem nákladu souvisejícím s držením je náklad na úvěr při pořízení, poplatek při nákupu akcie, spread při nákupu měnového páru apod. Nejběžnějším příjmem plynoucí z držení cenného papíru je dividenda akcie, případně kupón z obligace.

3.1.1 Akcie

Akcie jsou majetkovým cenným papírem umožňující emitentům zvýšení vlastního kapitálu. Akcie s sebou nesou různě silná práva pro účast na zisku a rozhodování o chodu společnosti. Akcie je možné obchodovat veřejně, či neveřejně. Při obchodování akcií je stanovována jejich cena, resp. kurz. Veřejná obchodovatelnost zvyšuje likviditu akcií, klade vyšší nároky na transparentnost a nutí společnosti plnit podmínky trhu (burzy) na kterém jsou její akcie obchodovány. (Rejnuš, 2014)

U akcií realizujeme zisk z prodeje v zásadě dvojitým způsobem. V případě nákupní neboli dlouhé pozice kupujeme akcii za určitou cenu a předpokládáme, že v budoucnu budeme realizovat prodej za cenu vyšší. V případě spekulace na pokles ceny si obchodník akcii vypůjčí a okamžitě ji prodá, čímž získá hotovost ve výši aktuální prodejní ceny akcie. Poté, co cena akcie klesne k předpokládané ceně, akcii koupí, čímž sníží svoji hotovost a akcii vrátí zpět. Zisk je dosažen díky poklesu ceny vypůjčené akcie. Typickým výnosem z držení akcie je dividenda připisovaná na akcii. Dividenda je většinou definovaná jako část rozdělovaného zisku na jednu akcii. Přístupy k dosažení zisku lze kombinovat, případně obchodovat namísto akcií jejich deriváty atd.

3.1.2 Měny

Měna slouží ke stanovení hodnoty aktiv. Jejím projevem jsou peníze. Peníze mohou mít hmotnou podobu mincí, bankovek případně komodit nebo nehmotnou podobu ve formě digitálního záznamu v bance nebo v distribuovaném digitálním řetězci. Peníze kromě transakční funkce poskytují i funkci uchovatele hodnoty.

Tradičně chápané měny jsou zpravidla řízeny centrálními bankami, které se pomocí nástrojů monetární politiky snaží dosahovat zákonem specifikovaných makroekonomických cílů. Měna může být akceptována i mimo území daného státu. Výměnné poměry mezi jednotlivými měnami nazýváme směnným kurzem. Hodnota měny je pak definována jejím směnným kurzem vůči ostatním měnám. Moderní digitální měny, nazývané kryptoměny, nejsou řízeny centrální autoritou. Jejich vznik je vázaný na poskytnutý výpočetní výkon a rychlost uvolňování dána předem známým algoritmem. Klíčovým faktorem úspěšnosti jakékoliv měny, tradiční nebo digitální, je důvěra, se kterou je přijímána účastníky trhu. Při ztrátě důvěry se hodnota měny ztrácí a může se stát bezcennou. Naopak v období davového nadšení, nazývaného někdy tržní bublina, může její hodnota, podobně jako u kteréhokoliv jiného aktiva růst exponenciálně a dosáhnout dlouhodobě neudržitelných hodnot.

Na přelomu 19. a 20. století byly světové měny pevně navázány na „zlatý standard“. Hodnota měn byla definována jako hodnota určité hmotnosti zlata. Po první světové válce byl v důsledku inflace zlatý standard opuštěn. V roce 1944 na konferenci 44 států vznikl Bretton-Wood systém. Základem systému byla hodnota amerického dolaru vůči zlatu. Hodnoty dalších měn byly stanoveny kurzem k dolaru. Americký dolar byl ustaven světovou rezervní měnou. Se statusem rezervní měny se vázala povinnost na vyžádání směnit americký dolar za zlato v předem stanoveném pevném kurzu. Postupující inflace znemožnila americké vládě tomuto závazku dostát. V roce 1971 odmítla žádost Velké Británie a Francie. Následkem se Bretton-Wood systém zhroutil. Po zhroutilí systému a pokusech o jeho přeměnu přešla většina států na systém plovoucích kurzů. Od té doby jsou kurzy mezi národními měnami stanovovány v zásadě poptávkou a nabídkou po měně. Z „Foreign exchange market“ se postupně vyvinul název Forex. Pod tímto názvem dnes chápeme mezinárodní trh měn, kde vlády a

organizace realizují své obchody vyplývající z jejich potřeby prodeje a nákupu měn. (Jílek, 2013)

3.1.3 Komodity

Komodity jsou jako primární suroviny spjaty s průmyslem a zemědělstvím. Doložitelná historie komoditních trhů sahá až do 4000 let před n.l.

3.2 FUNDAMENTÁLNÍ ANALÝZA

V případě komodit hodnotí fundamentální analýza výhled poptávky a nabídky po daných surovinách. Snaží se na základě hospodářského cyklu, fiskální a monetární politiky zemí hlavních odběratelů a producentů odhadnout jejich vývoj. Při svých odhadech hojně využívá předstihových ukazatelů z výroby i repotů o stavu zásob surovin. Hodnotí trendy ve spotřebě a produkci, jak pro lokální úroveň, tak celosvětově.

Analýza vývoje měn kromě výše uvedených údajů bedlivě sleduje monetární politiku centrálních bank a predikuje její možné dopady. V současné době se za jedny z nejvýznamnějších faktorů monetární politiky považují základní úrokové sazby a nákupy/prodej dluhopisů centrálními bankami. Z makroekonomických ukazatelů se mezi nejdůležitější řadí údaje o zaměstnanosti a vládním zadlužení. (Veselá, 2011)

Fundamentální analýza aplikovaná při analýze akcií hodnotí informace o hospodářské situaci podniku. Využívá podnikových ukazatelů ziskovosti, zadluženosti, obrátkovosti a dalších. K těmto ukazatelům bere při svém hodnocení do úvahy pozici podniku oproti konkurentům v odvětví, citlivost odvětví na fáze hospodářského cyklu. Hodnotí dopady fiskální a hospodářské politiky, právní prostředí a mnoho dalších. (Rejnuš, 2014)

3.3 TECHNICKÁ ANALÝZA

Cílem technické analýzy je na základě historických dat z kapitálových trhů odhadnout zákonitosti v jejich vývoji a predikovat vývoj budoucí. Technická analýza nemá za cíl analýzu příčin daného stavu aktiva nebo trhu. Technický přístup vychází z předpokladu, že cena v daném okamžiku je odrazem veškerých informací, které cenu formují. Není potřeba žádných dalších informačních vstupů. (Rejnuš, 2014)

Základními nástroji technické analýzy jsou různé typy grafů v různých časových rámcích, indikátory, cenové hladiny neboli hladiny podpory a odporu, a cenových formací neboli patterny. Technická analýza předpokládá, že průběh cen finančních instrumentů je odrazem lidských reakcí na určité cenové situace, ve kterých se aktivum ocitá. Průběh reakcí na dané historické situace bude v zásadě stejný, protože chování účastníků trhu je v zásadě stále stejné. Technická analýza předpokládá, že je možné identifikovat situace, které se v historii objevily a cenové reakce, které je následovaly. Na základě historických zkušeností lze dle technické analýzy odhadnout budoucí vývoj.

3.3.1 Grafy

Ke své činnosti využívá technická analýza nejčastěji svíčkové, čárové a sloupcové grafy. Grafy mohou být zpracovány pro různé časové rámce. Svíčkový a sloupcový graf obsahují pro každý časový úsek (v našem případě jeden den) informaci o otevírací, zavírací, maximální a minimální ceně. Čárové grafy jsou stručnější, zobrazují pouze cenu při poslední kotaci v rámci časového úseku. Svíčkové grafy jsou východiskem pro teorii svíčkových patternů. Jedná se o různé formace ve svíčkovém grafu, u nichž se předpokládá schopnost potvrdit trend nebo jeho změnu případně předpokládané momentum trhu. Jedním z největších úskalí grafů pro vyšší časové rámce, než jedna hodina je jejich různost pro různá časová pásma. Je zřejmé, že grafy pro stejné trhy budou mít různý tvar v závislosti na časovém pásmu, pro které budou vykreslovány.



Graf 1: Příklad svíčkového grafu pro kurz amerického dolaru a japonského jenu. Červené svíce značí pokles kurzu dolaru vůči jenu, zelené naopak posílení. **Zdroj:** autor



Graf 2 : Příklad čárového grafu pro kurz amerického dolaru a japonského jenu. Graf zobrazuje pouze uzavírací ceny.
Zdroj: autor



Graf 3: Příklad sloupcového grafu pro kurz amerického dolaru a japonského jenu. Levá linka prezentuje otevírací cenu, zatímco pravá cenu zavírací. Rozpětí ceny spodním a horním okrajem ceny. **Zdroj:** autor

3.3.2 Technické Indikátory

Technické indikátory lze charakterizovat jako matematickou funkci, jejíž vstupní proměnnou je cena. Indikátory lze použít k podobným účelům jako cenové patterny v grafech. (Rejnuš 2014)

Díky jednoduchému zpracování pomocí výpočetní techniky si indikátory získaly velkou oblibu. V zásadě lze rozlišit indikátory na trendové, objemové a oscilátory. Základním trendovým indikátorem je klouzavý průměr. Oscilátory se snaží interpretovat stav ceny v rámci cenového rozpětí za určité období a své uplatnění nacházejí většinou v období, kdy je trh mimo trend. Objemové indikátory kromě přímé informace o objemu

uzavřených obchodů dávají do souvislosti objem a směr pohybu ceny. Výsledkem by měla být schopnost přesnějšího odhadu budoucího trvání trendu

3.3.2.1 Trendové indikátory

Používají se pro schopnost určit dlouhodobější směr, trend, pohybu trhu. Zda se jedná v daném okamžiku o rostoucí, klesající či pohyb do strany, někdy nazývaný boční trend. Nejběžnějšími trendovými indikátory jsou klouzavé průměry, které vyrovnávají průběh časové řady. Podle způsobu stanovení váhy cen z různých období rozlišujeme jednoduchý, vážený a exponenciálně vážený klouzavý průměr. Liší se v různé váze historických cen na hodnotu klouzavého průměru. Doplněním hodnot standardních odchylek ke klouzavému průměru získáváme Bollinger Bands indikátor. (Rejnuš 2014)

Grafy klouzavých průměrů lze použít samostatně nebo kombinaci více klouzavých průměrů. Klouzavé průměry nejenom samy o sobě pomáhají určit směr pohybu trhu, ale grafické překřížení klouzavých průměrů různých period je používáno jako signál potvrzení změny trendu na trhu.

Výpočet **jednoduchého klouzavého** průměru je výpočtem průměru hodnot za období n .

$$SMA_n = \frac{c_1 + c_2 + \dots + c_n}{n} \quad [3.3.2.1.1]$$

Rejnuš, 2014

n – počet časových úseků v periodě

c – cena v čase n periody

Výpočet **váženého klouzavého** průměru. Váha nejstaršího časového úseku je rovna jedné a o jednu se zvyšuje až k poslednímu – zmírňuje vliv nejstarších hodnot na hodnotu MA.

$$WMA_n = \frac{c_1 + 2c_2 + \dots + nc_n}{n + (n - 1) + \dots + 1} \quad [3.3.2.1.2]$$

Rejnuš, 2014

po úpravě

$$WMA_n = \frac{c_1 + 2c_2 + \dots + nc_n}{n \frac{(n + 1)}{2}} \quad [3.3.2.1.3]$$

n – počet časových úseků v periodě
 c – cena v čase n periody

Výpočet **exponenciálně váženého klouzavého průměru**. Všechny hodnoty jsou zahrnuty ve výpočtu. Výpočet se neomezuje pouze na n hodnot zvoleného období. Váhy nejsou lineární. Definujeme α , které určuje, jak velký význam pro hodnotu průměru mají nejstarší a nejnovější hodnoty.

$$EWMA_n = \alpha c_n + (1 - \alpha)EWMA_{n-1}$$

[3.3.2.1.4]

NIST/SEMATECH, 2012

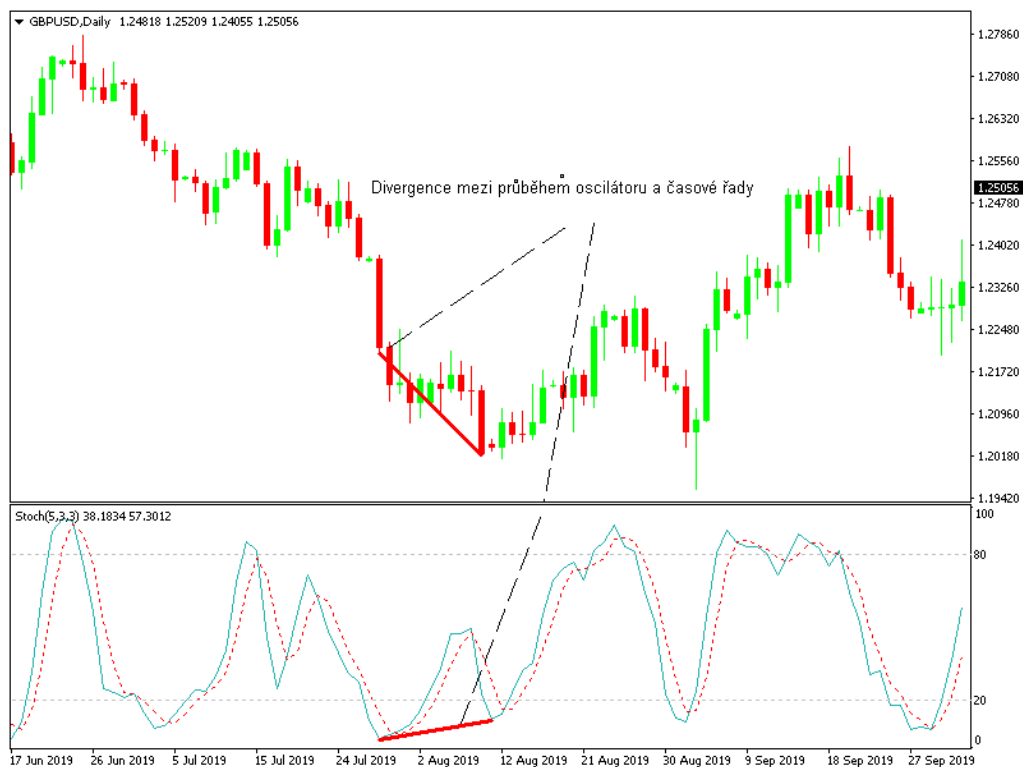
n – počet časových úseků v periodě
 c – cena v čase n periody

Za alfa běžně dosazujeme $\alpha = \frac{1}{m+1}$ kde $2m+1$ je délka klouzavých průměrů. (Cipra, 2013)

3.3.2.2 Oscilátory

Oscilátory jsou nejvíce používány v období, kdy je trh bez jasného rostoucího nebo klesajícího trendu. Jejich užití pomáhá odhadnout okamžik, kdy trh pohybuje se v rámci určitého rozpětí, dosahuje svých maxim, resp. minim a je pravděpodobné, že změní svůj směr. Využívá se samotných hodnot oscilátorů, grafického křížení aktuálních a průměrných hodnot indikátoru. Hodnoty oscilátorů se většinou pohybují v pásmu, nejčastěji 0-100. V době pohybu do trhu strany, hodnoty v horní části pásma signalizují překoupené hodnoty instrumentu a je obecně očekávaný pokles jeho ceny. Hodnoty v dolní části pásma bývají považovány za přeprodané a očekává se růst ceny.

Oblíbeným využitím oscilátorů je hledání rozdílů v průběhu oscilátoru a samotné ceny. Typické jsou divergence mezi průběhem indikátoru a cenového grafu. Stoupající hodnoty lokálních maxim časové řady a zároveň klesající hodnoty lokálních maxim oscilátoru. V grafu č.4 došlo k rozdílnému průběhu, divergenci, ceny instrumentu a hodnot oscilátoru. Zatímco cenový graf GBPUSD po 24. červenci 2019 pokračoval vytvářením nižších low, stochastik oscilátor (v dolním okně grafu) vytvořil v srpnu vyšší low než předchozí červencové. Tato situace je považována za silný signál pro budoucí růst ceny. Platí i obráceně, pokud cena vytváří vyšší cenové high a průběh oscilátoru nižší high, pak je očekávaný cenový pokles instrumentu.



Graf 4: Příklad divergence mezi cenovým grafem a oscilátorem. Divergence v průběhu cenového grafu a oscilátoru je považována za předzvěst změny trendu ceny Zdroj: autor

Výpočet **stochastického oscilátoru**. Nejdříve vypočteme hodnotu **%K**. Která transformuje aktuální cenu na procentní úroveň v rámci cenového rozpětí za zvolené období. Poté spočítáme **%D**, aritmetický průměr **%K** za vybrané období.

$$\%K = \frac{C_n - L_n}{H_n - L_n} \times 100 \quad [3.3.2.2.1]$$

Veselá, 2011

$$\%D = SMA_k(\%K) \quad [3.3.2.2.2]$$

Veselá, 2011

- n – délka periody sledovaného období
- C – závírací cena v čase n periody
- L – nejnižší cena v období
- H – nejvyšší cena v období
- k – délka periody pro výpočet aritmetického průměru v čase n
- SMA – aritmetický průměr

Výpočet **Relative strength index** RSI. Zpracovává informaci o vnitřní síle trhu porovnáváním kladných a záporných pohybů trhu za zvolené období.

$$RSI_n = 100 - \left[\frac{100}{1 + RS_n} \right] \quad [3.3.2.2.3]$$

Veselá, 2011

RSI_n – hodnota indikátoru pro čas n

RS_n – „Relative Strength“ – poměr mezi součtem kladných a součtem záporných změn (Rejnuš 2014), případně průměrnou kladnou a průměrnou zápornou změnou (Veselá 2011). Oba přístupy poskytují z definice stejný výsledek

n – délka sledovaného období

3.3.2.3 Volume indikátory

Objemové indikátory v sobě kromě informace o ceně zahrnují i informaci o realizovaném objemu. Výhodiskem při jejich návrhu je předpoklad, že největší objednávky realizované důležitými účastníky jsou směrodatné při dalším vývoji trhu.

Výpočet **OBV – On balance volume**. Pokud cena instrumentu uzavřela níže než předchozí den, odečítá se objem obchodů od přechozí hodnoty indikátoru. Pokud uzavřela výše, objem je k předchozí hodnotě OBV přičítaný. Pro první den výpočtu je OVB rovno objemu obchodů v daném dni. Aktuální hodnota indikátoru sama o sobě nemá význam a je do značné míry ovlivněna volbou počátečního data. K analýze se využívá průběhu indikátoru. Pokud se mění trend indikátoru na rostoucí, měly by akcie začít růst. Pokud se mění na klesající, měl by trh začít klesat. Indikátor je interpretovaný jako předzvěst pohybů na trhu. (Rejnuš, 2014)

Výpočet **MFI – Money Flow Index**. Indikátor v sobě kombinuje informaci o ceně a objemu obchodů. V zásadě se jedná konstrukci RSI oscilátoru doplněného o objem obchodů. Měl by generovat méně nákupních i prodejních signálů, poněvadž bere v úvahu i objemy obchodů.

$$MFI_n = 100 - \left[\frac{100}{1 + MR_n} \right] \quad [3.3.2.3.1]$$

$$MR_n = \frac{\textit{kladný tok peněz celkem}}{\textit{záporný tok peněz celkem}}$$

[3.3.2.3.2]

MFI_n – hodnota indikátoru v MFI pro období n

MR_n – „Money Ratio“ – poměr mezi kladným a záporným tokem peněz za období n

Kladný tok peněz – jestliže denní průměrná cena je vyšší než denní průměrná cena předchozího dne, pak denní kladný tok peněz je definovaný jako součin objemu a denní průměrné ceny

Využití MFI se podobá RSI. Lze reagovat na pohyb u horní a dolní meze pásma a reagovat na rozdíly v průběhu indikátoru a samotné ceny. Pokud cena roste a MFI klesá, jedná se o signál, že může dojít i k poklesu ceny a naopak.

3.3.2.4 Další prostředky technické analýzy

V technické analýze jsou dále hojně využívané různé svíčkové formace, identifikující možné změny ve směru trendu. Důležité hladiny odporu a rezistence, trendové linie a mnoho dalších. Často jsou využívány ve vzájemných kombinacích a analytici například sledují tvar svíce u některých z důležitých cenových hladin nebo trendových čar.

3.4 STATISTICKÉ METODY

3.4.1 Popisná statistika

Popisná statistika dává vyniknout charakteristickým rysům statistického souboru a umožňuje nám získat rychlý základní přehled o souboru.

Statistický soubor charakterizujeme pomocí průměrů a jemu podobných ukazatelů.

Nejčastěji používanými jsou aritmetický a vážený aritmetický průměr, modus a medián.

Důležitým parametrem, který spolu s průměrnou hodnotou souboru doplňuje

počáteční informace o souboru jsou rozptyl a standardní odchylka. Hodnoty

statistického souboru můžeme rozčlenit do kvantilů a s jejich pomocí sestrojít

histogram, pomocí hustoty a distribuční funkce předběžně odhadnout o jaké rozdělení

se jedná. (Hindls, 2004)

3.4.2 Náhodná veličina

Náhodná veličina je diskrétního nebo spojitého charakteru. Náhodné veličiny definujeme distribuční funkcí. Základními popisnými charakteristikami náhodných veličin jsou střední hodnota, rozptyl, medián, modus a další.

3.4.2.1 Charakteristiky náhodných veličin

Střední hodnota diskrétní veličiny

$$\mathbb{E}X = \sum_{i=1}^{\infty} x_i \cdot p_i \quad [3.4.2.1.1]$$

Mrkvička, 2008

$\mathbb{E}X$ - střední hodnota diskrétní náhodné veličiny
 x_i - reálné hodnoty náhodné veličiny
 p_i - pravděpodobnost hodnoty reálné veličiny

Střední hodnota spojitě náhodné veličiny za předpokladu, že integrál existuje.

$$\mathbb{E}X = \int_{-\infty}^{\infty} x f_x(x) dx \quad [3.4.2.1.2]$$

Mrkvička, 2008

f_x - hustota spojitě náhodné veličiny

Rozptyl

$$\text{var } X = \mathbb{E}(X - \mathbb{E}X)^2 \quad [3.4.2.1.3]$$

Mrkvička, 2008

var X - rozptyl

Směrodatná odchylka

$$s = \sqrt{\text{var } X} \quad [3.4.2.1.4]$$

Mrkvička, 2008

s - směrodatná odchylka

Medián \tilde{x} ,

$$P(X \leq \tilde{x}) \geq \frac{1}{2} \text{ a } P(X \geq \tilde{x}) \geq \frac{1}{2} \quad [3.4.2.1.5]$$

Mrkvička, 2008

Modus \hat{x} , hodnota s nejvyšší četností v náhodném prostoru.

$$f(\hat{x}) \geq f(x), \forall x \in (-\infty, \infty) \quad [3.4.2.1.6]$$

Mrkvička, 2008

Normální rozdělení je spojitě rozdělení s mimořádným významem v teorii pravděpodobnosti a matematické statistice.

Lze definovat pomocí funkcí hustoty

$$f(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \quad [3.4.2.1.7]$$

Mrkvička, 2008

kde μ reálné a σ^2 kladné jsou parametry

Kovariance

$$\text{cov}(X, Y) = \mathbb{E}(X - \mathbb{E}X)(Y - \mathbb{E}Y) \quad [3.4.2.1.8]$$

Anděl, 2019

$\text{cov}(X, Y)$ - kovariance veličin X a Y, zapisovaná také jako σ_{XY}

Korelační koeficient vyjadřuje sílu lineárního vztahu mezi dvěma náhodnými veličinami.

$$\rho = \frac{\sigma_{XY}}{\sqrt{\sigma_x^2 \sigma_y^2}} \quad [3.4.2.1.9]$$

Anděl, 2019

ρ - korelační koeficient

σ_{XY} - kovariance X, Y

σ_X - var X, σ_Y - var Y

3.4.3 Příklad použití základní popisné statistiky na vzorovém souboru závěrečných měsíčních cen indexu SP500 let 2008-2010

Soubor obsahuje 36 hodnot.

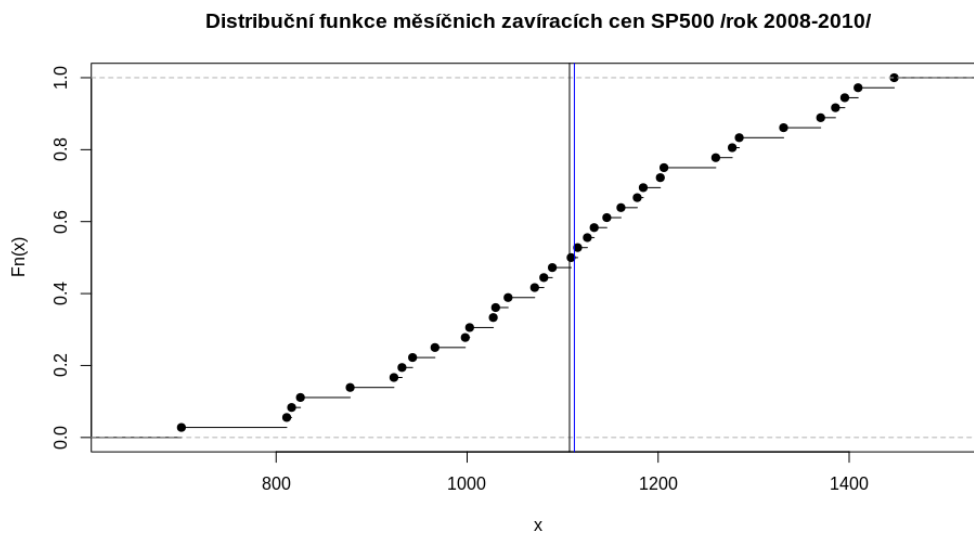
Průměrná měsíční závěrečná cena: 1107.2

Medián (prostřední) měsíční závěrečná cena: 1112.3

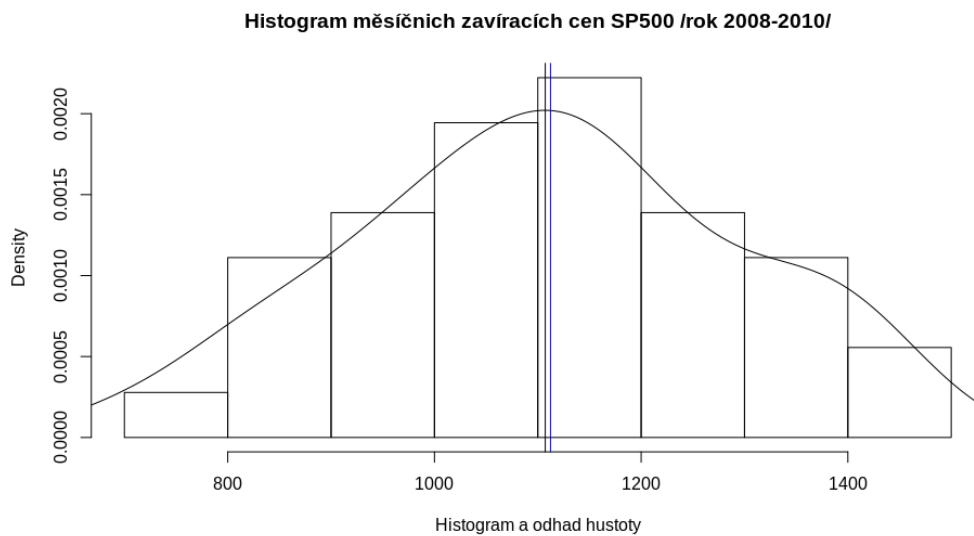
Minimální měsíční závěrečná cena: 700.8

Maximální měsíční závěrečná cena: 1447.2

Standardní odchylka měsíční závěrečné ceny: 187.39



Obrázek 1: Příklad distribuční funkce



Obrázek 2: Příklad histogramu a rozdělení hustoty pravděpodobnosti

Podle histogramu a odhadu hustoty pravděpodobnosti lze na první pohled konstatovat, že soubor hodnoty průměru a mediánu jsou blízké, histogram má jeden významný vrchol a je vcelku souměrný. Pravděpodobně by bylo možné testovat, zda rozdělení odpovídá normálnímu a je vhodné pro další běžné testy založené na normálním rozdělení.

Nejčastějším a ve statistické analýze nejvíce využívaným rozdělením je normální rozdělení. Graf hustoty normálního rozdělení odpovídá Gaussově křivce.

3.4.4 Testování normality

Testování normality rozdělení statistického souboru slouží mimo jiné jako výchozí bod k rozhodnutí o metodách použitých při dalším testování.

Prvním krokem k posouzení normality bývá histogram statistického souboru proložený křivkou normálního rozdělení, hodnota průměru a mediánu. Pokud má rozdělení jeden vrchol, hodnota průměru a mediánu je přibližně stejná a histogram se tvarem blíží normálnímu rozdělení, je pravděpodobné, že rozdělení bude mít charakteristiky rozdělení normálního.

V dalším kroku je vhodné sestavit Q-Q graf, kdy na jednu z os vyneseme kvantily zkoumaného rozdělení a na druhou osu normálního rozdělení. Pokud je námi zkoumané rozdělení normální, bude na Q-Q grafu přímka.

V případě potřeby je možné použít statistické testy normality, například regresní **Shapiro-Wilkův** test normality pro soubory od jednotek hodnot. Nulovou hypotézou testu je výrok, že testované rozdělení je normální. V praktickém použití v statistickém software **R** odmítneme nulovou hypotézu, pokud p-value testu je nižší než zvolená hladina jeho spolehlivosti.

3.4.5 Testy střední hodnoty

Testy středních hodnot umožňují porovnávat střední hodnoty souborů vůči zvolené hodnotě, případně mezi různými statistickými soubory. Volba testu závisí na počtu výběrů, normalitě a závislosti výběru.

Pokud se jedná o jednovýběrové normální rozdělení, používáme jednovýběrový t-test. V případě jednovýběrového rozdělení, které není normální, používáme jednovýběrový Wilcoxon test.

U dvou nezávislých výběrů v případě normality dat použijeme párový t-test. Pokud normalita přítomna není, použijeme párový Wilcoxonův test. Pokud jde o dva výběry, které nejsou nezávislé, použijeme u normálního rozdělení F-test.

V případě tří a více výběrů využíváme analýzu rozptylu (ANOVA).

3.5 INVESTIČNÍ INSTRUMENT, PORTFOLIO

Základními charakteristikami investičního instrumentu jsou výnos, riziko a likvidita.

Výnos je schopnost instrumentu poskytovat v čase zisk. Likvidita je schopnost přeměny instrumentu na hotovost a riziko představuje pravděpodobnost odchylky od očekávaného výnosu. Při rozhodování o investici se investoři v rámci svých preferencí snaží maximalizovat výnos, minimalizovat riziko a dosáhnout co nejvyšší likvidity.

V praxi stojí výnos, riziko a likvidita často na opačných stranách a vzájemně se vylučují.

Neexistuje instrument, u kterého by bylo možné dosáhnout vysokého výnosu s nulovým rizikem a okamžitou likviditou. (Veselá, 2011)

Výnos běžně definujeme jako procentní změnu mezi hodnotou instrumentu v minulosti a současnou hodnotou. Do výnosu zahrnujeme jak rozdíl mezi nákupní a prodejní cenou, tak případné náklady a výnosy, které souvisí s držetím instrumentu. Výnosy mohou být například dividendy, úroky, kupóny. Typickým nákladem jsou daně a různé poplatky.

Důležitou hodnotou, bodem zlomu, je vnitřní výnosové procento investice. Definuje, jaká hodnoty musí být úroková míra investice, aby čistá současná hodnota investice byla rovna nule.

$$0 = \sum_{t=1}^n \frac{CF_t}{(1 + IRR)^t} - I \quad [3.5.1.1.1]$$

Vochozka, 2012

IRR vnitřní výnosové procento investice

CF_t čistý peněžní příjem v nominálním vyjádření realizovaný v roce t

n doba životnosti projektu v letech

I vynaložený investiční výdaj

Riziko definujeme jako standardní odchylku výnosu v dílčích periodách sledovaného období od průměrného výnosu za celé období. Riziko rozdělujeme na systematické (tržní) dané vnějšími politicko-ekonomickými vlivy a riziko jedinečné spjaté s podnikatelským záměrem, managementem a dalšími specifiky souvisejícími s instrumentem. Zatímco systematické riziko volbou instrumentu neovlivníme, riziko nesystematické ovlivnit lze.

Citlivost výnosové míry instrumentu na pohybu tržní výnosové míry nazýváme **beta faktor**. Beta faktor zohledňuje intenzitu reakce instrumentu na změnu tržní výnosové míry. (Veselá, 2011)

$$\beta_{Si} = \frac{Cov(r_{Si}, r_m)}{(\sigma_m)^2} = \frac{\sigma_{Si}}{\sigma_m} * \rho_{Si,m} \quad [3.5.1.1.2]$$

Veselá, 2011

β_{Si} beta faktor instrumentu i

$Cov(r_{Si}, r_m)$ kovariance mezi výnosovou měrou instrumentu a výnosovou měrou tržního portfolia

$(\sigma_m)^2$ rozptyl výnosových měr tržního portfolia, reprezentovaných tržním indexem

σ_{Si} směrodatná odchylka výnosových měr z instrumentu

$\rho_{Si,m}$ korelační koeficient mezi výnosovou měrou z instrumentu a výnosovou měrou z tržního portfolia

3.5.2 Měření výkonnosti portfolia

Z hlediska investora je důležité mít možnost porovnat portfolia a rozhodnout se na základě objektivních kritérií pro nejvíce vyhovující portfolio. Hlavními parametry výkonnosti portfolia jsou výnos a riziko.

Historický výnos portfolia definujeme jako součet vážených průměrných výnosů instrumentů v portfoliu. Vahou jsou podíly jednotlivých instrumentů na tržní hodnotě portfolia. (Veselá, 2011)

$$r_p = \sum_{n=1}^N r_{aexp} \cdot X_o \quad [3.5.2.1.1]$$

Veselá, 2011

r_p celková historická výnosová míra portfolia

r_{aexp} průměrné historické výnosové míry jednotlivých instrumentů v portfoliu

X_o váhy (podíly) jednotlivých instrumentů na celkové tržní hodnotě portfolia

N počet instrumentů v portfoliu

Celkové riziko portfolia je ovlivněné rizikem jednotlivých složek, jejich vahou v portfoliu, stupněm závislosti výnosů mezi instrumenty v portfoliu a počtem instrumentů.

Obecný vztah pro výpočet směrodatné odchylky výnosů portfolia lze definovat:

$$\sigma_p = \sqrt{\sum_{i=1}^N \sum_{j=1}^N X_i * X_j * \sigma_{ij}} \quad [3.5.2.1.2]$$

Veselá, 2011

σ_p směrodatná odchylka portfolia jako míra celkového rizika portfolia

N počet instrumentů v portfoliu

X_i váha i. instrumentu portfolia

X_j váha j. instrumentu portfolia

σ_{ij} kovariance mezi instrumenty i a j portfolia

3.5.3 Dvoudimenzionální metody měření portfolia

Z hlediska investora je důležité mít možnost porovnat různá portfolia a rozhodnout se na objektivních kritériích a vlastních referencích pro nejvíce vyhovující portfolio. Hlavními parametry výkonosti portfolia jsou výnos a riziko. K měření výkonosti portfolia lze využít jednoduchých jednodimenzionálních metod, které pracují pouze s výnosovou mírou bez zohlednění rizika.

Spojené hodnocení výnosu a rizika nacházíme ve dvoudimenzionálních metodách.

Jedním z nejnámějších ukazatelů, který porovnává výnosnost portfolia vůči zvolené referenční úrokové míře se nazývá Sharpeho index. (Veselá, 2011)

$$I_{Sh} = \frac{r_{portf} - R_f}{\sigma_p} \quad [3.5.3.1.1]$$

Veselá, 2011

I_{Sh} - hodnota Sharpeho indexu

r_{portf} - historická výnosová míra dosažená portfoliem

R_f - bezriziková výnosová míra

σ_p - celkové riziko portfolia měřené směrodatnou odchylkou, tedy variabilitou výnosové míry portfolia v minulosti

Sharpeho index penalizuje ve své hodnotě kromě záporných výkyvů i kladné výkyvy ve výnosnosti portfolia. Proto došlo postupem času k jeho inovaci a Sortinův index dává do jmenovatele „negativní směrodatnou“ odchylku vypočtenou jako běžná standardní odchylka s použitím pouze záporných odchylek od průměrného výnosu.

$$I_{Sort} = \frac{r_{portf} - R_f}{\sigma_{\frac{p}{neg}}} \quad [3.5.3.1.2]$$

Veselá, 2011

I_{Sort} - hodnota Sortinova indexu

r_{portf} - historický výnosová míra dosažená portfoliem

R_f - bezrizikový výnosová míra

$\sigma_{\frac{p}{neg}}$ - celkové riziko portfolia měřené standardní odchylkou, nicméně zachycující pouze negativní odchylky od střední hodnoty

Sharpeho i Sortinův index pracují s celkovým rizikem portfolia. Nerozlišují mezi systematickými a jedinečnými riziky. Neposkytují informaci o absolutním výkonu portfolia, ale umožňují porovnat dvě portfolia mezi sebou.

4 PRAKTICKÁ ČÁST

4.1 Úvod

V praktické části chci zjistit, zda lze využitím vybraných metod technické analýzy dosáhnout ziskového řízení zvoleného portfolia.

Praktická část je rozdělena do pěti fází.

- Stanovení portfolia a definování strategie pomocí technické analýzy
- Volba technických prostředků a software
- Optimalizace modelu technické analýzy a out-of-sample analýza
- Zhodnocení získaných dat

4.2 STANOVENÍ PORTFOLIA A DEFINICE STRATEGIE

Portfolio se skládá ze dvou menších. Portfolia akciového a portfolia pro měnové spekulace. Váhy mezi menšími portfolii jsem rozdělil v poměru 1:1. První, zaměřené na obchody s akciemi disponovalo na začátku sledovaného období vkladem \$5. Druhé, určené pro spekulace na měnovém trhu disponovalo také vkladem \$5. Každá z nově otevřených pozic v rámci portfolia, měla v době otevření vždy hodnotu \$1.

Management akciového portfolia sestával z řízení nákupních a prodejních pozic na 27 akcií vybraných náhodně z indexu SP500 a NASDAQ. Management portfolia měnových spekulací sestával z řízení nákupních a prodejních pozic 51 předem náhodně vybraných měnových párů. Pro výběr jsem stanovil dvě hlavní technická kritéria. Za prvé dostupnost souvislé časové řady obsahující informace o denní otevírací a uzavírací ceně instrumentu za každý obchodní den. Druhým kritériem byla možnost ocenění hodnoty instrumentu v USD. Ocenění mohlo být provedeno buď přímo ve zdrojových datech nebo pomocí kurzu měny, ve které je instrument oceněný a USD.

Za zdroj kotací všech vybraných instrumentů jsem si zvolil web, resp. API finance.yahoo.com.

4.2.1 Akciové portfolio

V rámci akciového portfolio byly řízeny pozice na následujících akciových titulech:

ADM	Archer-Daniels-Midland Company
ADP	Automatic Data Processing, Inc.
AEE	Ameren Corporation
AIG	American International Group, Inc.
AMAT	Applied Materials, Inc.
BK	The Bank of New York Mellon Corporation
C	Citigroup Inc.
CAH	Cardinal Health, Inc.
CNP	CenterPoint Energy, Inc.
CSCO	Cisco Systems, Inc.
EIX	Edison International
FDX	FedEx Corporation
GL	Globe Life Inc.
IPG	The Interpublic Group of Companies, Inc.
KEY	KeyCorp
KMB	Kimberly-Clark Corporation
LUV	Southwest Airlines Co.
MAS	Masco Corporation
MSI	Motorola Solutions, Inc.
PCAR	PACCAR Inc
PNC	The PNC Financial Services Group, Inc.
STT	State Street Corporation
SWK	Stanley Black & Decker, Inc.
TFC	Truist Financial Corporation
VMC	Vulcan Materials Company
WMT	Walmart Inc.
XRX	Xerox Holdings Corporation

Tabulka 1: Instrumenty akciového portfolia

4.2.2 Portfolio měnových spekulací

V rámci portfolia měnových spekulací byly řízeny pozice na měnových párech následujících měn:

AUD	Australian dollar
BRL	Brazilian real
CAD	Canadian dollar
DKK	Danish krone
EUR	Euro
GBP	Pound sterling

HKD	Hong Kong dollar
CHF	Swiss franc
INR	Indian rupee
JPY	Japanese yen
KRW	South Korean won
MXN	Mexican peso
NOK	Norwegian krone
NZD	New Zealand dollar
RUB	Russian ruble
SGD	Singapore dollar
TRY	Turkish lira
USD	US dollar
ZAR	South African Rand

Tabulka 2: Měny portfolia měnových spekulací

4.3 DEFINOVÁNÍ STRATEGIE POMOCÍ TECHNICKÉ ANALÝZY

V rámci analýzy jsem chtěl zpracovat nejen trhy z různých geografických částí a různých instrumentů, ale i z různých časových úseků. Trh vykazuje v čase různé charakteristiky. V průběhu let se střídají období poklesu a růstu, které mají rozdílnou dynamiku a trvání. Za sledované období jsem zvolil roky 2008-2019 s cílem obsáhnout období krize roku 2008, růstu i bočního trendu.

Jedním z nejobtížnějších kroků bylo zvolení nástrojů technické analýzy. Nabízí se nepřehledné množství metod, přístupů a jejich vzájemných mixů. Zvažoval jsem, že trhy se neustále mění a není možné používat stále stejnou kombinaci nástrojů s pevnými parametry. Vzhledem k velkému množství zpracovávaných dat bylo brzy zřejmé, že bude nutné použít automatizované zpracování. Pro automatizaci se jevílo nejvhodnější využít indikátorů technické analýzy, jejichž výhodou je formalizovaná definice. Nezahrnul jsem do zkoumání svíčkové formace, ani supporty a resistance či vyhledávání patternů v časových řadách.

Při analýze jsem aplikoval v zásadě velmi jednoduchou, obecně prezentovanou **trendovou strategii** využívající jednoduchých klouzavých průměrů a stochastického oscilátoru. Určité konstelace **klouzavých průměrů** signalizují rostoucí nebo klesající trend a aktuální průběh **stochastického oscilátoru** signalizuje trend krátkodobější. Při

souladu odhadu dlouhodobého a krátkodobého trendu signalizuje strategie vstup do pozice. Při nesouladu signalizuje výstup.

Vstup do pozice:

- Pokud klouzavé průměry signalizují růstový trend a zároveň stochastický oscilátor začíná opouštět pásmo přeprodanosti, otevři dlouhou pozici.
- Pokud klouzavé průměry signalizují klesající trend a zároveň stochastický oscilátor začíná opouštět překoupené pásmo, otevři krátkou pozici.

Výstup z pozice:

- Pozici zavři, pokud klouzavé průměry signalizují opačný trend, než byl trend při otevření pozice.
- Pozici zavři, pokud stochastický oscilátor ukazuje opačný trend, než pro který byla otevřena pozice
- Pozici zavři, pokud jsou splněny podmínky pro otevření pozice v opačném směru

Předchozí definice je srozumitelná, zároveň však příliš obecná pro skutečné testování a bylo nutné ji objektivně specifikovat. Jaké periody mají být použity u klouzavých průměrů, jaké periody pro stochastický oscilátor? Dokonce je nutné specifikovat váhu jednotlivých indikátorů při rozhodování. Má být pozice otevřena při vhodné konstelaci klouzavých průměrů a můžeme ignorovat stochastický indikátor nebo naopak? Nebo je snad nutné stanovit váhy jednotlivým indikátorům při rozhodování?

Při neustále měnících se trzích je neefektivní používat stále stejné parametry strategie, optimalizované pro určitou historickou situaci. Chtěl jsem co nejdříveji simulovat reálný postup použití technické analýzy. Volbu přesných parametrů a vah indikátorů technické analýzy použitých ve strategii, jsem se rozhodl **optimalizovat vždy na předchozích 15 měsících** historických denních dat a **získané parametry** pak ve strategii používat **v následujících dvou měsících** „out of sample“ analýzy, které vygenerovaly vstupy a výstupy do trhu.

Optimalizace parametrů technické analýzy pro každý z instrumentů začaly za období 1.10.2006-31.12.2007. Po optimalizaci parametrů indikátorů a jejich vah následovalo

vygenerování out-of-sample pozic za období 1.1.2008-28.2.2008 a jejich uložení k dalšímu testování. Následně se celý cyklus posunul o dva měsíce. Proběhla optimalizace v období 1.1.2007-28.2.2008 opět následovaná vygenerování pozic v out of sample analýze za období 1.3.2008-30.4.2008. Celý skončil out-of-sample analýzou za období 1.10.2019-31.12.2019 a opakoval se identicky pro každý z instrumentů v portfoliu.

4.4 TECHNICKÉ PROSTŘEDKY

Bylo nutné zvolit vhodné softwarové prostředky pro získání dat, jejich kontrolu a uložení. Dále navrhnout nebo využít software pro optimalizaci, resp. trénování strategie na historických datech a následně pro aplikaci vytrénovaného modelu na out-of-sample datech. Nakonec bylo potřeba data vygenerovaná modelem vyhodnotit statisticky a z hlediska potenciální finanční úspěšnosti.

Za zdroj dat jsem zvolil službu `finance.yahoo.com`. Služba je bezplatně k dispozici široké veřejnosti a data z ní jdou získávat nejen ručně stažením dat ze stránek, ale i skrze API. Pro stažení dat jsem napsal krátký script v **Python3** s využitím knihovny **yfinance** (příloha 9.1).

Získaná data jsem ukládal do open-source databáze **MySQL**. Při zpracovávání dat jsem zjistil, že kotace čas od času vykazují chyby. Například kurz dánské koruny se v některých dnech změnil stonásobně apod. Aplikoval jsem proto filtr, který mě upozornil, pokud u některého titulu byl aktuální denní kurz odlišný od průměru za předcházející tři dny více než 15%. Odchylky kurzů větší než 15% od 3 denního průměru jsem individuálně zkontroloval a posoudil, zda se jedná o chybu ve zdroji dat nebo zda skutečně došlo k prudkému výkyvu instrumentu (příloha 9.3).

Pro trénování modelu jsem zvolil cestu napsání vlastního scriptu, opět v Python3 za využití běžných knihoven (**Pandas** a dalších). Dobu potřebnou pro trénování modelu při velmi hrubých iteracích parametrů v jedno vláknovém zpracování jsem odhadoval na přibližně tři měsíce strojového času. Proto jsem se rozhodl využít cluster vytvořený pomocí **Ray** knihovny, opět pro Python3. Do clusteru jsem v domácích podmínkách zařadil 14 CPU jader a zkrátil dobu, potřebnou pro trénink na historických datech a pro out-of-sample testy, na přibližně deset dní. Dalšího zrychlení by bylo možné dosáhnout zařazením dodatečných CPU jader, případně úpravou kódu pro využití jednotek GPU, které obsahují milióny jader. V neposlední řadě by bylo vhodné optimalizovat samotný Python script. (příloha 9.2)

Parametry modelu získané tréninkem na historických datech jsem opět ukládal do MySQL databáze. Dlužno podotknout, že od původně jednodiskového domácího databázového serveru jsem se, opět z výkonnostních důvodů, postupem času přesunul

k raid5 diskovému poli tvořenému čtyřmi běžnými SSD disky. Samotná tabulka vygenerovaných hodnot parametrů obsahuje 180.252.660 záznamů a během optimalizace a následné out of sample analýzy bylo vygenerováno 858 pozic, jejichž výsledky jsou datovou bází pro další analýzu.

Vygenerované pozice, nebyly vzájemně srovnatelné, pokud by byly ohodnoceny v různých měnách a objemech. Pro další hodnocení jsem výsledky v prvním kroku standardizoval převodem na USD, následně převedl objemy na jednotky odpovídající vždy \$1 při vstupu do pozice. Převodem na jednotnou měnu a jednotný objem pozic jsem získal standardizovaná, vzájemně porovnatelná data. Standardizaci a sumarizaci jsem opět provedl v MySQL.

Samotná statistická analýza byla provedena v prostředí **RStudio** rozšířeném o knihovny z **CRAN** repositáře a pomocí konektoru napojené na datové úložiště v MySQL. Výpočet ukazatelů finančního zhodnocení jsem vyhotovil v open-source **LibreOffice Calc**.

Pro některé z kroků mezi jednotlivými fázemi jsem použil bash skripty. Celé prostředí počínaje sběrem dat, přes optimalizace až po statistické testování a výpočet ukazatelů portfolia bylo provozováno v open-source operačním systému Ubuntu 18.04.

4.5 PŘÍKLADY VYGENEROVANÝCH POZIC

Při optimalizaci strategie na historických datech byly hledány dva jednoduché klouzavé průměry, jejichž překřížení by signalizovalo nový trend. Periody klouzavých průměrů byly hledány v rozmezí 140-320 dní pro první a 50-150 dní pro druhý průměr. Dále bylo zjišťováno, zda je nutné brát v úvahu stochastický oscilátor nebo lze pozici řídit pouze klouzavým průměrem. Pokud bylo nutné hledat vhodný Stochastický oscilátor, pak byly vyhledávány periody křivek v rozmezí 9-27 dní pro %K a 3-9 pro %D.

Pásma přeprodanosti a překoupenosti Stochastického oscilátoru byla zadána fixně na úrovni 25 a 75.

Přesnou specifikaci pro otevření a ukončení pozice lze najít v příloze ve scriptu Optimize.v.102.py ve třídě Analysis, proceduře CalcPositions. Třída Optimize, procedure DoOptimize iteruje nad historickými daty a optimalizuje nastavení indikátorů pro další out-of-sample období.

4.5.1 Příklad pozice na instrumentu AIG

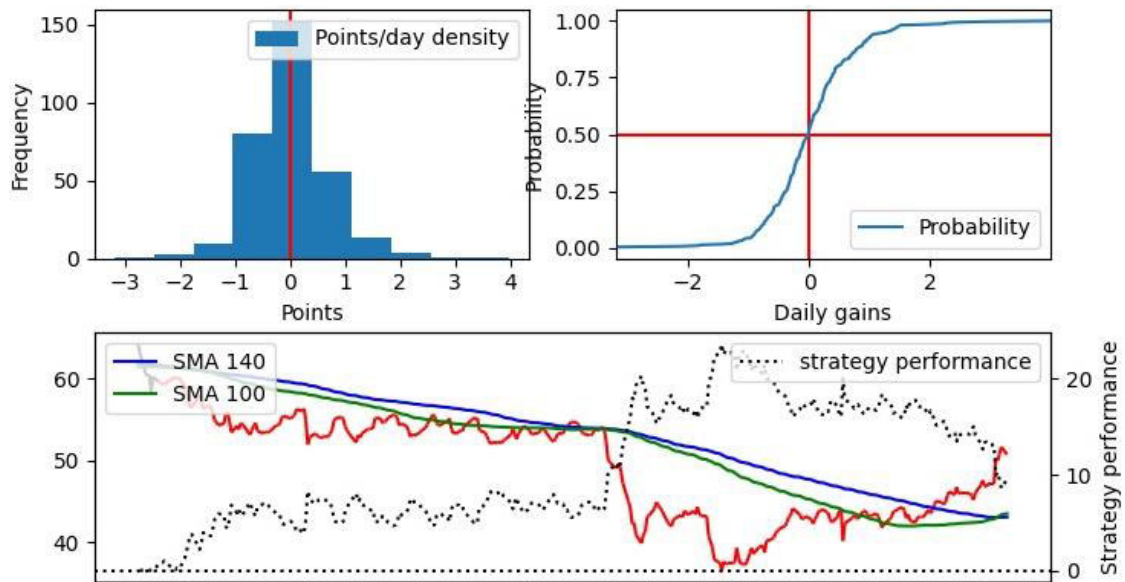
Kód AID značí akcie American International Group, Inc. obchodované v USD /NYSE

Optimalizace za období 1.11.2016-31.1.2018 vygenerovala následující nastavení strategie:

Parametr	Hodnota
Perioda prvního klouzavého průměru	140
Perioda druhého klouzavého průměru	100
Váha signálu klouzavých průměrů pro uzavření pozice	100/100
Váha signálu klouzavých průměrů pro otevření pozice	100/100
Stochastic %K	3
Stochastic %D	9
Váha stochastického oscilátoru pro uzavření pozice	40/100
Váha stochastického oscilátoru pro otevření pozice	40/100

Tabulka 3: Tabulka parametrů indikátorů technické analýzy pro AIG v období 1.2.2018-31.3.2018

V následujícím období 1.2.2018-31.3.2018 out-of-sample analýzy byla otevřena 12.2.2018 v 0:00 prodejní pozice na ceně 59.61, která byla uzavřena 13.5.2019 na ceně 50.72 se ziskem 9.5 bodů.



Graf 5: Příklad grafického výstupu třídy Analysis, pozice na akcii AIG; zdroj: autor

Graf 5 je rozdělený na tři části. V levém horním rohu je vidět histogram denních přírůstků pozice na AIG za období, kdy byla pozice otevřena. Pravý horní graf ukazuje distribuční funkci denních přírůstků pozice. Ve spodním grafu lze sledovat křivky jednoduchých klouzavých průměrů s periodami 140 a 100 dní, červenou čáru prezentující denní uzavírací ceny akcie AIG na NYSE a černá přerušovaná čára prezentuje změnu hodnoty akcie od doby otevření pozice. Standardizovanou hodnotu pozice přepočtem na jeden dolar při otevření by bylo možné dopočítat vydělením hodnoty pozice koeficientem 59.61.

4.5.2 Příklad pozice na instrumentu DKKUSD=X

Kód DKKUSD=X značí kolik USD stojí jedna DKK

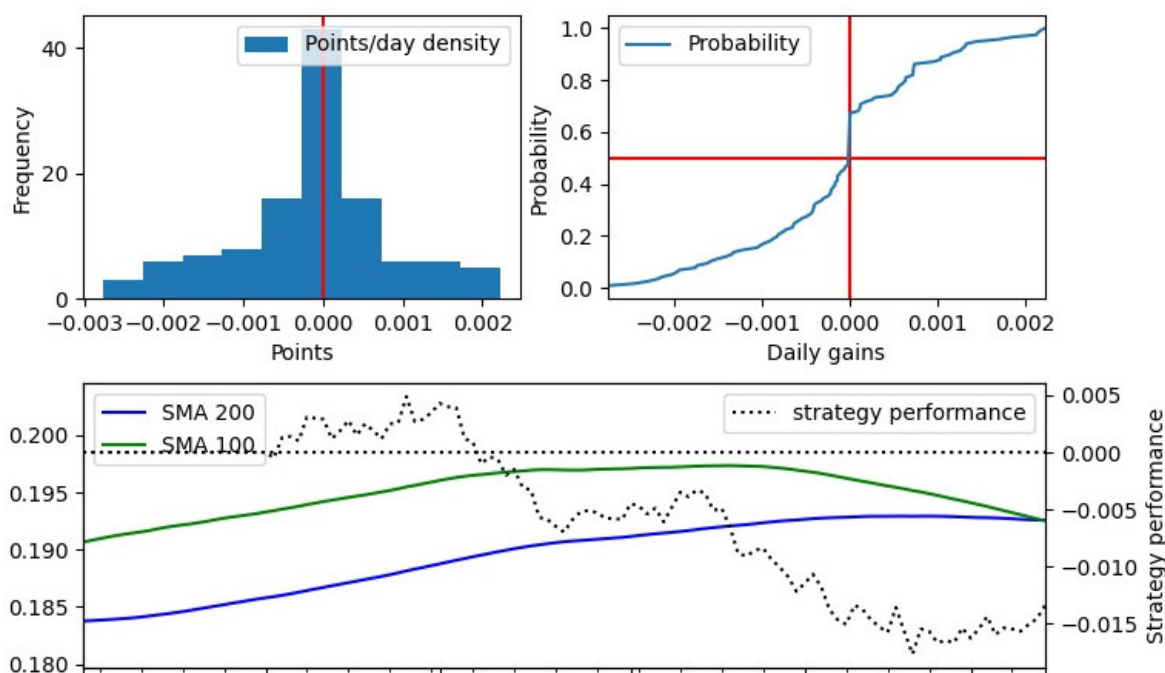
Optimalizace za období 1.7.2008-30.9.2009 vygenerovala následující nastavení strategie:

Parametr	Hodnota
Perioda prvního klouzavého průměru	200
Perioda druhého klouzavého průměru	100

Váha signálu klouzavých průměrů pro uzavření pozice	100/100
Váha signálu klouzavých průměrů pro otevření pozice	70/100
Stochastic %K	6
Stochastic %D	9
Váha stochastického oscilátoru pro uzavření pozice	40/100
Váha stochastického oscilátoru pro otevření pozice	70/100

Tabulka 4: Tabulka parametrů indikátorů technické analýzy pro DKKUSD v období 1.10.2009-30.11.2009

V následujícím období 1.10.2009-30.11.2009 out-of-sample analýzy byla otevřena 3.11.2009 v 0:00 nákupní pozice na ceně 0.1985, která byla uzavřena 12.3.2010 na ceně 0.1851 se ztrátou 0.0134 bodů.



Graf 6: Příklad grafického výstupu třídy Analysis, pozice na akcii AIG; zdroj: autor

Graf 6 je rozdělený na tři části. V levém horním rohu je vidět histogram denních přírůstků pozice na měnovém páru DKKUSD za období, kdy byla pozice otevřena. Pravý horní graf ukazuje distribuční funkci denních přírůstků pozice. Ve spodním grafu lze sledovat křivky jednoduchých klouzavých průměrů s periodami 200 a 100 dní a černá přerušovaná čára prezentuje změnu hodnoty pozice od doby otevření pozice. Pozice byla otevřena na základě informace o růstovém trendu definovaném klouzavými

průměry a hodnoty Stochastického oscilátoru (graf jeho průběh nezachycuje). Uzavřená pozice byla v okamžiku, kdy klouzavé průměry svým překřížením signalizovaly změnu trendu z rostoucího na klesající.

4.6 TESTY PORTFOLIÍ

Veškeré statistické testy v této kapitole jsou prováděny na hladině významnosti $\alpha = 5\%$.

4.6.1 Akcie

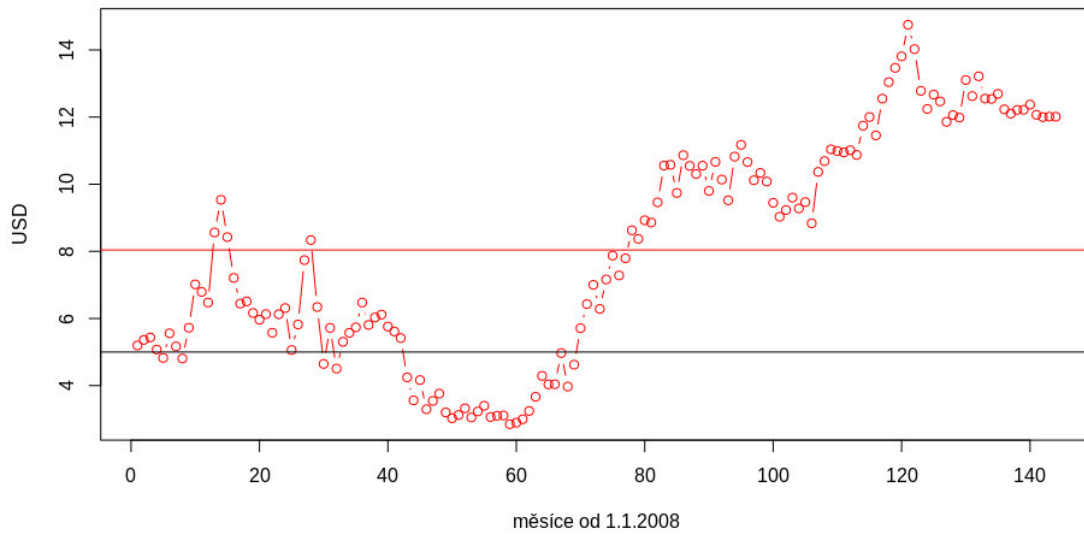
Akciové portfolio mělo k 1. lednu 2008 hodnotu \$5. Po dobu dvanácti let byla vždy na konci měsíce zaznamenána aktuální hodnota portfolia a uložena. Po skončení období jsem provedl analýzu s následujícím výsledkem:

Parametr	Hodnota
Počáteční hodnota	5.00
Minimum	2.85
Maximum	14.75
Průměr	8.04
Medián	7.832
Standardní odchylka	3.33

Tabulka 5: Základní statistické parametry akciového portfolia

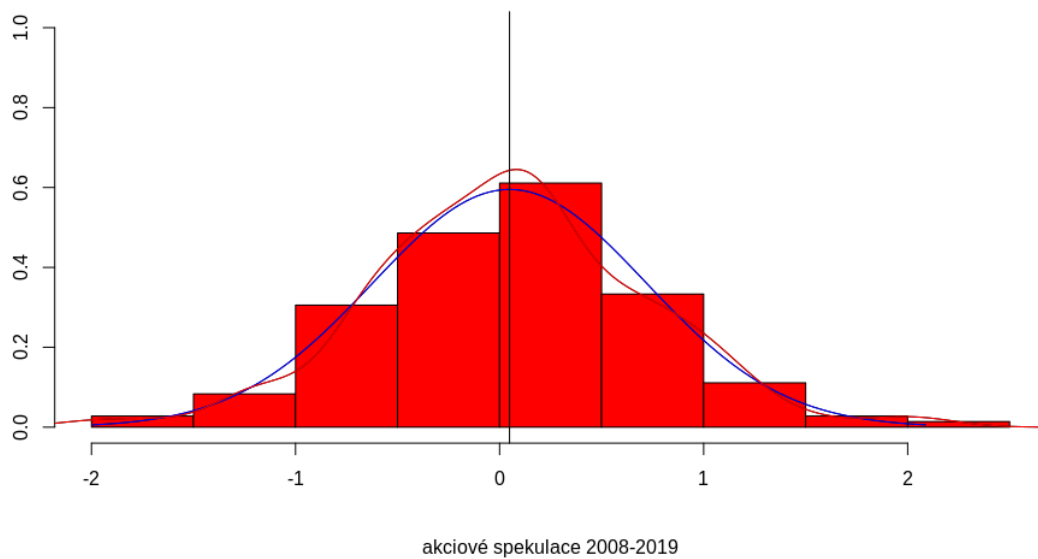
Vývoj akciového portfolia po jednotlivých měsících prezentuje následující graf.

Vývoj akciového portfolia 2008-2019



Graf 7: Vývoj akciového portfolia v letech 2008-2019. Černá linka \$5 značí počáteční stav, červená průměrný stav.

Histogram měsíčních změn portfolia



Graf 8: Histogram měsíčních přírůstků akciového portfolia, odhad hustoty rozdělení a odpovídajícího normálního rozdělení

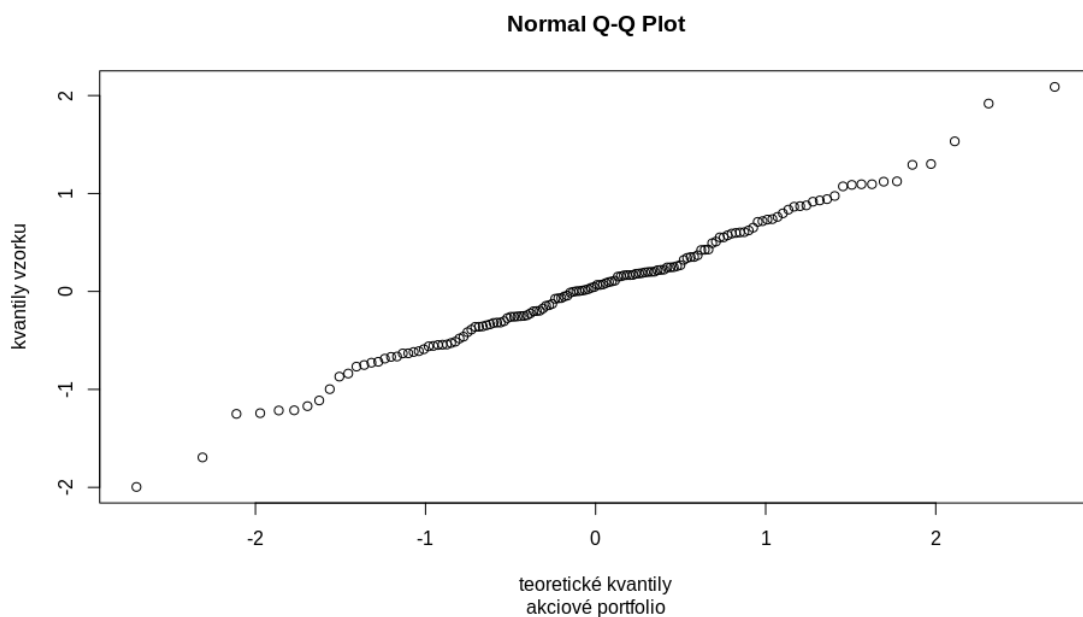
Na grafu č. 8 je zakreslený histogram denních přírůstků portfolia a červenou linkou odhad hustoty rozdělení měsíčních přírůstků. Z histogramu lze vyčíst, že rozdělení

měsíčních přírůstků portfolia se blíží normálnímu rozdělení definovaném pomocí průměru 0.049 a standardní odchylky 0.67.

V další části se budu snažit prokázat, zda je průměrný měsíční přírůstek akciového portfolia větší než nula. Nejprve budu testovat normalitu rozdělení a poté případně průměr.

4.6.1.1 Testování normality přírůstku akciového portfolia

Podle histogramu, který se svým tvarem blíží normálnímu rozdělení, podle blízkosti hodnoty průměru a mediánu se zdá, že lze rozdělení považovat za normální. Z hlediska dalších testů je vhodné zjistit, zda skutečně můžeme považovat soubor měsíčních přírůstků za normální rozdělení. Prvotní informaci z histogramu doplníme o Q-Q graf. Na ose y jsou vyneseny kvantily zkoumaného souboru, na ose x kvantily normálního rozdělení. V případě normálního rozdělení by v grafu byla vidět přímka.



Graf 9: Q-Q graf měsíčních přírůstků akciového portfolia

Z Q-Q grafu č. 9 je zřejmé, že v mezní hodnoty souboru se od normálního rozdělení odchyľují, uprostřed rozmezí se zdá, přesto je velmi pravděpodobné, že hodnoty odpovídají normálnímu rozdělení.

Poslední informaci o normalitě rozdělení nám poskytne Shapirův-Wilkův test normality v **R**. Testujeme nulovou hypotézu, že výběr pochází z normálního rozložení.

> shapiro.test(akcie)

Shapiro-Wilk normality test

data: akcie

W = 0.99215, p-value = 0.612

Závěr: Neodmítáme hypotézu, že rozdělení měsíčních přírůstků akciového portfolia má normální rozdělení.

4.6.1.2 Test průměrného měsíčního přírůstku akciového portfolia

Zdá se, že můžeme přikročit k testování hodnoty průměrného měsíčního přírůstku akciového portfolia. Chceme otestovat, zda je průměrný měsíční přírůstek akciového portfolia větší než 0.

K testování použijeme jednovýběrový t-test.

Stanovíme nulovou a alternativní hypotézu.

$$H_0: \bar{x} = 0$$

$$H_a: \bar{x} \geq 0$$

> t.test(akcie, alternative="greater")

One Sample t-test

data: akcie

t = 0.87152, df = 143, p-value = 0.1925

alternative hypothesis: true mean is greater than 0

95 percent confidence interval:

-0.04380064 Inf

sample estimates:

mean of x

0.04868648

Závěr: Nulovou hypotézu neodmítáme. Nepodařilo se prokázat, že průměrný měsíční přírůstek akciového portfolia je větší než 0.

4.6.1.3 Znaménkový test pozic akciové portfolia

V předchozím testu střední hodnoty jsme testovali, zda v období let 2008-2019 byl měsíční přírůstek významně větší než 0. Pozice mohou být otevřené i několik měsíců a chceme znát, zda byla technická analýza úspěšnější v předpovědi směru pohybu ceny instrumentu, než 50%. Otestujeme úspěšnost predikce technické analýzy. Za úspěšnou predikci bude považována pozice uzavřená v zisku větším než 0. Příklad pozice uzavřené ve ztrátě, v zisku menším než 0, budeme považovat za neúspěšný odhad. U pozic uzavřených s nulovým ziskem/ztrátou sečteme jejich počet a rozdělím v poměru 1:1 ve prospěch každé skupiny pozic. U lichého počtu nulových pozic přidělíme jednu přebývajících nulovou pozici do skupiny úspěšných.

K testování použijeme **binomický test** v R

Stanovíme nulovou a alternativní hypotézu:

$$H_0: p = 0.5$$

$$H_a: p \geq 0.5$$

Kde p je pravděpodobnost, že doporučená pozice bude zisková

Se ziskem bylo uzavřeno 87 pozic z 233.

```
> binom.test(87,233,0.5,alternative="g")
```

```
Exact binomial test
```

```
data: 87 and 233
```

```
number of successes = 87, number of trials = 233, p-value = 1
```

```
alternative hypothesis: true probability of success is greater than 0.5
```

```
95 percent confidence interval:
```

```
0.3205552 1.0000000
```

```
sample estimates:
```

```
probability of success
```

```
0.3733906
```

Závěr: Nulovou hypotézu neodmítáme. Nepodařilo se prokázat, úspěšnost předpovědi směru pohybu instrumentu je vyšší než 50%.

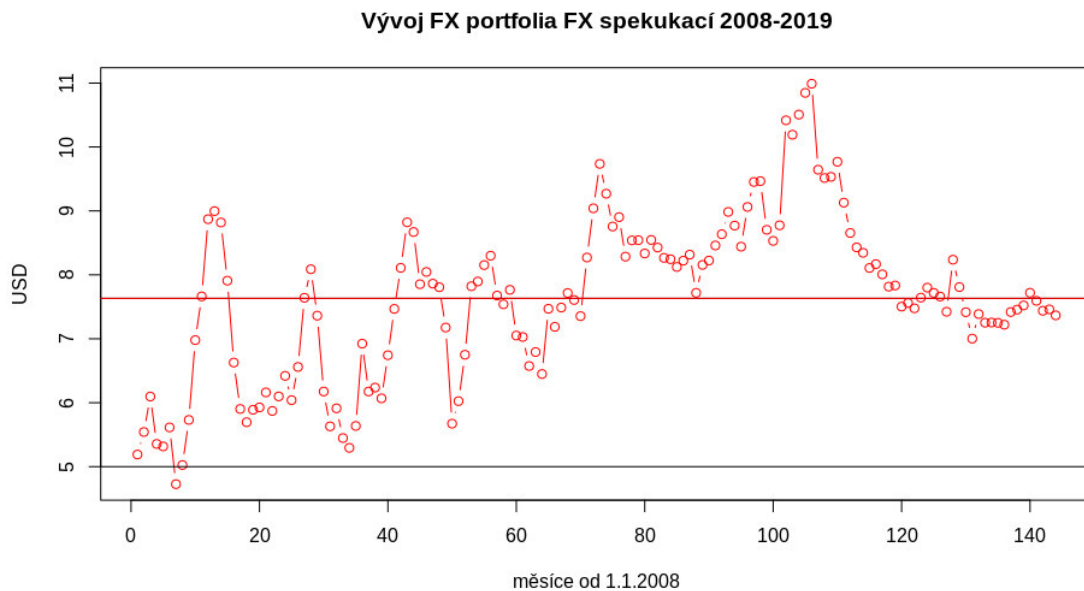
4.6.2 FX Spekulace

Portfolio fx spekulací mělo k 1. lednu 2008 hodnotu \$5. Po dobu dvanácti let byla vždy na konci měsíce zaznamenána aktuální hodnota portfolia a uložena. Po skončení období jsem provedl analýzu s následujícím výsledkem.

Parametr	Hodnota
Počáteční hodnota	5.00
Minimum	4.73
Maximum	10.99
Průměr	7.63
Medián	7.70
Standardní odchylka	1.26

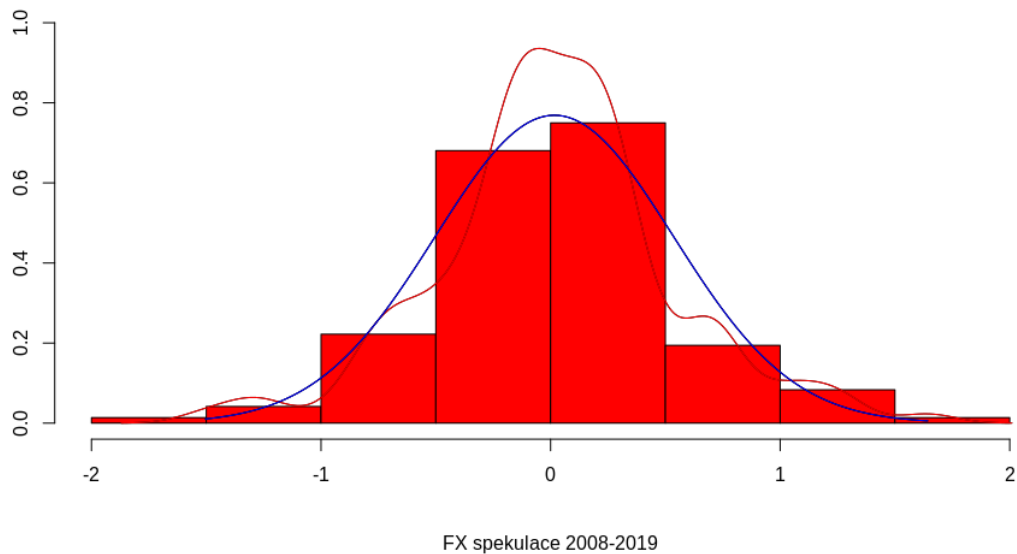
Tabulka 6: Základní statistické parametry portfolia měnových spekulací

Vývoj portfolia fx spekulací po jednotlivých měsících prezentuje následující graf.



Graf 10: Vývoj portfolia FX spekulací v letech 2008-2019. Černá linka \$5 značí počáteční stav, červená průměrný stav

Histogram měsíčních změn portfolia



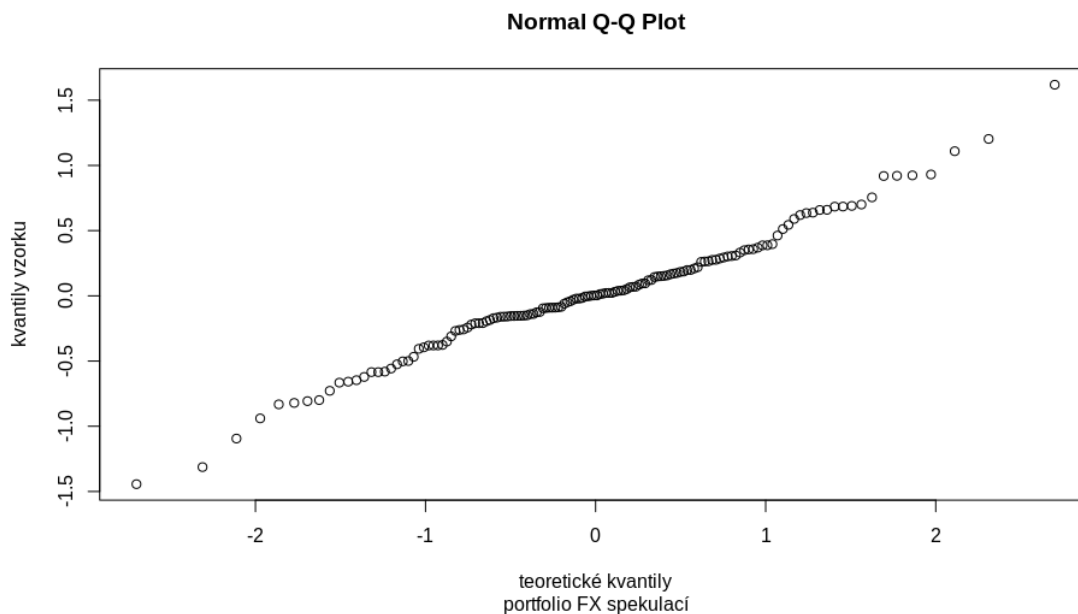
Graf 11: Histogram měsíčních přírůstků portfolia FX spekulací, odhad hustoty rozdělení a odpovídajícího normálního rozdělení

V grafu č. 11 je zakreslený histogram denních přírůstků portfolia a červenou linkou odhad hustoty rozdělení měsíčních přírůstků. Z histogramu lze vyčíst, že rozdělení měsíčních přírůstků portfolia se blíží normálnímu rozdělení definovaném pomocí průměru 0.016 a standardní odchylky 0.52. Odhadovaná hustota rozdělení se však odchyluje od křivky normálního rozdělení více než v případě akciového portfolia.

V další části se budu snažit prokázat, zda je průměrný měsíční přírůstek portfolia FX spekulací větší než nula. Nejprve budu testovat normalitu rozdělení a poté případně průměr.

4.6.2.1 Testování normality přírůstku portfolia FX spekulací

Podle histogramu a odhadu hustoty rozdělení se nezdá, že lze rozdělení považovat za normální. Z hlediska dalších testů je vhodné zjistit, zda soubor měsíčních přírůstků za normální rozdělení považovat můžeme. Prvotní informaci z histogramu doplníme o Q-Q graf. Na ose y jsou vyneseny kvantily zkoumaného souboru, na ose x kvantily normálního rozdělení. V případě normálního rozdělení by v grafu byla vidět přímka.



Graf 12: Q-Q graf měsíčních přírůstků akciového portfolia

Z Q-Q grafu je zřejmé, že hodnoty souboru se od normálního rozdělení odchylojí více než u akciového portfolia.

Poslední informaci o normalitě rozdělení nám poskytne Shapirův-Wilkův test normality v R. Testujeme nulovou hypotézu, že výběr pochází z normálního rozložení.

> shapiro.test(fx)

Shapiro-Wilk normality test

data: fx

W = 0.98191, p-value = 0.05429

Závěr: Neodmítáme hypotézu, že rozdělení měsíčních přírůstků akciového portfolia má normální rozdělení.

4.6.2.2 Test průměrného měsíčního přírůstku portfolia FX spekulací

Vzorek dat odpovídá normálnímu rozdělení, můžeme přikročit k testování hodnoty průměrného měsíčního přírůstku portfolia fx spekulací. Chceme otestovat, zda je průměrný měsíční přírůstek portfolia větší než 0.

K testování použijeme jednovýběrový t-test.

Stanovíme nulovou a alternativní hypotézu.

$$H_0: \bar{x} = 0$$

$$H_a: \bar{x} \geq 0$$

```
> t.test(fx, alternative="greater")
```

One Sample t-test

data: fx

t = 0.38018, df = 143, **p-value = 0.3522**

alternative hypothesis: true mean is greater than 0

95 percent confidence interval:

-0.05513614 Inf

sample estimates:

mean of x

0.01643525

Závěr: Nulovou hypotézu neodmítáme. Nepodařilo se prokázat, že průměrný měsíční přírůstek portfolia fx spekulací je větší než 0.

4.6.2.3 *Znaménkový test pozic portfolia měnových spekulací*

V předchozím testu střední hodnoty jsme testovali, zda v období let 2008-2019 byl měsíční přírůstek významně větší než 0. Pozice mohou být otevřené i několik měsíců a chceme znát, zda byla technická analýza úspěšnější v předpovědi směru pohybu ceny instrumentu, než 50%. Otestujeme úspěšnost predikce technické analýzy. Za úspěšnou predikci bude považována pozice uzavřená v zisku větším než 0. Případ pozice uzavřené ve ztrátě, v zisku menším než 0, budeme považovat za neúspěšný odhad. U pozic uzavřených s nulovým ziskem/ztrátou sečtu jejich počet a rozdělím v poměru 1:1 ve prospěch každé skupiny pozic. U lichého počtu nulových pozic přidělíme jednu přebývající nulovou pozici do skupiny úspěšných.

K testování použijeme **binomický test** v R

Stanovíme nulovou a alternativní hypotézu:

$$H_0: p = 0.5$$

$$H_a: p \geq 0.5$$

Kde p je pravděpodobnost, že doporučená pozice bude zisková

Se ziskem bylo uzavřeno 229 pozic z 625 a 5 pozic s nulovým ziskem.

```
> binom.test(229+3,625,0.5,alternative="g")
```

Exact binomial test

data: 229 + 3 and 625

number of successes = 232, number of trials = 625, p-value = 1

alternative hypothesis: true probability of success is greater than 0.5

95 percent confidence interval:

0.3391146 1.0000000

sample estimates:

probability of success

0.3712

Závěr: Nulovou hypotézu neodmítáme. Nepodařilo se prokázat, úspěšnost předpovědi směru pohybu instrumentu je vyšší než 50%.

4.7 HODNOCENÍ VÝKONNOSTI PORTFOLIÍ

Předchozí statistická analýza neprokázala, že by se průměrný měsíční přírůstek obou portfolií řízených technickou analýzou významně lišil od nulové hodnoty.

V posledním kroku srovnám výnos obou portfolií řízených technickou analýzou s výkonem akciového indexu SP500. Procentní změny v níže uvedené tabulce jsou vždy změnou ke konci roku podle vzorce:

$$změna = \frac{H_r - H_{r-1}}{H_{r-1}} \quad [4.7.1.1.1]$$

H_{r-1} Stav portfolia nebo SP500 na konci přechozího roku

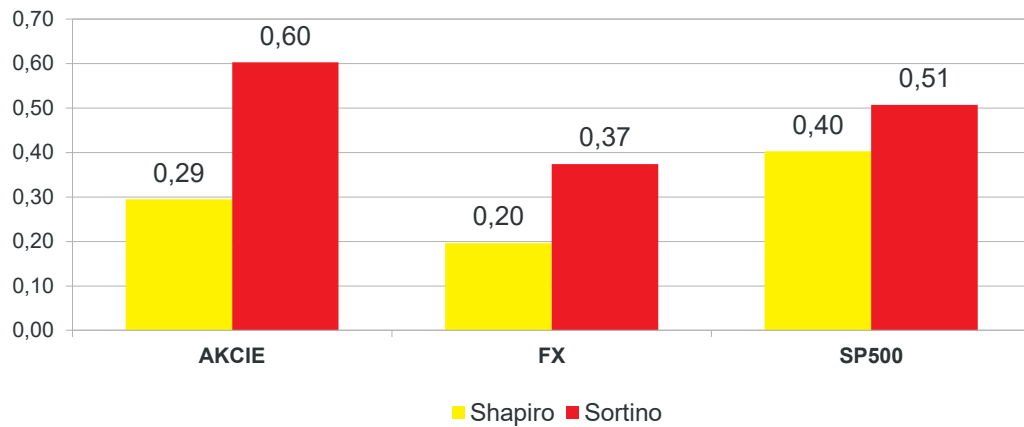
H_r Stav portfolia nebo SP500 na konci aktuálního roku

Pro výpočet indexů jsem jako bezrizikovou referenční sazbu zvolil průměrnou US repo sazbu.

Rok	AKCIE	FX	SP500	US repo vážený průměr
2008	29,48%	77,42%	-44,41%	1,92%
2009	-2,55%	-27,63%	35,78%	0,16%
2010	2,62%	7,84%	8,84%	0,18%
2011	-41,91%	12,75%	3,23%	0,10%
2012	-23,02%	-9,68%	13,17%	0,14%
2013	141,85%	28,22%	27,82%	0,11%
2014	51,09%	-8,80%	13,99%	0,09%
2015	0,78%	9,91%	2,44%	0,13%
2016	0,27%	5,00%	4,18%	0,39%
2017	29,20%	-21,14%	20,58%	1,00%
2018	-4,30%	-1,56%	5,60%	1,79%
2019	-9,12%	-0,27%	11,61%	2,16%
Průměr	14,53%	6,01%	8,57%	0,68%
Sharpe index	0,29	0,20	0,40	
Sortino index	0,60	0,30	0,51	

Tabulka 7: Sharpe a Sortino indexy portfolií a SP500 indexu

Sharpe a Sortino index portfolia & SP500 vs US repo



Graf 13: Sharpe a Sortino index portfolií a SP500

Z tabulky i grafu vyplývá, že akciové portfolio i portfolio FX spekulací vykazují horší Sortino index než index SP500. V okamžiku, kdy změníme standardní odchylku v jmenovateli zlomku za „negativní část standardní odchylky“, přestaneme penalizovat hodnocení kladnými výkyvy a akciové portfolio dosahuje nejvyšší hodnoty.

5 ZÁVĚR

Statistické testování neprokázalo, že by použití zvolených postupů technické analýzy zajistilo průměrný měsíční přírůstek některého z portfolií větší než 0. Neprokázali jsme schopnost naší strategie technické analýzy, odhadnout směr pohybu instrumentu s úspěšností vyšší než 50%. Zhodnocení výkonu portfolií Sharpeho a Sortinovým indexem nepotvrzuje výrazně vyšší hodnoty než samotné investování do nákupu akcií dle indexu SP500.

Práce byla zaměřena na jedinou strategii vyplývající z technické analýzy využívající dvou nejzákladnějších indikátorů. Zcela jistě by bylo zajímavé otestovat další kombinace prvků technické analýzy, různé periody optimalizace případně out-of-sample období. Zajímavé by mohlo být srovnání úspěšnosti technické analýzy s náhodným odhadem pozic, proti strategiím čerpajících z fundamentální analýzy, využití machine learningu a další postupy.

Za jeden z největších přínosů práce považuji informaci, že celý proces od získání dat po závěrečné testování je možné kompletně provést s minimálním rozpočtem a pomocí veřejně dostupných zdrojů, software i dat. Je zřejmé, že se zvyšující se komplexitou optimalizačního modelu by stoupaly nároky na výpočetní výkon.

Summary and keywords

Technical analysis is a widely spread tool in the field of retail financial markets. This thesis aims to assess its prediction reliability in stock and forex markets. The stock market is represented by randomly selected shares of the most known US stock indexes such a S&P 500 and NASDAQ. The foreign currency trade market is represented by currency pairs of the world most important currencies, most of them known as majors.

This work evaluates reliability of the the very basic components of technical analysis - trend indicators and oscillators. Together with the optimization of technical analysis based on historical data, it validates its ability to forecast future moves of markets.

The application of technical analysis is more or less a subject of a personal application. To ensure strictly objective implementation of used methods and to avoid any human influence, all applications of technical analysis are provided by a human independent optimization software and tested on out-of-sample data. Results produced by this software are gathered in SQL database and analysis itself is provided by statistical testing methods in the statistical software environment.

Keywords: technical analysis, stock market, forex, optimization, ouf-of-sample forecast, statistical testing, data

6 LITERATURA

Rejnuš, O. *Finanční trhy*. 4. aktualizované a rozšířené vydání. Praha: Grada Publishing 2014. ISBN 978-80-247-3671-6

Veselá, J. *Investování na kapitálových trzích*. 2. rozšířené a aktualizované vydání. Praha: Wolters Kluwer ČR, 2011. ISBN 978-80-7357-647-9

Mrkvička, T. *Úvod do teorie pravděpodobnosti.*, první vydání dotisk. České Budějovice: Jihočeská univerzita 2008. ISBN 80-7040-894-4

Cipra, T. *Finanční ekonometrie*. 2., upravené vydání. Praha: Ekopress 2013. ISBN 978-80-86929-93-4

Hindls, R. *Statistika pro ekonomy.*, páté vydání. Praha: Professional Publishing 2004. ISBN 80-86419-59-2

Jílek, J. *Finance v globální ekonomice I: Peníze a platební styk*. Praha: Grada Publishing 2013. ISBN 978-80-247-3893-2

Vochozka, M., Mulač, P. a kolektiv, *Podniková ekonomika*, první vydání. Praha: Grada Publishing 2012. ISBN 978-80-247-4372-1

7 INTERNETOVÉ ZDROJE

NIST/SEMATECH, *e-Handbook of Statistical Methods*,

<http://www.itl.nist.gov/div898/handbook/>, 2012.

United States Overnight Repo Rate, <https://tradingeconomics.com/united-states/repo-rate>

Kotace finančních instrumentů, <https://finance.yahoo.com>

8 REJSTŘÍKY

8.1 SEZNAM OBRÁZKŮ

Obrázek 1: Příklad distribuční funkce	17
Obrázek 2: Příklad histogramu a rozdělení hustoty pravděpodobnosti.....	17

8.2 SEZNAM GRAFŮ

Graf 1: Příklad svíčkového grafu pro kurz amerického dolaru a japonského jenu. Červené svíce značí pokles kurzu dolaru vůči jenu, zelené naopak posílení. Zdroj: autor	8
Graf 2 : Příklad čárového grafu pro kurz amerického dolaru a japonského jenu. Graf zobrazuje pouze uzavírací ceny. Zdroj: autor	9
Graf 3: Příklad sloupcového grafu pro kurz amerického dolaru a japonského jenu. Levá linka prezentuje otevírací cenu, zatímco pravá cenu zavírací. Rozpětí ceny spodním a horním okrajem ceny. Zdroj: autor	9
Graf 4: Příklad divergence mezi cenovým grafem a oscilátorem. Divergence v průběhu cenového grafu a oscilátoru je považována za předzvěst změny trendu ceny Zdroj: autor	12
Graf 5: Příklad grafického výstupu třídy Analysis, pozice na akcii AIG; zdroj: autor.....	32
Graf 6: Příklad grafického výstupu třídy Analysis, pozice na akcii AIG; zdroj: autor.....	33
Graf 7: Vývoj akciového portfolia v letech 2008-2019. Černá linka \$5 značí počáteční stav, červená průměrný stav.	35
Graf 8: Histogram měsíčních přírůstků akciového portfolia, odhad hustoty rozdělení a odpovídajícího normálního rozdělení	35
Graf 9: Q-Q graf měsíčních přírůstků akciového portfolia	36
Graf 10: Vývoj portfolia FX spekulací v letech 2008-2019. Černá linka \$5 značí počáteční stav, červená průměrný stav	39
Graf 11: Histogram měsíčních přírůstků portfolia FX spekulací, odhad hustoty rozdělení a odpovídajícího normálního rozdělení	40
Graf 12: Q-Q graf měsíčních přírůstků akciového portfolia	41
Graf 13: Sharpe a Sortino index portfolií a SP500	45

8.3 SEZNAM TABULEK

Tabulka 1: Instrumenty akciového portfolia.....	25
Tabulka 2: Měny portfolia měnových spekulací.....	26
Tabulka 3: Tabulka parametrů indikátorů technické analýzy pro AIG v období 1.2.2018-31.3.2018	31
Tabulka 4: Tabulka parametrů indikátorů technické analýzy pro DKKUSD v období 1.10.2009-30.11.2009	33
Tabulka 5: Základní statistické parametry akciového portfolia.....	34
Tabulka 6: Základní statistické parametry portfolia měnových spekulací.....	39
Tabulka 7: Sharpe a Sortino indexy portfolií a SP500 indexu	44

9 PŘÍLOHY

9.1 IMPORT DAT Z YAHOO.FINANCE.COM DO LOKÁLNÍ DATABÁZE V MYSQL

```
#!/usr/bin/python3

import yfinance as yf
import pymysql
import sqlalchemy as sqa
import argparse
import pandas as pd
import configparser

from datetime import datetime

class yf_dataset():
    def __init__(self):
        titles={}
        self.download=""
        for inx in titles:
            self.download+=titles[inx]+' '

        self.data=pd.DataFrame()
        self.config={}
        self.config['my_username']='*****1'
        self.config['my_password']='*****1'
        self.config['my_hostname']='192.168.2.33'
        self.config['my_port']=3306
        self.config['my_src']='ticks'

    def ReadList(self,configfile):
        try:
            if (configfile!=""):
                config_file=configfile
                f = open(config_file)
                f.close()
                config=configparser.ConfigParser()
                config.read(config_file)
                self.download=""
                for inx in config['downloads']:
                    self.download+=config['downloads'][inx]+' '

        except Exception as emsg:
            print('Error processing config file ',config_file)
            print(emsg)
            exit(1)

    def export(self):
        try:
            connstr="mysql+pymysql://" +self.config["my_username"]+"."+self.config["my_password"]+"@"+self.config["my_hostname"]+"."+str(self.config["my_port"])+"/"+self.config["my_src"]
            engine=sqa.create_engine(connstr)
            dbconn=engine.connect()
            self.data.to_sql('_DATATF_YAHOO_temp', dbconn, if_exists = 'replace')#, index=date) #alternatives if_exists='replace'
            dbconn.execute('delete from ticks._DATATF_YAHOO where (title,date) in (select distinct title,date from
ticks._DATATF_YAHOO_temp);')
            SQL=("INSERT INTO `ticks`.`_DATATF_YAHOO` (title,date,period,open,high,low,close,volume,interval_no,derivation_level) "
                "SELECT title,date(Date),period,open,high,low,close,Volume,interval_no,derivation_level FROM
`ticks`.`_DATATF_YAHOO_temp`");
            dbconn.execute(SQL)
            dbconn.execute('DROP TABLE _DATATF_YAHOO_temp;')
            dbconn.close()

        except Exception as emsg:
            print('Error exporting ')
            print(emsg)
```

```

def Import(self, date_from='1900-1-1', date_to='2099-12-31'):
    for inx in self.download.split(' '):
        if (inx!=""):
            print('Downloading ', inx)
            try:
                self.data=yf.download(inx, date_from, date_to, group_by='ticker')
                self.data=self.data.rename(columns={'Close':'close','Open':'open','High':'high', 'Low':'low'})
                self.data['title']=inx
                self.data['period']=1440
                self.data['interval_no']=0
                self.data['derivation_level']=0
                self.data=self.data.drop(['Adj Close'],axis=1)
                self.export()

            except Exception as emsg:
                print('Error while downloading ',inx)
                print(emsg)

def valid_date(s):
    try:
        return datetime.strptime(s, "%Y-%m-%d").date()
    except ValueError:
        msg = "Not a valid date: '{0}'.".format(s)
        raise argparse.ArgumentTypeError(msg)

class ParseArgs():

    def __init__(self):
        parser = argparse.ArgumentParser(prog='update.yahoo.py', usage='%(prog)s [options]')
        parser.add_argument('-e', '--enddate', type=valid_date, help='end date format YYYY-MM-DD (default today)', default='2099-1-1')
        parser.add_argument('-s', '--startdate', type=valid_date, help='start date format YYYY-MM-DD (default today)', default='1900-12-31')
        parser.add_argument('-t', '--title', help='title to import, importing default list if no title defined')
        parser.add_argument('-c', '--config', type=str, help='config file', default="./download.list")

        args = parser.parse_args()
        self.startdate=args.startdate
        self.enddate=args.enddate
        if args.title is not None:
            self.title=args.title
        else:
            self.title=""

        if args.config is not None:
            self.cfile=args.config

def main():
    args=ParseArgs()
    d=yf_dataset()
    if (args.title!=""):
        d.download=args.title
    else:
        d.ReadList(args.cfile)

    d.Import(date_from=str(args.startdate), date_to=str(args.enddate))

if __name__ == '__main__':
    main()

```

9.2 OPTIMALIZACE STRATEGIE NA HISTORICKÝCH DATECH, VYGENEROVÁNÍ OUT OF SAMPLE

DAT

```
#!/usr/bin/python3
VERSION = '0.102'

# prerequisites
# sudo pip3 install --upgrade pip
# sudo pip3 install mysql-connector pandas numpy yfinance matplotlib sqlalchemy ray setproctitle psutil pymysql argparse
# sudo apt install mysql-client

# changes
# v.101 verification after every round of optimization, caching yahoo quotes
# v.100 floating optimization addedd
# v.010 quantile added, graphs saved as jpeg
# v.009 Show Graphs
# v.008 avaluate mode, saving date range of optimization, more cmd line params, cmd params overwrite config file values
# v.007 anova removed, evaluate mode - half way
# v.006 - choose most robust and profitable strategy, datetime params in config, optimize mode
# v.005 - randomness added, params in config, sumarized_results, flushing unfinished positions in optimize mode
# v.004 config file
# v.003 clustering by ray
# v.002 Params data frame
# financial data source yahoo, dukascopy, quandl
# https://www.quora.com/Using-Python-whats-the-best-way-to-get-stock-data
#https://blog.quantinsti.com/stock-market-data-analysis-python/
#https://docs.quandl.com/docs/error-codes
import math
import argparse
import configparser
import random
import mysql.connector as mysql
import pandas as pd
import numpy as np
import time
import yfinance as yf #yahoo financial data source
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection
import matplotlib.dates as mpd
from matplotlib.colors import ListedColormap, BoundaryNorm
from datetime import datetime, timedelta, date
from dateutil.relativedelta import relativedelta
import sqlalchemy as sqla
import pymysql
import hashlib
import ray
import os
import psutil as pu
import quandl as q
q.ApiConfig.api_key = "*****"

debug_level=1

def debug(msg, passed_level=1):
    if (debug_level >= passed_level):
        print(msg)

def reap():
    while 1:
        try:
            result=os.waitpid(-1, os.WNOHANG)
            if not result[0]: break
        except:
            break

def valid_date(s):
    try:
        return datetime.strptime(s, "%Y-%m-%d").date()
```

```

except ValueError:
    msg = "Not a valid date: '{0}'. ".format(s)
    raise argparse.ArgumentTypeError(msg)

class Random():
    # source #https://www.random.org/integers/?num=10000&min=1&max=100&col=1&base=10&format=plain&rnd=new
    def __init__(self):
        self.file='./random.txt'
        self.randomdata=pd.read_csv(self.file, sep=" ")

@ray.remote
class Analysis():

    def __init__(self):
        t=datetime.now()
        todaystr=str(t.year)+"-"+str(t.month)+"-"+str(t.day)
        self.sdate = datetime.strptime(todaystr, "%Y-%m-%d").date()
        self.edate = datetime.strptime(todaystr, "%Y-%m-%d").date()
        self.MALong=200
        self.MAShort=100
        self.MAClose=25
        self.OBVMA=50
        self.VolumeMA=200
        self.OBV=1
        self.symbol=""
        self.StochK_per=9
        self.StochK_avg=6
        self.StrategyType=""
        self.correl=0
        self.C1per=610
        self.C2per=144

        self.c1field='Corel'+str(self.C1per)
        self.c2field='Corel'+str(self.C2per)
        self.quantile1l='Qntl'+str(self.C1per)+'low'
        self.quantile1h='Qntl'+str(self.C1per)+'high'
        self.quantile2l='Qntl'+str(self.C2per)+'low'
        self.quantile2h='Qntl'+str(self.C2per)+'high'

        self.trendwo=0
        self.trendwc=0

        self.sidewo=0
        self.sidewc=0

        self.corelo=0
        self.corelc=0

        self.bignoo=0
        self.bignoc=0

        self.CalcRunning=0
        self.initialized=np.nan

    def mean(self):
        return(self.data['Pos_Value'].mean())

    def median(self):
        return(self.data['Pos_Value'].median())

    def mode(self):
        return(self.data['Pos_Value'].mode())

    def min(self):
        return(self.data['Pos_Value'].min())

    def max(self):
        return(self.data['Pos_Value'].max())

    def range(self):

```

```

return(self.max-self.min)

def std(self):
    return(self.data['Pos_Value'].std())

#df['column_name'].describe(percentiles=[percentile1, percentile2, percentile3, percentile4] - muze byt i vice percentilu
def quantile(self, qntl): #0<qntl<1
    return(self.data['Pos_Value'].quantile(qntl))

def sum(self):
    return(self.data['Pos_Value'].sum())

def count(self):
    return(self.data['Pos_Value'].count())

def cv(self): # coeficient of variation
    return(self.data['Pos_Value'].std()/self.data['Pos_Value'].mean())

def quantilesum(self, qntl):
    return(self.data.loc[self.data['Pos_Value']<qntl]['Pos_Value'].sum())

def FirstDate(self):
    return(self.data['Date'].min())

def LastDate(self):
    return(self.data['Date'].max())

def DataInit(self, config):
    self.symbol=config["symbol"]
    self.csymbol1=config["csymbol1"]
    self.csymbol2=config["csymbol2"]
    tf=str(config["tf"])
    self.sdate=config["startd"]
    self.edate=config["endd"]
    edate=self.edate
    days=config["initial_back"]
    chdays=days-4
    debug("Mode :"+config['mode'])
    if (config['mode']=="verify"):
        #kvuli dokonceni pozic otevrenyc v obdobi potrebujeme data i po ukonceni obdobi (rok by mohl stacit)
        edate=(datetime.strptime(self.edate,"%Y-%m-%d")+timedelta(days=1200)).strftime("%Y-%m-%d")
        #kvuli vypoctu indikatoru musime vzit i drivejsi data (indi se vypocitaji, pozice se ale budou generovat az v zadanem obdobi
        (rok asi staci i pro nejdelsi MA)
        sdate=(datetime.strptime(self.sdate,"%Y-%m-%d")+timedelta(days=-days)).strftime("%Y-%m-%d")
        check_date=(datetime.strptime(self.sdate,"%Y-%m-%d")+timedelta(days=-chdays)).strftime("%Y-%m-%d")
        elif (config["mode"]=="optimize"):
            #kvuli dokonceni pozic otevrenyc v obdobi potrebujeme data i po ukonceni obdobi (rok by mohl stacit)
            # edate=(datetime.strptime(self.edate,"%Y-%m-%d")+timedelta(days=1200)).strftime("%Y-%m-%d")
            #kvuli vypoctu indikatoru musime vzit i drivejsi data (indi se vypocitaji, pozice se ale budou generovat az v zadanem obdobi
            (rok asi staci i pro nejdelsi MA)
            self.sdate=config["opti_start"]
            self.edate=config["opti_end"]
            sdate=(datetime.strptime(self.sdate,"%Y-%m-%d")+timedelta(days=-days)).strftime("%Y-%m-%d")
            edate=(datetime.strptime(self.edate,"%Y-%m-%d")+timedelta(days=1200)).strftime("%Y-%m-%d")
            check_date=(datetime.strptime(self.sdate,"%Y-%m-%d")+timedelta(days=-chdays)).strftime("%Y-%m-%d")
    else:
        edate=self.edate
        sdate=self.sdate
        check_date=(datetime.strptime(self.sdate,"%Y-%m-%d")+timedelta(days=-4)).strftime("%Y-%m-%d")
        self.CalcRunning=1
        self.correl=((self.csymbol1!="") & (self.csymbol2!=""))+0
        if (config["source"]=="yahoo"):
            if (self.correl==0):
                try:
                    self.db_connection = mysql.connect(host=config["my_hostname"], database=config["my_dbname"],
user=config["my_username"], password=config["my_password"])
                    SQL = "select DATE_FORMAT(date, '%Y-%m-%d') as Date, " as 'Adj close',close as Close, high as High,low as Low,open as
Open,Volume from `ticks`.`_DATATF_YAHOO` \
where title='"+self.symbol+"' and period='"+tf+"' and date>='"+sdate+"' and date<='"+edate+"' order by date asc;"
                    self.data_raw = pd.read_sql(SQL, con=self.db_connection)
                    self.data_raw['Date']=pd.to_datetime(self.data_raw['Date'])

```

```

        self.data_raw=self.data_raw.set_index('Date')
    except Exception as emsg:
        print("Aborted while catching yahoo data")
        print(emsg)
        exit(1)
    else:
        ## needs to be rewritten (data stored in own db)
        try:
            data_s=pd.merge(pd.merge(yf.download(self.symbol,sdate,
            edate),yf.download(self.csymbol1,sdate,edate)['Close'],on=['Date'],validate='one_to_one'),
                yf.download(self.csymbol2,sdate, edate)['Close'],on=['Date'],validate='one_to_one')
            self.data_raw=data_s.rename(columns={'Close':'c2_close','Close_y':'c1_close','Close_x':'Close'})
        except Exception as emsg:
            print("Error #2 downloading data from yahoo")
            print("Exiting script")
            print(emsg)
            exit(1)

    elif (config["source"]=="dukascopy"):
        self.db_connection = mysql.connect(host=config["my_hostname"], database=config["my_dbname"],
        user=config["my_username"], password=config["my_password"])
        if ((self.csymbol2!="") and (self.csymbol1!="")):
            SQL=("select "
                "date_format(a.date,'%Y-%m-%d') as Date," as 'Adj close',a.ask_close as Close, a.ask_high as High,"
                "a.ask_low as Low,a.ask_open as Open,0 as Volume, c1.ask_close as c1_close,c2.ask_close as c2_close,a.interval_no "
                "from "
                "`_DATATF.UTC` a, "
                "`_DATATF.UTC` c1,"
                "`_DATATF.UTC` c2 "
                " where "
                "a.title="+self.symbol+" and a.period="+tf+" and a.date>="+sdate+" and a.date<="+edate+" and "
                "c1.title="+self.csymbol1+" and c1.period="+tf+" and c1.date>="+sdate+" and c1.date<="+edate+" and "
                "c2.title="+self.csymbol2+" and c2.period="+tf+" and c2.date>="+sdate+" and c2.date<="+edate+" "
                "and a.date=c1.date and c2.date=c1.date "
                "and a.interval_no=c1.interval_no and c1.interval_no=c2.interval_no "
                "order by date asc,interval_no asc;")
            self.correl=1
        else:
            SQL = "select date_format(date,'%Y-%m-%d') as Date," as 'Adj close',ask_close as Close, ask_high as High,ask_low as
            Low,ask_open as Open,0 as Volume from `ticks`.`_DATATF.UTC` \
            where title="+self.symbol+" and period="+tf+" and date>="+sdate+" and date<="+edate+" order by date asc;"
            self.data_raw = pd.read_sql(SQL, con=self.db_connection)
            self.data_raw['Date']=pd.to_datetime(self.data_raw['Date'])
            self.data_raw=self.data_raw.set_index('Date')

        debug("dates : "+sdate+"|"+edate+"|"+check_date)

        if (self.data_raw.size==0):
            print("No data to import")
            self.initialized=np.nan
        else:
            debug(str(self.data_raw.index[0])+str(sdate)+str( check_date))
            if (self.data_raw.index[0]>datetime.strptime(check_date,"%Y-%m-%d")):
                debug('Missing data')
                self.initialized=(self.data_raw.index[0]+timedelta(days)).strftime("%Y-%m-%d")

        debug("dates : sdate|edate|check_date|initialized "+sdate+"|"+edate+"|"+check_date+"|"+str(self.initialized))
        self.CalcRunning=0
        debug("DataInit: Done")

    def CalcPositions(self, params):

        def BoolToInt(bl, b):
            if (b==True):
                return 1
            elif (b==False):
                return -1
            else:
                return 0

```

```

self.data['pos_open']=0
self.data['pos_id']=0
self.data['Long_Entry']=False
self.data['Short_Entry']=False
self.data['Long_Close']=False
self.data['Short_Close']=False
self.data['Position_hold']=np.nan
self.data['Pos_Value']=0
self.data['rnd']=np.nan
self.data['Trend_long']=0
self.data['Trend_short']=0
self.data['Sideways_long']=0
self.data['Sideways_short']=0
self.data['Correl_long']=0
self.data['Correl_short']=0
self.data['big_no']=0

self.data['Trend_long']= \
    (self.data['MALong'].shift(1)>self.data['MALong'].shift(7)) & \
    (self.data['MASHort'].shift(1)>self.data['MASHort'].shift(5)) & \
    (self.data['MALong'].shift(1)<self.data['MASHort'].shift(1))

self.data['Trend_short']= \
    (self.data['MALong'].shift(1)<self.data['MALong'].shift(7)) & \
    (self.data['MASHort'].shift(1)<self.data['MASHort'].shift(5)) & \
    (self.data['MALong'].shift(1)>self.data['MASHort'].shift(1))

#checking for correl
if (self.correl==0):
    self.data['Correl_long']=0
    self.data['Correl_short']=0
else:
    self.data['Correl_long']= \
        ((self.data[self.quantile1].shift(2)>self.data['Close'].shift(2)) |
(self.data[self.quantile1].shift(1)>self.data['Close'].shift(1))) & \
        (self.data['Close'].shift(2)<self.data['Close'].shift(1)) & \
        (self.data['Open'].shift(1)<self.data['Close'].shift(1)) & \
        (abs(self.data[self.c1field])>0.7)

    self.data['Correl_short']= \
        ((self.data[self.quantile1h].shift(2)<self.data['Close'].shift(2)) |
(self.data[self.quantile1h].shift(1)<self.data['Close'].shift(1))) & \
        (self.data['Close'].shift(2)>self.data['Close'].shift(1)) & \
        (self.data['Open'].shift(1)>self.data['Close'].shift(1)) & \
        (abs(self.data[self.c1field])>0.7)

if (('StochLower' in params['name'].values) & ('StochUpper' in params['name'].values)):
    self.data['Sideways_long']= \
        (self.data['%K'].shift(1) > self.data['%D'].shift(1)) & \
        (self.data['%K'].shift(2) < self.data['%D'].shift(2)) & \
        (self.data['%D'].shift(1) < int(params.loc[params['name']=='StochLower']['current'].values[0])) #20
    self.data['Sideways_short']= \
        (self.data['%K'].shift(1) < self.data['%D'].shift(1)) & \
        (self.data['%K'].shift(2) > self.data['%D'].shift(2)) & \
        (self.data['%D'].shift(1) > int(params.loc[params['name']=='StochUpper']['current'].values[0])) #80

rnd=Random()
startrow=random.randint(1, 181000-len(self.data))
lastrow=len(self.data)+startrow
self.data['rnd']=rnd.randomdata['RND'].iloc[startrow:lastrow].values
self.data['Rnd_short']=self.data['rnd']<51
self.data['Rnd_long']=self.data['rnd']>50

try:
    self.randomw=params.loc[params['name']=='randomw']['current'].values[0]
except:
    self.randomw=0

try:
    self.trendwo=params.loc[params['name']=='trendwo']['current'].values[0]
    self.trendwc=params.loc[params['name']=='trendwc']['current'].values[0]

```

```

except:
    self.trendwo=0
    self.trendwc=0

try:
    self.sidewo=params.loc[params['name']=='sidewo']['current'].values[0]
    self.sidewc=params.loc[params['name']=='sidewc']['current'].values[0]
except:
    self.sidewo=0
    self.sidewc=0

try:
    self.quantilwo=params.loc[params['name']=='quantilwo']['current'].values[0]
    self.quantilwc=params.loc[params['name']=='quantilwc']['current'].values[0]
except:
    self.quantilwo=0
    self.quantilwc=0

try:
    self.corelo=params.loc[params['name']=='corelo']['current'].values[0]
    self.corelc=params.loc[params['name']=='corelc']['current'].values[0]
except:
    self.corelo=0
    self.corelc=0

try:
    self.bignoo=params.loc[params['name']=='bignoo']['current'].values[0]
    self.bignoc=params.loc[params['name']=='bignoc']['current'].values[0]
except:
    self.bignoo=0
    self.bignoc=0

if ((self.quantil_long_close!=np.nan) & (self.quantil_short_close!=np.nan)):
    try:
        self.data['quantil_long_close']=self.data['quantil'].shift(1)>=self.quantil_long_close
        self.data['quantil_short_close']=self.data['quantil'].shift(1)<=self.quantil_short_close
    except:
        self.data['quantil_long_close']=0
        self.data['quantil_short_close']=0
        self.quantilwc=0

if ((self.quantil_long_open!=np.nan) & (self.quantil_short_open!=np.nan)):
    try:
        self.data['quantil_long_open']=self.data['quantil'].shift(1)==self.quantil_long_open
        self.data['quantil_short_open']=self.data['quantil'].shift(1)==self.quantil_short_open
    except:
        self.data['quantil_long_open']=0
        self.data['quantil_short_open']=0
        self.quantilwo=0

self.data['Long_Entry']=(self.data['Trend_long']*self.trendwo+
    self.data['Sideways_long']*self.sidewo+
    self.data['Correl_long']*self.corelo+
    self.data['Rnd_long']*self.randomw+
    self.data['bigno']*self.bignoo
    )>=100
self.data['Short_Entry']=(self.data['Trend_short']*self.trendwo+
    self.data['Sideways_short']*self.sidewo+
    self.data['Correl_short']*self.corelo+
    self.data['Rnd_short']*self.randomw+
    self.data['bigno']*self.bignoo
    )>=100

##
self.data['Short_Close']=((self.data['Trend_long']*70+self.data['Sideways_long']*40+self.data['Correl_long']*40+(abs(self.data[self.
c1field])<0.7)*40)>=40) | (self.data['Long_Entry']==True)
##
self.data['Long_Close']=((self.data['Trend_short']*70+self.data['Sideways_short']*40+self.data['Correl_short']*40+(abs(self.data[se
lf.c1field])<0.7)*40)>=40) | (self.data['Short_Entry']==True)
    self.data['Short_Close']=(( self.data['Trend_long']*self.trendwc+
        self.data['Sideways_long']*self.sidewc+

```



```

        self.data['Correl_long']*self.corelc+
        self.data['bigno']*self.bignoc
        #self.data['quantil_long_close']*self.quantilwc
        )>=80) | (self.data['Long_Entry']==True)

self.data['Long_Close']=(( self.data['Trend_short']*self.trendwc+
        self.data['Sideways_short']*self.sidewc+
        self.data['Correl_short']*self.corelc +
        self.data['bigno']*self.bignoc
        #self.data['quantil_short_close']*self.quantilwc
        )>=80) | (self.data['Short_Entry']==True)

try:
    self.data.loc[self.data['Long_Entry'],'long_pos_open'] = 1
    self.data.loc[self.data['Long_Close'],'long_pos_open'] = 0
    self.data['long_pos_open'] = self.data['long_pos_open'].fillna(method='ffill')

    self.data.loc[self.data['Short_Entry'],'short_pos_open'] = 1
    self.data.loc[self.data['Short_Close'],'short_pos_open'] = 0
    self.data['short_pos_open'] = self.data['short_pos_open'].fillna(method='ffill')
except:
    print("Error during optimization")
    print("Optimization skipped")
    exit(1)

self.data.loc[self.data['long_pos_open']==1, 'pos_open']=1
self.data.loc[self.data['short_pos_open']==1, 'pos_open']=-1

self.data['pos_id']=(self.data['pos_open']!=self.data['pos_open'].shift(1)).cumsum()
self.data['Pos_Value']=self.data['pos_open']*(self.data['Close']-self.data['Close'].shift(1))
self.data['Cumsum_Value']=self.data['Pos_Value'].cumsum().round(4)

def CalcIndi(self, params, config):

#Simple Moving Averages
try:
    self.MALong=params.loc[params['name']=='MALong']['current'].values[0]
    self.data['MALong']=self.data['Close'].rolling(window=self.MALong).mean()
except:
    self.data['MALong']=np.nan

try:
    self.MAShort=params.loc[params['name']=='MAShort']['current'].values[0]
    self.data['MAShort']=self.data['Close'].rolling(window=self.MAShort).mean()
except:
    self.data['MAShort']=np.nan

try:
    self.MAClose=params.loc[params['name']=='MAClose']['current'].values[0]
    self.data['MAClose']=self.data['Close'].rolling(window=self.MAClose).mean()
except:
    self.data['MAClose']=np.nan

try:
    self.quantil_long_open=params.loc[params['name']=='quantil_long_open']['current'].values[0]
    self.quantil_short_open=params.loc[params['name']=='quantil_short_open']['current'].values[0]
except Exception as emsg:
    self.quantil_long_open=np.nan
    self.quantil_short_open=np.nan
    debug("No open quantil params set | all ignored")
    debug(emsg)

try:
    self.bignoi=params.loc[params['name']=='bignoi']['current'].values[0]
    koef=self.data['Close'].mean()
    if (koef<10):
        self.bignoi=self.bignoi*0.0001
    elif (koef<1000):
        self.bignoi=self.bignoi*0.01
    elif (koef<10000):
        self.bignoi=self.bignoi*0.1
    self.data['bignoi']=self.bignoi

```

```

        self.data['bigno']=(np.floor(self.data['Open']).shift(1)/self.bignoi)!=np.floor(self.data['Close']/self.bignoi)
    except :
        self.data['bigno']=0
        self.data['bignc']=0
        debug("No bigno found in config file")

    try:
        self.quantil_long_close= params.loc[params['name']=='quantil_long_close']['current'].values[0]
        self.quantil_short_close=params.loc[params['name']=='quantil_short_close']['current'].values[0]
    except Exception as emsg:
        self.quantil_long_close=np.nan
        self.quantil_short_close=np.nan
        debug("No close quantil params set | all ingnored", 2)
        debug(emsg, 2)

    self.data['quantil']=(self.data['Close']-self.data['MALong'])/self.data['Close'].rolling(window=self.MALong).std()
    self.data['qquantil']=round(self.data['quantil'])

    try:
        self.StochK_per=params.loc[params['name']=='StochK_per']['current'].values[0]
        self.StochK_avg=params.loc[params['name']=='StochK_avg']['current'].values[0]
        self.data['StochL'] = self.data['Low'].rolling(window=self.StochK_per).min()
        self.data['StochH'] = self.data['High'].rolling(window=self.StochK_per).max()
        self.data['%K'] = 100*((self.data['Close'] - self.data['StochL']) / (self.data['StochH'] - self.data['StochL'])) # actual close
        position in StochK_per days range
        self.data['%D'] = self.data['%K'].rolling(window=self.StochK_avg).mean() # %K StochK_avg day SMA
    except:
        self.data['StochL']=np.nan
        self.data['StochH']=np.nan
        self.data['%K']=np.nan
        self.data['%D']=np.nan

#OBV
    try:
        self.VolumeMA=params.loc[params['name']=='VolumeMA']['current'][0].values[0]
        self.OMVMA=params.loc[params['name']=='OBVMA']['current'][0].values[0]
        self.data['dir']=np.nan
        self.data['OBV_delta']=0
        self.data['OBV']=0
        self.data['Volume']=self.data['Volume']/100000
        self.data['VolumeMA']=self.data['Volume'].rolling(window=self.VolumeMA).mean()
        self.data['dir']= ((self.data['Close']-self.data['Close'].shift(1)) / (self.data['Close']-self.data['Close'].shift(1)).abs())
        self.data['dir'].iloc[0]=1
        self.data['OBV_delta']=self.data['Volume']*self.data['dir']
        self.data['OBV']=self.data['OBV_delta'].cumsum()
        self.data['OBVMA']=self.data['OBV'].rolling(window=self.OBVMA).mean().round()
    except:
        self.data['Volume']=np.nan
        self.data['OBV']=np.nan
        self.data['OBVMA']=np.nan

    if (self.correl==1):
        #Correlations
        try:
            self.C1per=params.loc[params['name']=='C1per']['current'].values[0]
            self.C2per=params.loc[params['name']=='C2per']['current'].values[0]
            self.c1field='Corel'+str(self.C1per)
            self.c2field='Corel'+str(self.C2per)
            self.quantile1l='Qntl'+str(self.C1per)+'low'
            self.quantile1h='Qntl'+str(self.C1per)+'high'
            self.quantile2l='Qntl'+str(self.C2per)+'low'
            self.quantile2h='Qntl'+str(self.C2per)+'high'

        self.data[self.c1field]=self.data['c1_close'].rolling(window=self.C1per).corr(self.data['c2_close'].rolling(window=self.C1per))

        self.data[self.c2field]=self.data['c1_close'].rolling(window=self.C2per).corr(self.data['c2_close'].rolling(window=self.C2per))
        self.data[self.quantile1l]=self.data['Close'].rolling(window=self.C1per).quantile(0.05)
        self.data[self.quantile1h]=self.data['Close'].rolling(window=self.C1per).quantile(0.95)
        self.data[self.quantile2l]=self.data['Close'].rolling(window=self.C2per).quantile(0.05)
        self.data[self.quantile2h]=self.data['Close'].rolling(window=self.C2per).quantile(0.95)
    except:

```

```

self.data[self.c1field]=np.nan
self.data[self.c2field]=np.nan
self.data[self.quantile1]=np.nan
self.data[self.quantile1h]=np.nan
self.data[self.quantile2]=np.nan
self.data[self.quantile2h]=np.nan

def ExportData(self, directory, params, config):
    self.export=self.data[['pos_open', 'pos_id', 'Pos_Value', 'Cumsum_Value', 'Close']]
    additional_params=pd.DataFrame( [['startd', 'Start date',str(self.sdate), "",
                                     ['enddd', 'End date', str(self.edate), ""],
                                     ['symbol', 'base dat', self.symbol, self.symbol],
                                     ['symbolc1', 'correl symbol1', self.csymbol1, self.csymbol1],
                                     ['symbolc2', 'correl symbol2', self.csymbol2, self.csymbol2]],
                                     columns=['name', 'description','initial', 'current']
    )
    params=params.append(additional_params, ignore_index=True, sort=True)
    params["mode"]=config["mode"]

strategy_id=(hashlib.md5((params.to_string()+config["mode"]+str(config["strategy_batch"])+str(self.sdate)+str(self.edate)).encode()).hexdigest())

    params.loc[:, 'strategy_id']=strategy_id
    self.export.loc[:, 'strategy_id']=strategy_id
    self.export.loc[:, 'mode']=config["mode"]

    if (('Date' in self.export.columns)!=False):
        self.export['Date']=self.export.index.copy()

    strategy_batch_df=pd.DataFrame([[config["strategy_batch"], config["symbol"], strategy_id, config["startd"], config["enddd"],
    config["name"]]]),
                                columns=['strategy_batch', 'symbol', 'strategy_id', 'date_from', 'date_to', 'strategy_name'])
    strategy_batch_df=strategy_batch_df.set_index('strategy_batch')

    self.export.loc[:, 'strategy_batch']=config["strategy_batch"]

    if (config['sumarized_results']=='yes'):
        try:
            summarized=pd.DataFrame([[self.export.index.min(), self.export.index.max(),self.export['pos_id'].max(),
            self.export['Cumsum_Value'].min(), self.export['Cumsum_Value'].max(),
            self.export['Cumsum_Value'].iat[-1],
            strategy_id, config["strategy_batch"], config["mode"]]] ,
            columns=['date_from', 'date_to', 'positions', 'min_op', 'max_op', 'pips', 'strategy_id', 'strategy_batch',
            'mode']
        )
        summarized=summarized.set_index('date_from')
    except Exception as emsg:
        print('Error while sumarizing results .. oroginal error message as following')
        debug(emsg)
        exit(1)

    if (config["module"]=="csv"):
        self.export.to_csv(directory+"/"+strategy_id+".csv")
    elif (config["module"]=="mysql"):
        do_write=True
        reap()
        epid=os.fork()
        if (epid==0):
            while (do_write):
                try:
connstr="mysql+pymysql://"+config["my_username"]+"."+config["my_password"]+"@"+config["my_hostname"]+"."+str(config["my_port"])+"/"+config["my_results"]
            engine=sqla.create_engine(connstr)
            dbconn=engine.connect()

            if (config['detail_results']=='yes'):
                try:
                    self.export.to_sql("results", dbconn, if_exists = 'append')#, index=date) #alternatives if_exists='replace'
                except Exception as emsg:

```

```

        print('Error exporting results '+str(msg))

    if (config['sumarized_results']=='yes'):
        try:
            summarized.to_sql("suma_results", dbconn, if_exists = 'append')#, index=date) #alternatives if_exists='replace'
        except Exception as msg:
            print('Error exporting results '+str(msg))

    try:
        strategy_batch_df.to_sql("strategy_batch", dbconn, if_exists='append')
    except Exception as msg:
        print('Error exporting batch '+str(msg))

    try:
        if (config['mode']!='verify'):
            params.to_sql("parameters", dbconn, if_exists = 'append')#, index=date) #alternatives if_exists='replace'
    except Exception as msg:
        print('Error exporting parameters '+str(msg))

    dbconn.close()
    del self.export
    del engine
    do_write=False
except Exception as msg:
    print('Sending to mysql server failed..' +str(msg))
    do_write=True
os._exit(0)

def ShowIndiGraphs(self, config):

    #pos=11
    pos=1
    if (self.data['Volume'].sum())>0:
        graphs=6
    else:
        graphs=5

    plt.figure(figsize=(20, graphs*11))
    plt.subplots_adjust(hspace=0.3)

    ax5=plt.subplot(221)
    pos=pos+1
    plt.ylabel("Days")
    plt.xlabel("Points")
    tmp=self.data.loc[self.data['Pos_Value'].notnull()]['Pos_Value']
    tmp.plot(kind='hist', legend=True, label="Points/day density")
    plt.xticks(rotation=0)
    plt.axvline(x=0, color='r', linestyle='-')
    plt.locator_params(axis='x', nbins=9)
    plt.locator_params(axis='y', nbins=4)

    ax6=plt.subplot(222)
    pos=1
    plt.ylabel("Probability")
    plt.xlabel("Daily gains")
    prob=self.data['Pos_Value'].value_counts(normalize=True).sort_index()
    probdf=pd.DataFrame()
    probdf['probabilities']=prob
    plt.axvline(x=0, color='r', linestyle='-')
    plt.axhline(y=0.5, color='r', linestyle='-')
    probdf['probabilities'].cumsum().plot(legend=True, label="Probability")
    plt.xticks(rotation=0)
    plt.xlim(left=self.data['Pos_Value'].min())
    plt.xlim(right=self.data['Pos_Value'].max())
    try:
        breakev=sorted(loc[sorted['Scum_sum']>0].iloc[0]['Pos_Value'])
        plt.axvline(x=breakev, color='r', linestyle='-')
    except:
        {}

```

```

ax2=plt.subplot(212)
pos=pos+1
plt.ylabel("")

x=mpd.date2num(self.data.index.to_pydatetime())
y=self.data['Close'].values
c=self.data['pos_open'].values
points = np.array([x, y]).T.reshape(-1, 1, 2)
segments = np.concatenate([points[:-1], points[1:]], axis=1)
cmap = ListedColormap(['red', 'black', 'g'])
norm = BoundaryNorm([-100, -0.5, 0.5, 100], cmap.N)
lc = LineCollection(segments, cmap=cmap, norm=norm)
lc.set_array(c)
ax2.add_collection(lc)
plt.gca()
self.data['MALong'].plot(label=self.symbol+" SMA "+str(self.MALong), legend=True, color="blue")
self.data['MAShort'].plot(label=self.symbol+" SMA "+str(self.MAShort), legend=True, color="red")

ax2.legend((self.symbol, ), loc='upper left')
plt.ylabel(self.symbol)
ax2b=ax2.twinx()
self.data['Cumsum_Value'].plot(label="Strategy performance", legend=False, color="black", linestyle='dotted')
ax2b.legend(('strategy performance', ), loc='upper right')
plt.ylabel('Strategy performance')
plt.axhline(y=0, color='black', linestyle='dotted')

#self.data['Close'].plot(label=self.symbol, legend=True, color='black')

plt.xticks(rotation=20)
plt.locator_params(axis='x', nbins=6)
plt.locator_params(axis='y', nbins=5)
plt.xlabel("date")
ax2b.set_xticklabels([])

if (self.data['Volume'].sum())>0:
    ax1=plt.subplot(235)
    plt.ylabel(self.symbol+' Daily volume')
    self.data['Volume'].plot(label=" Volume", legend=True , color="blue")
    plt.xticks(rotation=0)
    plt.locator_params(axis='x', nbins=11)
    plt.locator_params(axis='y', nbins=5)
    ax1.set_xticklabels([])
    plt.xlabel("")
    pos=pos+1

plt.suptitle(self.symbol+" "+config['name']+" analysis \n"+self.sdate+" - "+self.edate)
fig= plt.gcf()
fig.set_size_inches(10, 5.5) # set a suitable size
plt.subplots_adjust(right=0.75) # adjust plot area
plt.savefig(config['name'].strip()+ '_' +config['symbol']+'_'+config['strategy_batch']+'.jpeg')

if (config["showgraph"] == 'yes'):
    #plt.show(block=True)
    plt.show(block=False)
    plt.pause(5)
plt.close()

try:
    del ax1, ax2, ax3, ax4, ax5, ax6
except NameError:
    {}#print('no ax1 to delete' )

def CalcPrepare(self, config):
    self.data=self.data_raw.copy()
    if (config['mode']=="optimize"): #cutting data set et the end of training period
        edate=config["endd"]
        self.data=self.data.loc[self.data.index<=edate]
    else:
        self.data=self.data_raw.copy()

```

```

def Calc(self, counter, params, config):
    while (self.CalcRunning==1): # previous calc not finished yet
        time.sleep(0.001)
        print('not finished.. wait')

    debug("Calc ...initialized |startd |"+str(self.initialized)+" |"+str(config["startd"]))
    if (self.initialized is not np.nan) & (self.initialized<=datetime.strptime(config["startd"],"%Y-%m-%d").strftime("%Y-%m-%d")):
        self.CalcRunning=1
        debug("Calc ...preparing data")
        self.CalcPrepare(config)
        debug("Calc ...calc indi")
        self.CalcIndi(params, config) #indicators on
        debug("Calc ...preparing positions")
        self.data=self.data.loc[self.data.index>=config["startd"]] #calc positions from the begining of period to the end
        #tady
        self.CalcPositions(params)

    # while in optimize mode, flush unfinished position, while in verify mode finish position
    if (config['mode']=="optimize"):
        if ((self.data['long_pos_open'].iat[-1]==1) | (self.data['short_pos_open'].iat[-1]==1)):
            pos2flush=self.data['pos_id'].iat[-1]
            self.data=self.data.loc[self.data['pos_id']<pos2flush] #training period positions only, cut unfinished one

    if (config['mode']=="verify"): #let the last position finish and cut the rest
        edate=config["endd"]
        pos2extend=""
        while (self.data.loc[self.data.index==edate]['pos_id'].empty):
            print(edate,self.data.index[0] , self.data.index[-1])
            edate=(datetime.strptime(edate,"%Y-%m-%d")+timedelta(days=-1)).strftime("%Y-%m-%d")
            pos2extend=self.data.loc[self.data.index==edate]['pos_id'][0] # the last position id stored in pos2extend
            self.data=self.data.loc[self.data['pos_id']<=pos2extend] # cutting all but the last position and lower

    if (config["nosave"]==0):
        self.ExportData("", params, config)

    print(counter, '***** SUM ', round(self.data['Cumsum_Value'][-1],4), '| POSITIONS ', self.data.pos_id.max(), '| BATCH ',
    config["strategy_batch"],
        '|', config["symbol"], config["startd"], config["endd"])
    self.CalcRunning=0
    else:
        debug("Unable to calc indi & positions - data not initialized")

def Running(self, status=-1):
    if (status==0): self.CalcRunning=0; #non running status can be set, otherwise just reading status
    return self.CalcRunning

def IsCorrel(self):
    debug("Calling self.correl")
    debug("self.correl="+str(self.correl))
    return self.correl

def Result(self):
    print('trendwo ', self.trendwo,'| trendwc ', self.trendwc, '| sidewo ', self.sidewo,'| sidewc ', self.sidewc,'| MALong
    ',self.MALong,'| MAShort ', self.MAShort, '| corelo ', self.corelo,'| corelc ', self.corelc)

def InitDate(self):
    return self.initialized

class Optimize():
    def __init__(self):
        #Soucet vsech vah pro otevrenee/zavrenee musi byt >=100
        self.max_threads=1;
        self.opti_finished=0
        self.config={}
        self.config["tf"]=1440
        self.config["source"]="dukascopy"
        self.config["symbol"]="AUDNZD"
        self.config["csymbol1"]="AUDJPY"
        self.config["csymbol2"]="NZDJPY"
        self.config["module"]="mysql"
        self.config["startindex"]=0

```

```

self.config['mode']="optimize"
self.config["period_back"]=1200
self.Params=pd.DataFrame(columns=['name', 'description', 'class', 'initial', 'current', 'max', 'step'])

def NextParams(self):
def CheckWeights(params):
    try:
        trendwo=params.loc[params['name']=='trendwo']['current'].values[0]
    except:
        trendwo=0
    try:
        sidewo=params.loc[params['name']=='sidewo']['current'].values[0]
    except:
        sidewo=0
    try:
        corelo=params.loc[params['name']=='corelo']['current'].values[0]
    except:
        corelo=0
    try:
        bignoo=params.loc[params['name']=='bignoo']['current'].values[0]
    except:
        bignoo=0
    try:
        randomw=params.loc[params['name']=='randomw']['current'].values[0]
    except:
        randomw=0
    return ((trendwo+sidewo+corelo+randomw+bignoo)>=100)

# do budoucna bude vhodnejsi udelat filtry algoritmy a ne tak specificky, jak nasleduje

go_on=True
weights_bad=True
while (weights_bad):
    inx=0
    do_loop=True
    while (do_loop):
        if ((self.Params.at[inx, 'class']!='d') | (self.Params.at[inx, 'class']!='Nan')):
            self.Params.at[inx, 'current']=self.Params.at[inx, 'current']+self.Params.at[inx, 'step']
        else:
            self.Params.at[inx, 'current']=self.Params.at[inx, 'current']+timedelta(days=self.Params.at[inx, 'step'] )

        if (self.Params.at[inx, 'current']>self.Params.at[inx, 'max']):
            self.Params.at[inx, 'current']=self.Params.at[inx, 'initial']
            if (inx<len(self.Params.index)-1):
                inx+=1
            else:
                do_loop=False
                go_on=False
                self.opti_finished=1
        else: do_loop=False
        weights_bad=not(CheckWeights(self.Params))      # jestli nejsme na konci a vahy pro otevreni pozice jsou mensi nez limit
100, pak jedeme dal
    return(go_on)

def NextInx(self, inx):      # vrat hodnotu volneho AnalysisList[]
    InxGot=0
    if (self.max_threads>0):
        while (InxGot==0):
            if (self.opti_finished==0):      #jeste jsme neskoncili vsechny varianty
                inx+=1
                if (inx==self.max_threads):
                    inx=0
            InxGot=(ray.get(self.AnalysisList[inx].Running.remote())==0)
    else:
        InxGot=(self.AnalysisList[0].Running()==0)
    return(inx)

def VerifyInternal(self, startd, endd):
    oos=Optimize()
    oos.ConfigRead()
    oos.config["mode"]='verify'

```

```

oos.config["startd"]=startd
oos.config["endd"]=endd
oos.config["strategy_batch"]=self.config["strategy_batch"]
oos.config["nosave"]=0
oos.config["sumarized_results"]="yes"
oos.config["detail_results"]="yes"
oos.max_threads=1
debug('Out of sample summary :')
debug(oos.config["startd"]+"|" +oos.config["endd"]+"|" +oos.config["strategy_batch"])
oos.Verify()
del oos

def InitDate(self):
    if (self.max_threads>1):
        id=(ray.get(self.AnalysisList[0].InitDate.remote()))
    else:
        id=(self.AnalysisList[0].initialized)
    return(datetime.strptime(id,"%Y-%m-%d").date())

def DoOptimize(self):
    #optimization criteria - daily profit, profit std, profit range, profit histogram - potential - avoid 5% left ruin all the profits
    #test for student or normal distribution
    #test hypotesis profit<0
    inx=0
    startindex=self.config["startindex"]
    if (startindex>0): print("waiting for parameters of ", startindex)

    debug('max threads'+str(self.max_threads))
    o_start=datetime.strptime(self.config['opti_start'], "%Y-%m-%d").date()
    o_end=datetime.strptime(self.config['opti_end'], "%Y-%m-%d").date()

    #Defaults for floating opti
    Q = relativedelta(months=3) # default for floating optimization
    TP = relativedelta(months=18) #training period - currently 1.5 year

    #Specific for one year (and with mistake in optimize.sh (startts from 1.6.)) - Trend Following
    Q = relativedelta(months=4)
    TP = relativedelta(months=18) #training period - currently 1.5 year

    #Specific for one year (and with mistake in optimize.sh (startts from 1.6.)) - Trend Following
    Q = relativedelta(months=1)
    TP = relativedelta(months=18) #training period - currently 1.5 year

    Q=relativedelta(months=self.config['OOS'])
    TP=relativedelta(months=self.config['TP'])

    D = relativedelta(days=1)
    # training period

    base_batch=self.config["strategy_batch"]
    minor_batch=0
    debug("Do Optimize o_start "+str(o_start))
    if (o_start<self.InitDate()):
        o_start=datetime.strptime(str(int(o_start.strftime("%Y"))+1)+"-1-1", "%Y-%m-%d").date()
    debug("Do Optimize o_start "+str(o_start))

    while ((o_start+TP-D)<=o_end):
        counter=0
        self.config["startd"]=str(o_start)
        self.config["endd"]=str(o_start+TP-D)
        print("Currently optimizing ", self.config["startd"], "|", self.config["endd"], "|", o_end)
        self.config["strategy_batch"]=str(base_batch*1000+minor_batch)
        counting=True
        self.opti_finished=0
        while counting:
            counter+=1
            if (counter>=startindex):
                if (self.max_threads>1):
                    inx=self.NextInx(inx)
                    self.AnalysisList[inx].Calc.remote(counter, self.Params, self.config)

```



```

        else:
            self.AnalysisList[0].Calc(counter, self.Params, self.config)
        elif ((math.floor(counter/10000)*10000)==counter):
            print("Actual index " , counter)
            counting=self.NextParams()

Finished=False
if (self.max_threads==1):
    Finished=self.AnalysisList[0].Running()
else:
    while (not(Finished)):
        for i in range(self.max_threads):
            Finished=(Finished or (not(ray.get(self.AnalysisList[i].Running.remote()))))
            time.sleep(0.01)

self.Evaluate()

debug('Training summary :')
debug(self.config["startd"]+"|" +self.config["endd"]+"|" +self.config["strategy_batch"])
debug("")
#do not verify when evaluation aborted
self.VerifyInternal(str(o_start+TP), str(o_start+TP+Q-D))

o_start += Q
minor_batch += 1

def ConfigRead(self):
    parser = argparse.ArgumentParser(prog='Optimize.py', usage='% (prog)s [options]')
    parser.add_argument('-b', '--batch', type=str, help='strategy batch')
    parser.add_argument('-c', '--config', type=str, help='config file (default ./optimize.cfg)', default='./optimize.cfg')
    parser.add_argument('-e', '--enddate', type=valid_date, help='end date format YYYY-MM-DD (default today)')
    parser.add_argument('-m', '--mode', type=str, help='working mode [optimize|verify|evaluate], default=optimize',
    default='optimize')
    parser.add_argument('-o', '--cores', type=int, help='reserved cores. Running on more than 1 core means running in ray cluster')
    parser.add_argument('-s', '--startdate', type=valid_date, help='start date format YYYY-MM-DD (default today)')
    parser.add_argument('-tt', '--ttitle', type=str, help='instrument equal to symbol in config file')
    parser.add_argument('-sr', '--saveresults', type=str, help='sumarized | detailed | all | none')
    parser.add_argument('-g', '--showgraph', type=str, help='yes | no', default='no')

    parser.add_argument('-v', '--version', action='version', version='Version: %(prog)s-{version}'.format(version=VERSION))
    parser.add_argument('-inx', '--index', type=str, help='starting index in optimize mode', default=0)
    args = parser.parse_args()

    if args.index is not None:
        self.config["startindex"] = int(args.index)

    try:
        f = open(args.config)
        f.close()
        config=configparser.ConfigParser()
        config.read(args.config)
        self.config["my_hostname"]=config['DB']['my_hostname']
        self.config["my_dbname"]=config['DB']['my_dbname']
        self.config["my_results"]=config['DB']['my_results']
        self.config["my_username"]=config['DB']['my_username']
        self.config["my_password"]=config['DB']['my_password']
        self.config["my_port"]=int(config['DB']['my_port'])
        self.config["symbol"]=dict(config.items('DATA'))['symbol']
        try:
            self.config["csymbol1"]=dict(config.items('DATA'))['csymbol1']
            self.config["csymbol2"]=dict(config.items('DATA'))['csymbol2']
        except:
            self.config["csymbol1"]=" "
            self.config["csymbol2"]=" "
        self.config["startd"]=dict(config.items('DATA'))['startd']
        self.config["endd"]=dict(config.items('DATA'))['endd']
        self.config["source"]=config['DATA']['source']
        self.config["strategy_batch"]=int(config['DATA']['strategy_batch'])
        self.config["sumarized_results"]=config['DATA']['sumarized_results']
        self.config["detail_results"]=config['DATA']['detail_results']

```

```

self.config["mode"]=config['DATA']['mode']
self.config["initial_back"]=int(config["DATA"]["period_back"])

try:
    if (config["DATA"]["nosave"]=="yes"):
        self.config["nosave"]=1
    else:
        self.config["nosave"]=0
except:
    self.config["nosave"]=0

try: self.config["name"]=config['ENV']['name']
except Exception as emsg:
    print("Strategy name not found. Setting default name 'testing'")
    self.config["name"]="testing"
    debug(emsg)

try: self.max_threads=int(dict(config.items('ENV'))['cpus'])
except:
    print("Error reading num of threads. Setting to default (1)")
    self.max_threads=1

self.config["redis"]=config['ENV']['redis'].strip()

l=0
for inx in config['FACTORS']:
    self.Params.loc[l]=config['FACTORS'][inx].split(",")
    self.Params.iloc[l]['class']=self.Params.iloc[l]['class'].strip()
    if (self.Params.iloc[l]['class']=='d'): #date params handled separately
        self.Params.iloc[l]['initial']=datetime.strptime(self.Params.iloc[l]['initial'].strip(),"%Y-%m-%d")
        self.Params.iloc[l]['current']=datetime.strptime(self.Params.iloc[l]['current'].strip(),"%Y-%m-%d")
        self.Params.iloc[l]['max']=datetime.strptime(self.Params.iloc[l]['max'].strip(),"%Y-%m-%d")
        self.Params.iloc[l]['step']=int(self.Params.iloc[l]['step'])
    else:
        self.Params.iloc[l]['initial']=int(self.Params.iloc[l]['initial'])
        self.Params.iloc[l]['current']=int(self.Params.iloc[l]['current'])
        self.Params.iloc[l]['max']=int(self.Params.iloc[l]['max'])
        self.Params.iloc[l]['step']=int(self.Params.iloc[l]['step'])
    l+=1

except Exception as emsg:
    print(args.config,' file read/parsing error'+str(emsg))
    parser.exit(1)

if args.startdate is not None:
    self.config["startd"] = args.startdate.strftime("%Y-%m-%d")
if args.enddate is not None:
    self.config["enddd"] = args.enddate.strftime("%Y-%m-%d")
if args.mode is not None:
    self.config["mode"] = args.mode.strip()
if args.batch is not None:
    self.config["strategy_batch"] = args.batch
if args.ttitle is not None:
    self.config["symbol"] = args.ttitle
if args.cores is not None:
    self.max_threads = args.cores
if args.showgraph is not None:
    self.config["showgraph"] = args.showgraph.lower()
if args.saveresults is not None:
    self.config["nosave"]=0
    if (args.saveresults=='sumarized'):
        self.config["sumarized_results"]="yes"
        self.config["detail_results"]="no"
    elif (args.saveresults=='detailed'):
        self.config["sumarized_results"]="no"
        self.config["detail_results"]="yes"
    elif (args.saveresults=='all'):
        self.config["detail_results"]="yes"
        self.config["sumarized_results"]="yes"

```

```

elif (args.saveresults=='none'):
    self.config["detail_results"]="no"
    self.config["sumarized_results"]="no"
    self.config["nosave"]=1
else:
    debug('Unknown --saveresults parameter')
    exit(1)

if (self.config["mode"]=='optimize'):
    try:
        self.config['opti_start']=dict(config.items('OPTIMIZATION'))['opti_start']
        self.config['opti_end']=dict(config.items('OPTIMIZATION'))['opti_end']
        self.config['TP']=int(config['OPTIMIZATION']['TP'])
        self.config['OOS']=int(config['OPTIMIZATION']['Q'])

    except Exception as emsg:
        debug('Unknown optimize period')
        debug(emsg)
        exit(1)

def LoadInit(self):
    if (self.max_threads>1):
        try:
            ray.init(address=self.config['redis'], redis_password='*****',
                    driver_object_store_memory=768* 1024 * 1024)
            debug("Ray connected")
        except Exception as emsg:
            debug("ReadConfig: Unable to connect "+self.config['redis'])
            exit(1)

        self.AnalysisList=[Analysis.remote() for i in range(self.max_threads)]
        for i in range(self.max_threads):
            self.AnalysisList[i].DataInit.remote(self.config)

        debug("LoadInit : "+str(i)+" Initialized")
        self.correl=ray.get(self.AnalysisList[0].IsCorrel.remote())
        debug("LoadInit multi.. self.correl="+str(self.correl))

    else:
        self.AnalysisList=[]
        self.AnalysisList.append(Analysis)
        self.AnalysisList[0].DataInit(self.config)
        self.correl=self.AnalysisList[0].IsCorrel()
        debug("LoadInit single ..self.correl="+str(self.correl))

    if (self.correl==0): self.Params=self.Params.loc[~(self.Params['class']=='c')]
    debug("LoadInit : Done ")

def Evaluate(self):
    min_positions=2
    posun=1 # 1~20%, 2~40%....
    try:
        SQL1=("select min(pips),max(pips),count(*) into @mn,@mx,@records "
            "from suma_results sr, strategy_batch sb "
            "where sb.strategy_id=sr.strategy_id and sb.strategy_batch="+str(self.config["strategy_batch"])+" and sr.pips>0 and
sr.positions>"+str(min_positions)+"; ")
        debug(SQL1)

        SQL2=("select floor(pips/((@mx-@mn)/5)),floor(pips/((@mx-@mn)/5))*((@mx-@mn)/5),(1+floor(pips/((@mx-
@mn)/5)))*((@mx-@mn)/5), count(*) "
            "into @inx,@lower,@upper,@cnt "
            "from suma_results sr, strategy_batch sb "
            "where sb.strategy_id=sr.strategy_id and sb.strategy_batch="+str(self.config["strategy_batch"])+" and
sr.pips>((@mx-@mn)/5)*"+str(posun)+" and sr.positions>"+str(min_positions)+" "
            "group by floor(pips/((@mx-@mn)/5)),floor(pips/((@mx-@mn)/5))*((@mx-@mn)/5),(1+floor(pips/((@mx-
@mn)/5)))*((@mx-@mn)/5) "
            "order by 4 desc "
            "limit 1; ")
        debug(SQL2)

        SQL3=("select p.name,p.current,count(*) as cnt FROM suma_results sr, strategy_batch sb, parameters p ")

```

```

        "where sr.pips>=@lower and sr.pips<@upper and sr.positions>" + str(min_positions) + " and
sr.strategy_id=sb.strategy_id and sb.strategy_batch=" + str(self.config["strategy_batch"]) + " "
        "and sr.strategy_id=p.strategy_id and current is not null "
        "group by p.name,p.current "
        "order by 1,3 desc; ")

    debug(SQL3)
    self.db_connection = mysql.connect(host=self.config["my_hostname"], database=self.config["my_results"],
user=self.config["my_username"], password=self.config["my_password"])
    c1 = self.db_connection.cursor()
    c1.execute(SQL1) # mysql user variable are held all the session time
    debug('EVe1')
    c1.execute(SQL2)
    debug('EVe2')
    c1.execute(SQL3)
    debug('EVe3')
    self.matrix=pd.DataFrame(c1.fetchall(), columns=['name','value', 'count']).groupby('name').first()
    SQL=("select p.strategy_id,count(*) from parameters p, strategy_batch sb where sb.strategy_id=p.strategy_id "
        "and sb.strategy_batch=" + str(self.config["strategy_batch"]) + " and ( ")
    debug(SQL)
    for index, value in self.matrix.iterrows():
        SQL=SQL+"(p.name='"+index+"' and p.current='"+str(value['value'])+"' or "
        SQL=SQL[0:-3]+") group by p.strategy_id order by 2 desc limit 1;"
    debug(SQL)
    c1.execute(SQL)
    debug('EVe4')
    strategy_id=c1.fetchall()[0][0]

    recomendations_df=pd.DataFrame([[str(self.config["strategy_batch"]), strategy_id,
self.config["symbol"],self.config["startd"], self.config["endd"]]],
        columns=['strategy_batch', 'strategy_id','symbol','date_from', 'date_to'])
    recomendations_df=recomendations_df.set_index('strategy_batch')

    try:
        connstr="mysql+pymysql://" + self.config["my_username"] + ":" + self.config["my_password"] + "@" + self.config["my_hostname"] + ":" +
        \
            str(self.config["my_port"]) + "/" + self.config["my_results"]
        engine=sqla.create_engine(connstr)
        dbconn=engine.connect()
        recomendations_df.to_sql("recomendations", dbconn, if_exists='append')
        dbconn.close()

    except Exception as emsg:
        print('Error exporting recomendation ')
        print(str(emsg))

    except Exception as emsg:
        print("Evaluation aborted ")
        print(emsg)

    def Verify(self, mode=""):
        startd=self.config["startd"]
        endd=self.config["endd"]
        SQL=("SELECT name,initial,current,class FROM recomendations r, parameters p "
            "where r.strategy_id=p.strategy_id and r.strategy_batch=" + str(self.config["strategy_batch"]) + " and current is not
null;")

        self.db_connection = mysql.connect(host=self.config["my_hostname"], database=self.config["my_results"],
user=self.config["my_username"], password=self.config["my_password"])
        c1 = self.db_connection.cursor()
        c1.execute(SQL)
        self.matrix=pd.DataFrame(c1.fetchall(), columns=['name','initial','current', 'class'])
        for index, values in self.matrix.iterrows():
            if ( values['name'] in self.config):
                self.config[values['name']]=values['current']
            else:
                try : self.Params.loc[self.Params['name']==values['name'], ['current']]=int(values['current'])
                except : self.Params.loc[self.Params['name']==values['name'], ['current']]=values['current']
                try: self.Params.loc[self.Params['name']==values['name'], ['initial']]=int(values['current'])
                except: self.Params.loc[self.Params['name']==values['name'], ['initial']]=values['current']

```

```

        try: self.Params.loc[self.Params['name']==values['name'], ['max']] = int(values['current'])
        except: self.Params.loc[self.Params['name']==values['name'], ['max']] = values['current']
self.config['startd'] = startd
self.config['endd'] = endd
inx = 0
counter = 0
if (mode == ""):
    self.LoadInit()

if (self.max_threads > 1):
    inx = self.NextInx(inx)
    self.AnalysisList[inx].Calc.remote(counter, self.Params, self.config)
#    counting = self.NextParams()
else:
    self.AnalysisList[0].Calc(1, self.Params, self.config)
    if (self.config["showgraph"] == 'yes'):
        self.AnalysisList[0].ShowIndiGraphs(self.config)

def main():
    test = Optimize()
    debug("Class created")
    test.ConfigRead()
    debug("Config file read")

    if ((test.config['mode'] == "optimize")):
        test.LoadInit()
        debug("Optimization Initialized")
        test.DoOptimize()
        print("Optimization finished")

    elif (test.config['mode'] == "verify"):
        debug("Verify Initialized")
        test.Verify()
        debug("Verify finished")

    elif (test.config['mode'] == "evaluate"):
        test.Evaluate()
        debug("Recommendation generated")

if __name__ == "__main__":
    main()

```

9.3 KONTROLA KONZISTENCE STAŽENÝCH DAT

```
#!/bin/bash

dback=$1
echo "Checking $dback days back"
mysql -h 192.168.2.33 -u ***** --password='*****' ticks -se "select distinct title
from _DATATF_YAHOO" | while read line
do
  if [ "$line" != "" ]; then
    SQL="set @MA_period=5; \
set @title='$line'; \
set @dt='1900-01-01'; \
select * from ( \
  select tmp1.id1, tmp1.title, tmp1.date, tmp1.close, avg(tmp2.close) as av from \
    (select @b:=@b+1 as id1,t.title,t.date,t.close from \
      (select title,date,close from _DATATF_YAHOO where title=@title and
date>(now()-interval $dback day)) t,(select @b:=0) bb order by title,date asc) tmp1, \
      (select @a:=@a+1 as id2,t.title,date,close from \
        (select title,date,close from _DATATF_YAHOO where title=@title and
date>(now()-interval $dback day)) t,(select @a:=0) aa order by title,date asc) tmp2 \
      where id1>@MA_period and id1>=id2 and id2>(id1-@MA_period) \
      group by id1,tmp1.title,tmp1.date,tmp1.close \
      order by title,date asc \
    ) tmp where close>1.15*av or close<0.85*av;"
    mysql -h 192.168.2.33 -u ***** --password='*****' ticks -se "$SQL"
  fi
done
```