TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Helene Lillemägi 182842IABM

AEGRUUMI KUUBI JÄRJESTAMINE JA SELLE RAKENDAMINE FINANTSTURUL

Magistritöö

Juhendaja: Innar Liiv
PhD

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Helene Lillemägi

10.05.2021

Annotatsioon

Käesoleva magistritöö teemaks on aegruumi kuubi järjestamine. Töö peamiseks eesmärgiks on välja töötada üks võimalik meetod suure ajalist järjestust sisaldava kolmemõõtmelise andmekogu ehk aegruumi kuubi järjestamiseks. Täiendavaks ja peamist eesmärki toetavaks eesmärgiks on järjestatud kuubi visualiseerimine. Töös kasutatakse S&P 500 indeksisse kuuluvate aktsiate 5 aasta päevaseid sulgemishindasid, mille põhjal genereeritakse väärtpaberite omavaheliste tugevate korrelatsiooniseoste põhjal aegruumi kuup.

Järjestamise eksperiment viiakse läbi rakendades kuubi ühele kihile (baaskiht) valitud järjestusalgoritmi, millega koos järjestuvad ümber ka kõik ülejäänud kuubi kihid. Parim kuubi järjestus selgitatakse välja praktilise katsetuse tulemusena selliselt, et iga kuubi kiht on olnud korra baaskihiks, millele on rakendatud erinevaid järjestamise algoritme. Autori poolt püstitatakse parima baaskihi leidmise hüpotees, mis leiab ka valideerimisel kinnitust.

Andmete töötlemine ja eksperiment on läbi viidud statistikatarkvara R jaoks integreeritud arenduskeskkonnas RStudio. Samuti on R-ga interpreteeritud eksperimendi esmane tulemus. Lisaks on lühidalt tutvustatud programmi Cubix visualiseerimise võimalusi, mille kasutamine ei eelda programmeerimise oskust.

Töö tulemusena valmib praktiliselt kasutatav meetod aegruumi kuubi järjestamiseks ning tulemi visualiseerimiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 40 leheküljel, 6 peatükki, 35 joonist, 2 tabelit.

Abstract

Reordering of the Space-Time Cube and its Application to Financial Markets

The topic of this master's thesis is the reordering of the space-time cube. The main goal of the work is to develop one possible method for reordering a large three-dimensional database or space-time cube containing a chronological sequence. An additional goal that supports the main goal is to visualize the reordered cube. The paper uses the 5-year daily closing prices of the shares included in the S&P 500 index, on the basis of which a space-time cube is generated on the basis of strong correlations between the securities.

The reordering experiment is performed by applying a selected seriation algorithm to one slice of the cube (base slice), with which all other slices of the cube are rearranged. The best cube sequence is determined by practical experimentation so that each slice of the cube has once been the base slice to which seriation algorithms have been applied. The author also puts forward the hypothesis of finding the best base slice, which is also confirmed during validation.

Data processing and experimentation were performed in the development environment RStudio integrated for the statistical software R. The primary result of the experiment is also interpreted with R. In addition, the possibilities of visualizing the Cubix program, the use of which does not require programming skills, are briefly introduced.

As a result of the work, a practical method for reordering the space-time cube and visualizing the result is completed.

The thesis is in Estonian and contains 40 pages of text, 6 chapters, 35 figures, 2 tables.

Lühendite ja mõistete sõnastik

2D	*Two-dimensional* ehk kahemõõtmeline ehk tasapinnaline
3D	*Three-dimensional* ehk kolmemõõtmeline ehk ruumiline
BEA	*Bond Energy Algorithm* ehk sidemeenergia algoritm
ETF	*Exchange-traded fund* ehk börsil kaubeldav fond
Korrelatsioon	Kahe muutuja vaheline seos
Maatriks	Teist järku tensor, mida saab esitada kahemõõtmelise massiivina
Massiiv	Andmete hulk, millel igaühel on oma indeks (järjekorranumber)
MDS	*Multidimensional Scaling* ehk mitmemõõtmeline skaleerimine
ME	*Measure of Effectiveness* ehk efektiivsusmõõdik
MP4	*MPEG-4 Part 14* ehk multimeedia konteinerformaat
PCA	*Principal Component Analysis* ehk peakomponentanalüüs
PNG	*Portable Network Graphics* ehk porditav võrgugraafika
Sümmeetriline maatriks	Ruutmaatriks lineaaralgebras, mis langeb kokku oma transponeeritud maatriksiga ja mille elemendid asetsevad selle peadiagonaali suhtes sümmeetriliselt
Tensor	Matemaatiline objekt lineaaralgebras, mis üldistab skalaari, vektori, maatriksi ja bilineaarse vormi mõistet
Tensori järk	Tensori esitamiseks vajaliku massiivi mõõde
Transponeeritud maatriks	Maatriks lineaaralgebras, mis saadakse maatriksi *A* ridade ja veergude vahetamisel

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Korrelatsioonianalüüs on paljude finantsteooriate ja -praktikate (näiteks Markowitz'i modernne portfelliteooria, finantsriskide uurimine jt) üheks oluliseks aspektiks. Investeerimisriski maandamiseks on vaja põhjalikku analüüsi muutuvate korrelatsioonide kohta, mille puhul on tarvis tuvastada sellised finantsvarade vahelised positiivsed ja negatiivsed seosed, mis on olulised. Sharpe'i sõnul suureneb portfelli mitmekesistamise tagajärjel portfelli üldise riski vähenemine ehk varade korrelatsiooni ja riski vahel on tugev seos [1].

Süstemaatiline korrelatsioonide jälgimine on portfelli riskide juhtimisel olulise tähtsusega, kuid seda on üha keerulisem teostada, kuna võimalike seoste hulk on ajaga hüppeliselt kasvanud. Keskmise suurusega investeerimisportfellis (144 väärtpaberit) on üle 10 000 paarikaupa seose [1].

Cook'i, Soramäki ja Laubsch'i [2] hinnangul on finantsturgude riskijuhtimiseks oluline mõista suurte korrelatsioonide struktuure ja nende muutumist aja jooksul. Tihti kasutatakse suurte korrelatsioonimaatriksite minimaalse toese või tasapinnalise maksimaalselt filtreeritud graafide visualiseerimiseks ja klastrite moodustamiseks korrelatsioonide filtreerimist. Kui aga visualiseeritakse korrelatsioonimaatriksite rea jada, näitab visualiseerimine palju "hüppeid" ja on raske näha ning mõista, kuidas korrelatsiooni struktuur aja jooksul muutub.

Käesolev töö pakub välja ühe võimaliku meetodi suurte korrelatsioonimaatriksite rea jada järjestamiseks ning lõpptulemi visualiseerimiseks.

1.1 Eesmärgid

Töö peamiseks eesmärgiks on välja töötada üks võimalik meetod suure ajalist järjestust sisaldava kolmemõõtmelise andmekogu ehk aegruumi kuubi järjestamiseks. Aegruumi kuup saadakse, kui portfelli kuuluvate varade omavahelisi ajas muutuvaid korrelatsioone vaadeldakse kui kolmemõõtmelist andmekogu, mis on moodustatud kuise intervalliga

koostatud kahemõõtmeliste binaarsete korrelatsioonimaatriksite liitmise tulemusena. Kuup, mis on järjestatud nii, et väärtpaberite omavaheliste tugevate korrelatsiooniseoste olemasolu ja muutumist ajas on võimalik visuaalselt interpreteerida, on kasulik näiteks finantsanalüütikute jaoks, kuna aitab teha kiireid järeldusi ja põhjendatud kitsendusi andmete valikul edasises analüüsis, mis hoiab kokku väärtuslikku aega. Sellest tulenevalt on töö täiendavaks ja peamist eesmärki toetavaks eesmärgiks järjestatud aegruumi kuubi visualiseerimine.

Sõltuvalt valitud lähenemise viisist on eesmärgini jõudmiseks püstitatud järgmised uurimisküsimused:

- Kuidas genereerida suurest andmehulgast aegruumi kuup? Millised elemendid peaksid olema kuubis esindatud ja millised mitte?
- Kuidas järjestada aegruumi kuupi? Kas selleks saab kasutada mõnda olemasolevat järjestusmeetodit? Milline meetod võimaldab aegruumi kuupi järjestada sellisest, et säiliks andmete ajaline järgnevus? Millise kihi järgi järjestamine annab parima tulemuse?
- Kuidas ja millise rakendusega visualiseerida suurt aegruumi kuupi? Kas visuaal on kergelt interpreteeritav?

1.2 Ülevaade tööst

Töö baseerub disainiteaduse metoodikal (ingl *Design Science*), mille puhul püstitatakse esmalt peamine uurimisküsimus. Uuritava valdkonna hetkeolukord kaardistatakse muuhulgas teaduskirjanduse abil, millega ühtlasi veendutakse, et sellist tehist nagu soovitakse luua veel ei ole, kuid vajadus selleks on olemas. Algselt püstitatud probleem lahendatakse parimaid praktikaid, tehtud vigu ja teaduslik-tehnilisi edusamme arvesse võttes uue infotehnoloogilise tehise loomisega, mille headust hinnatakse formaalsete meetodite abil [3].

Käesoleva magistritöö esimeses, teoreetilises osas tutvustatakse kolmemõõtmelise andmekogu ehk tensori olemust ja selle peamisi omadusi ning selgitatakse aegruumi kuubi mõistet ja selle seost tensoritega. Seejärel antakse ülevaade varasematest uuringutest, mis käsitlevad mitmemõõtmeliste andmekogude järjestamist ja aktsiaturgude korrelatsioonide visualiseerimist. Teoreetilise osa lõpetuseks tutvustatakse lühidalt korrelatsioonianalüüsiga seotud teemakohaseid rakendusi.

Töö praktiline osa on valdavalt läbi viidud tarkvaraga R ning loodud terviklik lähtekood on toodud Lisas 2. Läbi viidud eksperiment on esitatud käesolevas töös kolme peatükina.

Eksperimendis kasutatavad andmed. Praktilise töö esimeses osas genereeritakse lähteandmetest, milleks on aktsiate päevased sulgemishinnad, järjestamiseks ja visualiseerimiseks vajaminev binaarseid korrelatsioonimaatrikseid sisaldav aegruumi kuup. Andmetöötlus sisaldab järgmisi etappe: mittetäieliku ajalooga andmete eemaldamine, päevaste tootluste leidmine, korrelatsioonimaatriksite koostamine, tugevate korrelatsiooniseoste põhjal binaarsete maatriksite moodustamine ning selle juhuslikest seostest ehk mürast puhastamine. Arvestada tuleb, et tugevad positiivsed ja tugevad negatiivsed korrelatsioonid moodustavad erinevad aegruumi kuubid.

Aegruumi kuubi järjestamise eksperiment. Praktilise töö teises ja peamises osas teostatakse aegruumi kuubi järjestamine selliselt, et säilib andmete ajaline järgnevus. Lähtuvalt sellest kitsendusest tutvustatakse üldiselt ühte võimalikku järjestamise meetodit ning selleks kasutatavaid algoritme. Kuna antud meetodi puhul ei ole järjestamise alguspunkt (järjestamise aluseks olev baaskiht) valideeritud, siis testitakse eksperimendi käigus läbi kõik võimalused. Nimetatud probleemi lahendamiseks sõnastab töö autor parima baaskihi määramise hüpoteesi ning rakendab seda ka antud eksperimendis. Järjestamisprotsessi vältel salvestatakse maha visualiseerimise ja valideerimise jaoks vajalikud vahetulemused. Antud praktilise osa lõpuks esitletakse järjestatud aegruumi kuubi visualiseerimise tulemusi, mis on loodud rakendustega R ja Cubix.

Tulemuste valideerimine ja analüüs. Praktilise osa viimases peatükis selgitatakse esmalt lähemalt valideerimiseks kasutatavate mõõdikute sisu ning seejärel teostatakse saadud võimalike järjestuste valideerimine. Valideerimise tulemusel antakse hinnangud nii kasutatud järjestusalgoritmidele kui ka kasutatud valideerimismõõdikutele. Saadud valideerimistulemustele tuginedes hinnatakse lisaks ka praktilise töö teises osas sõnastatud ja testitud hüpoteesi parima järjestuse leidmiseks. Peatüki lõpus analüüsitakse eksperimenti erinevate kriteeriumite põhjal ning tuuakse välja tekkinud kitsaskohad ning võimalikud edasised arengukohad.

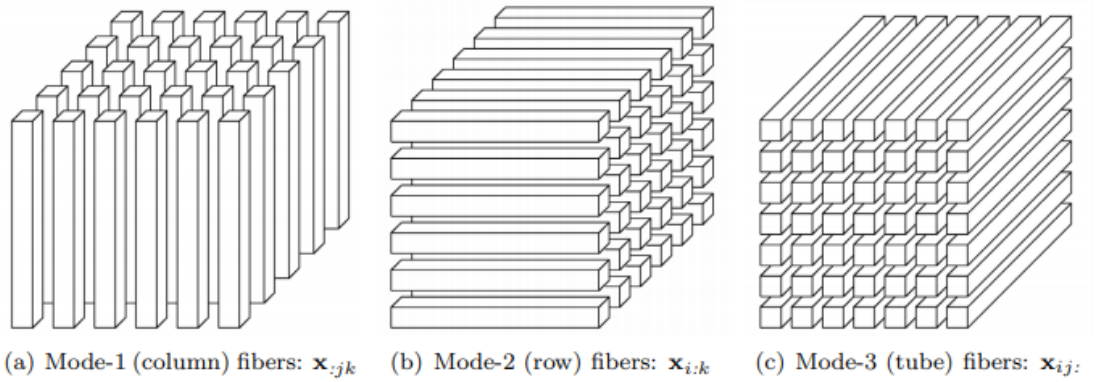
2 Teoreetiline raamistik ja seotud rakendused

Käesolevas peatükis antakse ülevaade kolmemõõtmelisest andmehulgast kui tensorist, selle kasutamisest mitmemõõtmeliste andmehulkade esitamisel ning järjestamisest. Ülevaate käigus tutvustatakse ka aegruumi kuubi mõistet ning seost tensoritega. Lisaks on kajastatud autori hinnangul olulisemaid teadustöid aktsiate ja aktsiaturgude korrelatsioonide valdkonnast ning toodud näiteid korrelatsioonianalüüsi rakendustest.

2.1 Tensor

Algebras kasutatakse tensori mõistet üldistamaks mitmemõõtmelisi massiive. Tensorid leiavad rakendust enam kui kahemõõtmeliste ja mitmeseoseliste (ingl *multirelational*) või mitmepoolsete seostega (ingl *multiway relational*) massiivide puhul. Skalaar, vektor ja maatriks, kui lihtsamat tüüpi tensorid, on vastavalt 0-järku, 1-järku ja 2-järku tensorid. Tensori järk näitab selle esitamiseks vajaliku massiivi mõõdet ehk *k*-järku tensor on *k*-mõõtmeline massiiv [4].

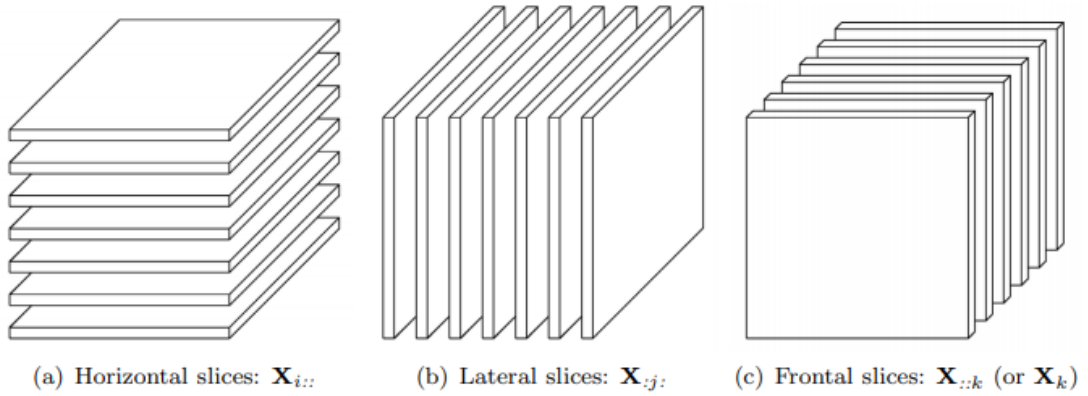
Analoogiliselt vektorite ja maatriksitega tähistatakse 3-järku tensori $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ elementi (i, j, k) kujul x_{ijk}. Tensoritest saab moodustada alammassiive fikseerides indeksite mingi alamhulga. Maatriksi $\boldsymbol{A}$ j-nda veeru elemendid on alamhulk elementidega $a_{:j}$ ning fikseerides reaindeksi i saame alamhulga elementidega $a_{i:}$. Sarnaselt maatriksi ridadele ja veergudele nimetatakse kõrgemat järku tensorites vastavaid alammassiive kiududeks (ingl *fiber*). Kiud saadakse, kui tensoris fikseeritakse kõik indeksid peale ühe. Joonisel 1 on kujutatud 3-järku tensoris eksisteerivad veeru $\left(x_{:jk}\right)$, rea $(x_{i:k})$ ja toru (ingl *tube*) $\left(x_{ij:}\right)$ kiud [4].



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

Joonis 1. Tensori kiud.

Kihid (ingl *slice*) on tensori kahemõõtmelised lõiked (maatriksid), mis saadakse, kui fikseeritakse kolmest indeksist üks. Joonis 2 illustreerib tensori $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ kihte ehk maatrikseid $\boldsymbol{X}_{i::}$, $\boldsymbol{X}_{:j:}$ ja $\boldsymbol{X}_{::k}$ (tähistatakse ka kui $\boldsymbol{X}_k$) [4].



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

Joonis 2. 3-järku tensori kihid.

Tensorit nimetatakse sümmeetriliseks (kirjanduses kasutatakse sageli ka mõistet supersümmeetriline), kui tema elemendid jäävad samaks iga indeksite permutatsiooni puhul. Ehk iga *i*, *j*, *k* korral kehtib $x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji}$ [4].

Tensor saab olla ka osaliselt sümmeetriline (ingl *partially symmetric*). Näiteks 3-järku tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ on osaliselt sümmeetriline, kui iga kihi $\boldsymbol{X}_{::k}$ puhul on vastav maatriks sümmeetriline $\boldsymbol{X}_k = \boldsymbol{X}_k^T$ ehk maatriks võrdub enda transponeeritud maatriksiga [4].

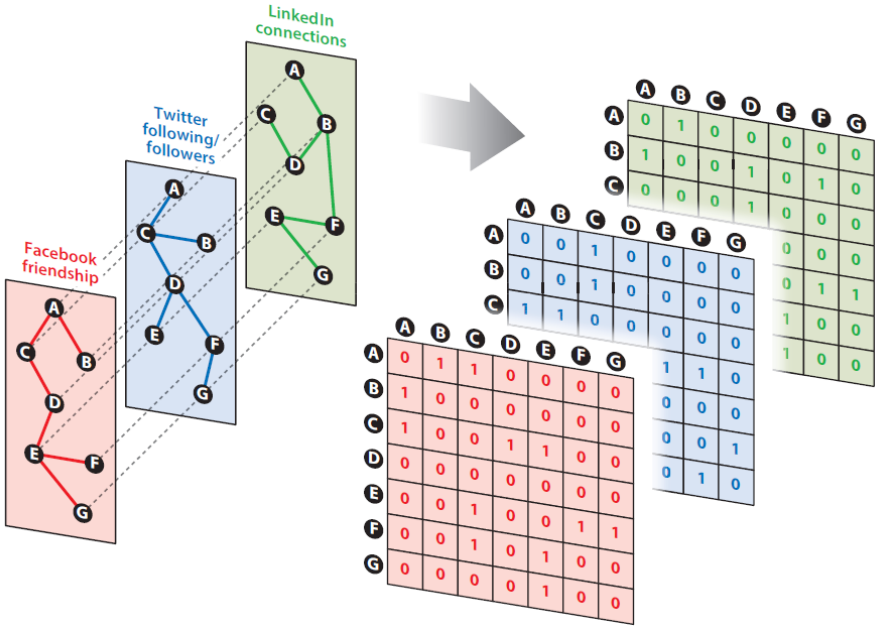
2.2 Tensori kasutamine mitmemõõtmeliste andmehulkade esitamisel

Käesolevas töös analüüsitavat andmehulka saab esitada 3-järku tensorina $\mathcal{X} = \{x_{ijk}\} \in \mathbb{R}^{n \times \mathrm{n} \times \mathrm{m}}$, kus tensori element x_{ijk} näitab *i*-nda ja *j*-nda aktsia korrelatsiooni ajahetkel *k*.

Antud tensor on osaliselt sümmeetriline, kuna iga fikseeritud ajahetke *k* korrelatsioonimaatriks on sümmeetriline.

Algsest andmehulgast moodustatakse binaarsed korrelatsioonimaatriksid iga fikseeritud ajahetke *k* kohta selliselt, et väga tugevad seosed kahe aktsiapaari vahel tähistatakse väärtusega 1 ning ülejäänud väärtusega 0. Kuna korrelatsioon iseendaga ei ole ei analüüsi ega visualiseerimise seisukohast oluline, siis tähistatakse ka need nulliga. Saadakse 3-järku binaarsete elementidega tensor $\mathcal{X} \in \{0,1\}^{n \times n \times m}$, kus element x_{ijk} näitab, kas ajahetkel *k* on *i*-nda ja *j*-nda aktsia omavaheline korrelatsioon väga tugev või mitte.

Artiklis „Tensorid statistikas“ [5] alapunktis 5 näitlikustatakse erinevates sotsiaalvõrgustikes (erinevad kihid) kasutajate sõbrastaatuste (1 või 0) andmeid (Joonis 3).

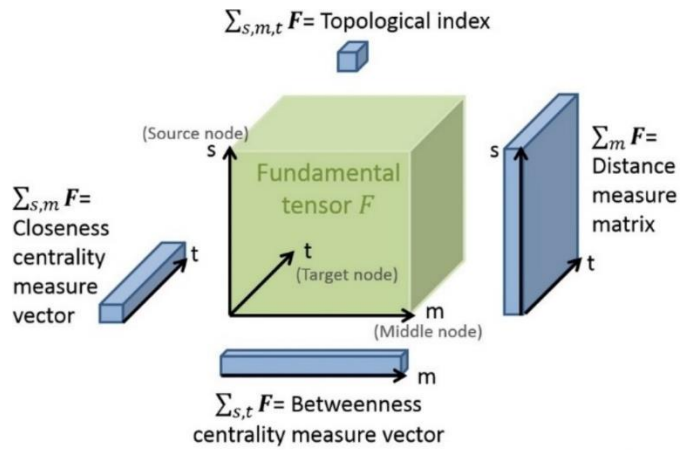


Joonis 3. Tensori moodustamise illustratsioon.

Konkreetse sotsiaalvõrgustiku andmeid on algselt kujutatud graafina, kus tippudeks on kasutajad (hulk $V = \{A, B, C, D, E, F, G\}$) ning servad kujutavad sõbrastaatust. Seoste põhjal genereeritakse binaarsed maatriksid, mis omakorda moodustavad 3-järku tensori kihid. Nii käesoleva töö kui ka sotsiaalvõrgustiku näite puhul on tegemist mitmeseoselise võrguga (ingl *multrirelational network*), kus iga serv ühendab antud kihis kahte tippu (sõbrastaatus kasutajate vahel või väga tugev korrelatsioon aktsiate vahel).

Golnari, Zhang ja Boley [6] illustreerivad tensori kasulikkust ja tõhusust võrgustiku analüüsis juhusliku ekslemise ja Markovi ahela näitel. Nad defineerivad fundamentaalse

maatriksi laiendusena kolmemõõtmelise fundamentaalse tensori F_{smt}, kus s on algtipp, m keskmine ning t sihttipp, ja mis tähistab eeldatavat arvu kordi, mil Markovi ahel läbib tippu m, kui alustab tipust s ja jõuab esimest korda tippu t. Joonisel 4 on kujutatud seost Markovi fundamentaalse tensori ja nelja erineva mõõdiku arvutusi ühendava raamistiku vahel.



Joonis 4. Markovi fundamentaalne tensor ja mõõdikute arvutusi ühendav raamistik.

Olgu tensorite kasutamise põhjuseks mitmeseoselised või suuremahulised andmehulgad, enamasti on sooviks andmeid korrastada selliselt, et need oleksid kompaktsemad, paremini analüüsitavamad ja visualiseeritavamad.

2.3 Mitmemõõtmeliste andmehulkade järjestamine

Hõre tensor on sageli loomulik viis mitmeteguriliste (ingl *multifactor*) või mitmeseoseliste andmehulkade esitamiseks ning seda on rakendatud andmete analüüsimisel ja kaevandamisel muuhulgas tervishoiu, loomuliku keele töötlemise, masinõppe ja sotsiaalvõrgustike analüüsi valdkondades. Tensorit nimetatakse hõredaks, kui enamus tema elemente on nullid. Hõreda tensori (ümber)järjestamist elementide indeksite ümberpaigutamise tulemusena ühes või mitmes tensori kihis (dimensioonis) on kasutatud näiteks mälu asukohaviidete parendamiseks tensorarvutuste tegemisel [7].

Kui üldjuhul on tensori kihid järjestuste osas sõltumatud, siis käesolevas töös eeldatakse, et ajahetki tähistavad tensori kihid $\boldsymbol{X}_k$ jäetakse andmete järjestamisel algsesse järjestusse. Sarnast lähenemist on kasutanud dünaamiliste võrkude visualiseerimisel ka Bach, Pietriga ja Fekete [8], kes nimetavad sellist andmete käsitlust aegruumi kuubiks (ingl *space-time cube*), mille defineerivad järgnevalt: aegruumi kuup on kujutis, mis kaardistab

andmed kahes mõõtmes, samal ajal kui aega näidatakse kolmanda ruumilise mõõtmena. Aegruumi kuubid on võimelised näitama nii ruumilist kui ka ajalist teavet üheaegselt.

I. Liiv [9] defineerib järjestamise (ingl *seriation*) kui avastusliku kombinatoorse andmeanalüüsi tehnika objektide seadmiseks järjendisse mööda ühemõõtmelist kontiinumi selliselt, et see paljastaks kogu seerias maksimaalselt regulaarsust ja mustreid. Lisaks toob Liiv välja, et mitmemõõtmelist järjestamist võib vaadelda kui mitme andmehulga samaaegset järjestamist, mille käigus erinevate andmehulkade elemendid omavahel ei segune ning iga andmehulk säilitab eraldiseisva ühemõõtmelise kontiinumi.

Valdavalt on aga teaduskirjanduses mitmemõõtmeliste andmekogude järjestamist puudutavates artiklites kasutatud mõistet klasterdamine (ingl *clustering*), mille alamjuht järjestamine tegelikult on. Kui klasterdamine jagab elemendid mingi tunnuse alusel gruppidesse, siis järjestamine määrab iga elemendi asukoha järjendis [10]. Ehk et järjestamine annab grupi elementide kohta lisainfot selle kohta, kuidas on elemendid omavahel ning erinevad grupid omakorda omavahel seotud.

Alamruumi klasterdamine on sisuliselt traditsioonilise klasterdamise laiendus ja selle eesmärk on leida klastreid andmekogumi erinevates alamruumides. Alamruumi klasteralgoritmid lokaliseerivad asjakohaste dimensioonide otsingu, võimaldades neil leida klastreid, mis eksisteerivad mitmes võimalikus kattuvas alamruumis. Otsimisstrateegia põhjal jaguneb alamruumi klasterdamine kaheks. Ülalt alla algoritmid leiavad esialgse klasterduse kõikide dimensioonide komplektidest ja hindavad seejärel iga klastri alamruume, parandades tulemusi iteratiivselt. Alt üles suunatud lähenemisviisid leiavad madalamõõtmelistest ruumidest tihedaid piirkondi ja ühendavad need klastrite moodustamiseks [11].

McCallum, Nigam ja Ungar [12] on välja pakkunud ühe efektiivse viisi mitmemõõtmelise andmehulga klasterdamiseks. Näitena võetakse andmestik, mis on suur kõigis kolmes suunas korraga ja koosneb miljonitest andmepunktidest, mis eksisteerivad tuhandetes mõõtmetes ja esindavad tuhandeid klastreid. Nende meetodi kohaselt jagatakse andmed esmalt ligikaudse kauguse mõõdu abil kattuvateks alamhulkadeks (laotused) ning seejärel klasterdatakse ühte laotusesse sattunud punktid täpse kauguse mõõdu järgi.

Lisaks eelnevale on mitmemõõtmelisi andmehulki võimalik järjestada dimensioonide vähendamise tehnikate abil, mille keskseks eesmärgiks on tekitada ridade ja veergude

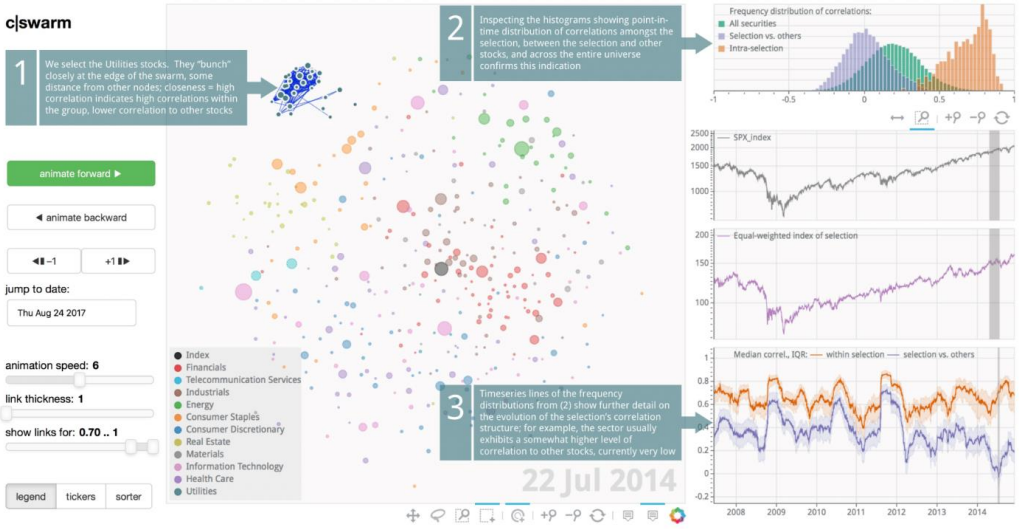
ühemõõtmeline paigutus/järjestus, mis kajastab olulisemaid (mitte)lineaarseid ridade ja veergude vahelisi seoseid. Dimensioonide vähendamise peamisteks tehnikateks on peakomponentanalüüs (ingl *Principal Component Analysis*) ja selle variatsioonid ning mitmemõõtmeline skaleerimine (ingl *Multidimensional Scaling*) [13].

2.4 Aktsiaturgude korrelatsioonide visualiseerimine

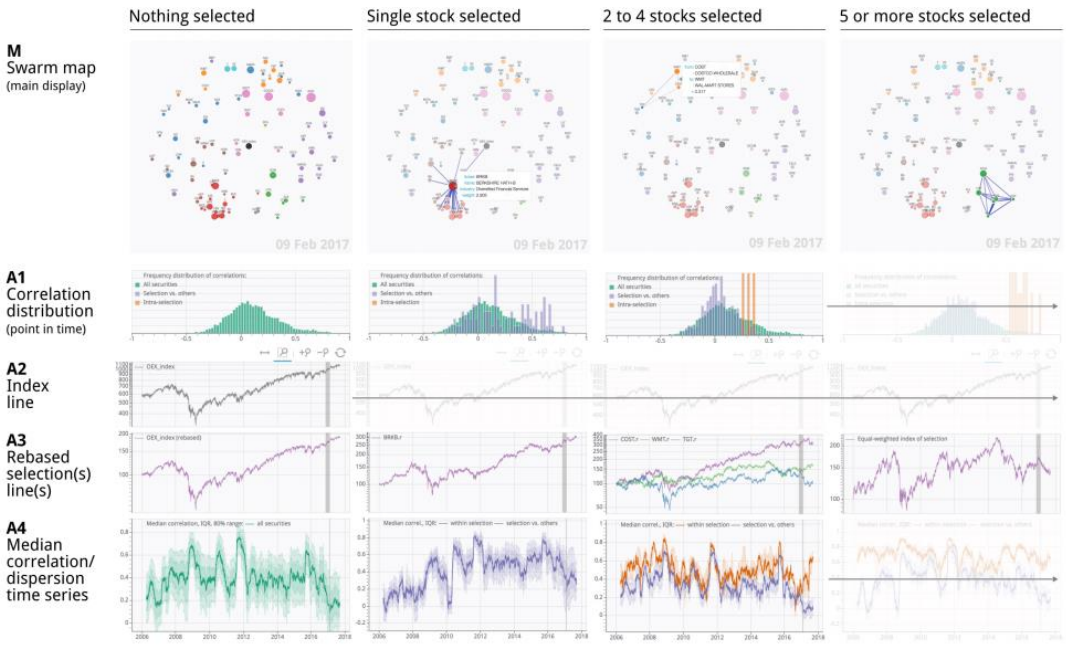
Viimase kahe aastakümne jooksul on finantsturgude korrelatsioonide, hierarhiate, võrgustike ja klastrite teemal avaldatud ligi pool tuhat uurimustööd. Standardse ja laialt levinud meetodi kohaselt teisendatakse paaridevahelised korrelatsioonid esmalt kaugusteks ning seejärel moodustatakse kaugustest minimaalne toes (ingl *Minimum Spanning Tree*), mille lõpptulemusena saadakse ja visualiseeritakse graaf [14].

Simon ja Turkay [1] pakuvad välja uudse visuaalse analüüsi raamistiku korrelatsiooniseoste ja -struktuuride interaktiivseks analüüsiks mitmemõõtmelistes dünaamilistes andmekogudes. Kasutades näitena väärtpaberiturgude dünaamiliste korrelatsioonide analüüsi, kavandavad ja konstrueerivad nad interaktiivseid visualiseerimise lahendusi, mis sobivad positiivsete ja negatiivsete korrelatsiooniseoste samaaegseks tuvastamiseks (ja üldisemalt kõrgemõõtmeliste andmehulga paarikaupa kauguste visuaalseks analüüsiks) ja nende muutuste jälgimiseks ajas. Nende käsitlusviis pakub efektiivset lahendust, võimaldades analüütikutel uurida investeerimisportfellide ja mitmesajast väärtpaberist koosnevate indeksite siseseid korrelatsiooniseoseid eesmärgiga avastada ja analüüsida korrelatsioonistruktuuride muutuseid ajas.

Programm c|swarm võimaldab korrelatsiooniseoseid uurida terve portfelli, sealhulgas ka sektorite (Joonis 5), aga ka üksikute aktsiate põhiselt (Joonis 6) [1].

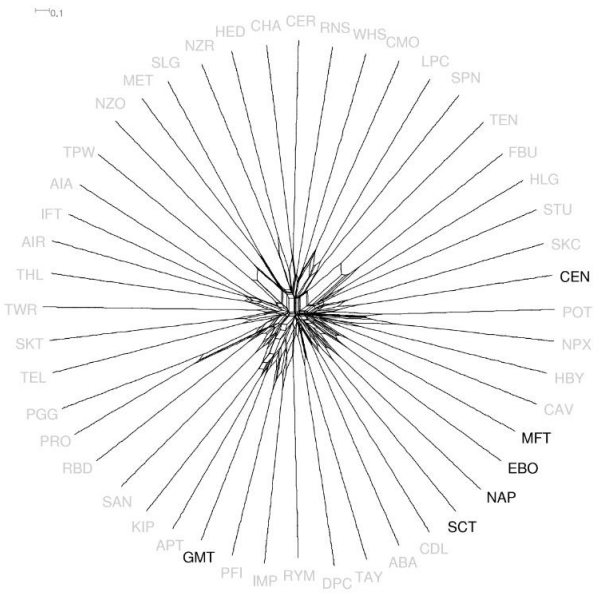


Joonis 5. Programmi c|swarm portfellipõhine vaade.



Joonis 6. Programmi c|swarm kontekstipõhine vaade.

Varasemalt on aktsiaturgude korrelatsioonide visualiseerimisel kasutanud tavapärasest erinevat lähenemist ka A. Rea ja W. Rea [15], kes rakendasid finantsandmetele esmakordselt naabervõrgu (ingl *Neighbor-Nets*) teooriat. Oma töös pakuvad nad välja mooduse hajutatud portfelli koostamiseks, kasutades selleks naabervõrkude jagatud graafide meetodit SplitsTree programmis. Joonis 7 illustreerib sel viisil 48 Uus-Meremaa börsil noteeritud aktsia hulgast tuvastatud kuute kõige tugevamini korreleeruvat aktsiat.



Joonis 7. SplitsTree programmi väljund.

Valdkonna kõige hilisema uurimustöö on avaldanud Groenen ja Franses [16] aastal 2000, kes pakkusid välja taaskord ühe graafidel põhineva kirjeldava meetodi, kuid seda seekord rahvusvaheliste aktsiaturgude omavaheliste võimalike ajas muutuvate seoste visuaalseks analüüsiks. Nende hinnangul võiks meetod olla kasulik sarnase käitumisega aktsiaturgude stabiilsete või siis tekkivate klastrite jälgimiseks. Artiklis tutvustatakse programmis MATLAB koostatud meetodit 13 olulise aktsiaturu tootluse ja absoluuttootluse jaoks, millele rakendatakse mitmemõõtmeliste skaleerimismeetodite (edaspidi MDS) tehnikaid.

Artiklis toodud käsitlust võib mõnes mõttes pidada kõige sarnasemaks käesoleva töö lähenemisega. Protseduur, mida kasutatakse, on järgmine. Esiteks määratletakse akna pikkus. Järgmisena arvutatakse muutujate vahelised korrelatsioonid. Saadud 13 x 13 korrelatsioonimaatriks esitatakse seejärel graafiliselt MDS abil. Järgmises etapis nihutatakse akna pikkust ühe või mitme päeva võrra. Selle uue perioodi jaoks arvutatakse uued korrelatsioonid ja MDS saab uue graafilise kujutise. Saadud kujutis määratleb teise kaadri. Neid samme korratakse seni, kuni kõik andmed on töödeldud. Lõpuks esitatakse lühifilm MDS-piltide järjestusest, mille abil on võimalik ajas muutuvaid dünaamilisi mustreid jälgida [16].

2.5 Olemasolevad korrelatsioonianalüüsi vahendid

Lisaks eelnevale on korrelatsioone võimalik analüüsida ka järgnevas loetelus toodud rakendustega. Loetelu ei ole lõplik, kuid on autori hinnangul piisav ilmestamaks selle valdkonna võimalusi ja vahendeid.

2.5.1 Bloomberg Terminal

Bloomberg Terminal on ettevõtte Bloomberg L.P. poolt pakutav arvutitarkvara süsteem, mis võimaldab finantsteenuste sektori ja muude majandusharude spetsialistidel pääseda juurde Bloombergi professionaalsetele teenustele. Terminali kaudu saavad kasutajad jälgida ja analüüsida reaalajas toimivaid finantsturu andmeid ning teha tehinguid elektroonilisel kauplemisplatvormil [17].

Nimetatud programmis on võimalik funktsiooni CORR abil koostada meelepärastest väärtpaberitest 10 rea ja 10 veeruga korrelatsioonimaatriks, mille tulemusena kuvatakse kasutajale joonisel 8 kujutatud tabel.

Joonis 8. Bloomberg Terminali funktsiooni CORR väljund.

Antud tabelist olulise informatsiooni leidmine võib aga osutuda üpris tülikas ja ajamahukas kuna rakenduse spetsiifiline kasutajaliidese disain ei paku kasutajale võimalust tulemuste kiireks visuaalseks eristamiseks (näiteks värvide järgi).

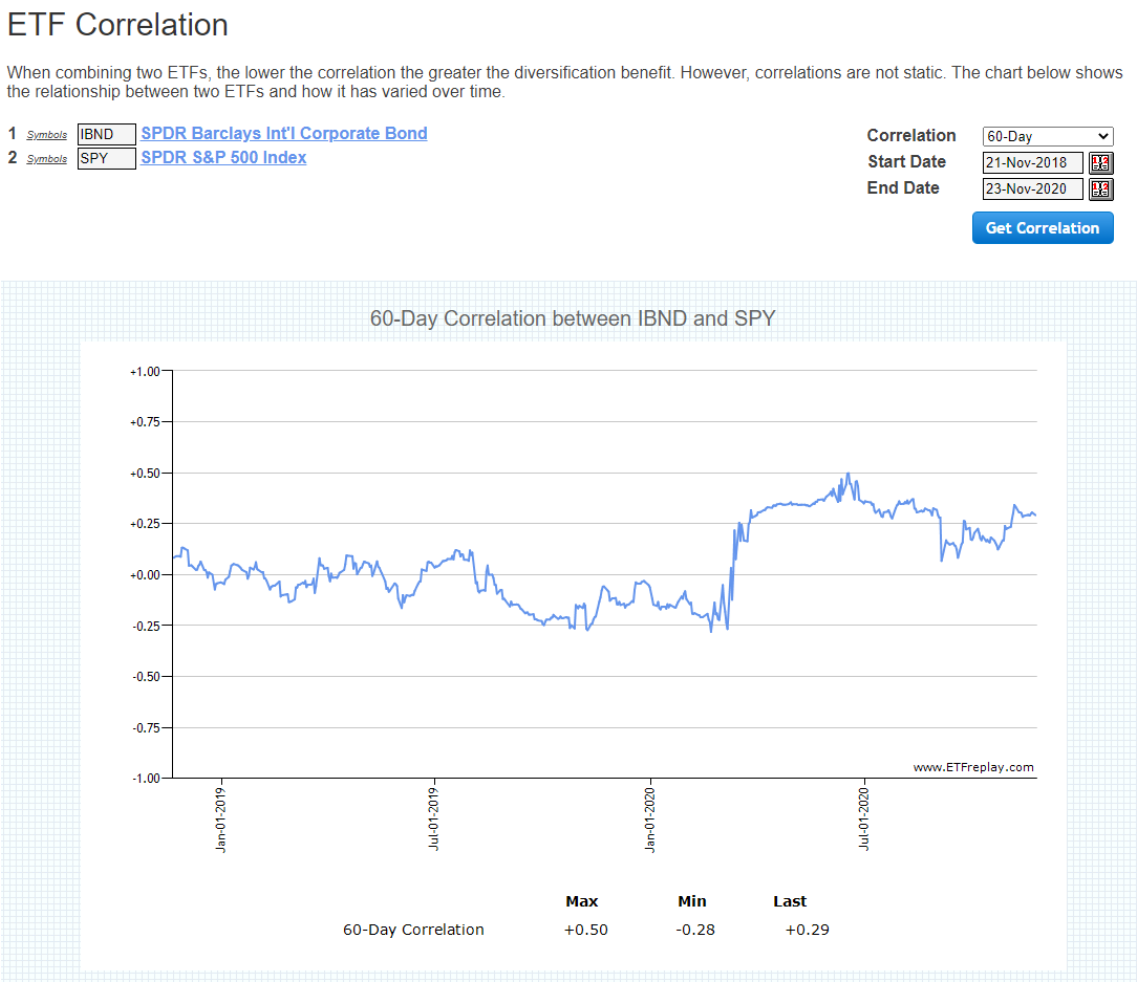
Lisaks eelnevale sisaldab Bloombergi ajalooline regressioonianalüüs (ingl *Historical Regression*) muuhulgas kahe finantsinstrumendi omavahelisi korrelatsiooninäitajad R ja R^2 (Joonis 9) ning uudiste trendide (ingl *News Trends*) graafik pakub võimalust otsida seoseid uudiste ja väärtpaberi hinnaliikumiste vahel (Joonis 10) [18].

Joonis 9. Bloomberg Terminali regressioonianalüüs.

Joonis 10. Bloomberg Terminali uudiste trendide graafik.

2.5.2 ETFreplay.com

ETFreplay.com on veebirakendus, mis keskendub väärtpaberituru loodud teabe kasutamisele koos portfellihalduse tehnikatega, et suurendada hallatavate portfellide tootlust ja vähendada riski [19]. Lisaks mitmetele muudele analüüsivahenditele pakub ETFreplay.com ka korrelatsioonianalüüsi tööriista kahe börsil kaubeldava fondi (ingl *Exchange-Traded Fund* ehk ETF) vahelise seose graafiliseks kujutamiseks. Joonisel 11 on toodud ekraani kuvatõmmis kahe ETF-i (IBND ja SPY) ETFreplay.com poolt pakutavast korrelatsiooniseose graafikust. Antud rakenduse puudusteks võib pidada asjaolusid, et valik on piiratud vaid ETF-dega ning huvipakkuvate fondide võrdlemine on võimalik vaid paarikaupa.

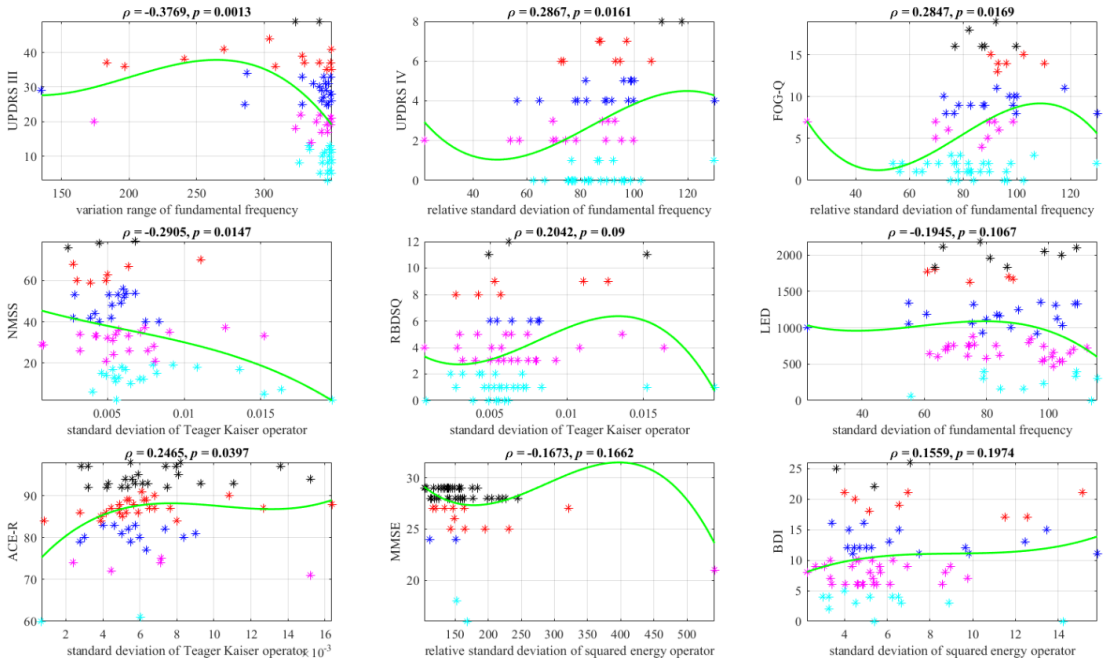


Joonis 11. ETFreplay.com kuvatõmmis.

Sarnaseid paarikaupa korrelatsioonianalüüsi teostavaid veebipõhiseid rakendusi leidub veel mitmeid teisigi. Erinevus võib küll peituda võrreldavates väärtpaberites (lisaks ETF-dele võib mujalt leida ka aktsiaid ja muid investeerimisfonde), kuid üldjuhul on nende töötamise põhimõtted siiski sarnased ja kahemõõtmelisest joonisest midagi enamat kasutajatele ei pakuta.

2.5.3 CORR

Pakett CORR [20] erineb eelpool toodud näidetest selle poolest, et see on loodud kasutamiseks programmeerimiskeskkonnas MATLAB. Programm pakub lihtsat viisi korrelatsioonianalüüsi tegemiseks ja võimaldab tulemusi automaatselt salvestada *.xlsx formaati ning visualiseerida lisaks ka niinimetatud korrelatsioonidiagramme (Joonis 12). Algselt küll biomeditsiini signaalitöötluseks loodud pakett, kuid nagu artiklist [16] nähtub on MATLAB piisava ja oskusliku programmeerimiskogemusega analüütikule heaks töövahendiks ning ilmselt oleks ka seda paketti võimalik aktsiaanalüüsi valdkonnas mingil viisil rakendada.

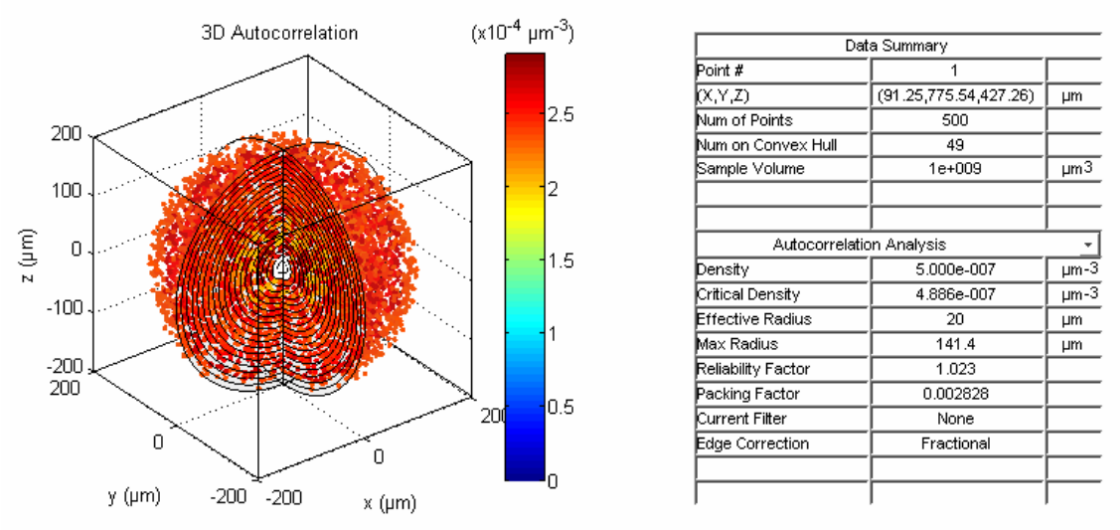


Joonis 12. Paketi CORR korrelatsioonidiagrammid.

2.5.4 Spatial Analysis 3D

Spatial Analysis 3D on kasutajasõbralik graafiline kasutajaliides (ingl *Graphical User Interface*), mis võimaldab statistilisi ja visuaalseid manipuleerimisi reaalsete ja simuleeritud kolmemõõtmeliste ruumipunktide mustritega. Programm on arendatud vajadusest uurida neuronite positsioneerimist kesknärvisüsteemis, kuid selle rakendamine võib programmi loojate sõnul ka näiteks inseneriteaduse, statistika või matemaatika valdkonnas kasulikuks osutuda [21].

Tarkvara on koostatud programmeerimiskeskkonnas MATLAB ning selle kasutusjuhendist [21] nähtub, et selle abil on võimalik uurida ja visualiseerida (Joonis 13) autokorrelatsiooni. Autokorrelatsiooni puhul on tegemist sama protsessi eri ajahetkedele vastavate korrelatsioonidega. Teisisõnu, autokorrelatsiooni puhul sõltub aegrida varasemate perioodide väärtustest [22].



Data Summary		
Point #	1	
(X,Y,Z)	(91.25,775.54,427.26)	µm
Num of Points	500	
Num on Convex Hull	49	
Sample Volume	1e+009	µm3
Autocorrelation Analysis		
Density	5.000e-007	µm-3
Critical Density	4.886e-007	µm-3
Effective Radius	20	µm
Max Radius	141.4	µm
Reliability Factor	1.023	
Packing Factor	0.002828	
Current Filter	None	
Edge Correction	Fractional	

Joonis 13. Spatial Analysis 3D autokorrelatsiooni vaade.

Käesolevas peatükis tutvustatud lahendustest võimaldavad korrelatsiooniseoste muutumist ajas uurida selleks spetsiaalselt ehitatud või kohandatud programmid c|swarm ja MATLAB. Teised välja toodud rakendused oma algsel kujul ei ole selleks sobilikud ja samuti ei võimalda need väärtpaberiportfelli terviklikku korrelatsioonianalüüsi, kus seoste hulk on kordades suurem. Käesolev töö pakub lisaks veel ühe võimaliku viisi selle probleemi lahendamiseks.

Ka finantsandmete visuaalse analüüsi meetodite uuring [23] kinnitab, et praegusel suurandmete ajastul on analüütikutel ja investoritel aina keerulisem teha optimaalse investeerimise ja riskijuhtimise otsuseid, kuna olemasolevad tehnikad ja vahendid on ajale jalgu jäämas. Nimetatud uuringust selgub lisaks, et sellist meetodit, nagu käesolevas töös välja pakutakse, varasemalt kellegi poolt esitletud ei ole. Samuti ei suutnud käesoleva töö autor tuvastada ühtegi sellekohast teadustööd, mis oleks avalikustatud pärast viidatud uuringu avaldamist.

3 Eksperimendis kasutatavad andmed

Käesolevas peatükis on kirjeldatud, milliseid andmeid eksperimendis kasutati ning milliseid kitsendusi seati. Teostatud andmetöötluse peamised etapid on alusandmete korrastamine, korrelatsioonimaatriksite elementide leidmine ning oluliste korrelatsioonide eraldamine ebaolulistest. Ebaolulisust vaadeldakse nii korrelatsiooni tugevust silmas pidades kui ka arvestades tugevate korrelatsiooniseoste esinemise sagedust. Andmete töötlemiseks kasutatakse statistikatarkvara R jaoks integreeritud arenduskeskkonda RStudio [24]. Programmi lähtekoodist (Lisa 2) parema ülevaate saamiseks on käesolevas ja järgmises peatükis R-i realisatsioonid välja toodud joonistena.

3.1 Alusandmed

Eksperimendi lähteandmeteks on S&P 500 indeksisse kuuluvate aktsiate sulgemishinnad iga börsipäeva kohta perioodil 08.02.2013−07.02.2018 [25]. S&P 500 on börsiindeks, mille moodustavad New Yorgi börsil ja NASDAQ-il noteeritud Ameerika Ühendriikide 500 suurima börsiettevõtte aktsiad. S&P on lühend indeksi looja ja haldaja, krediidi-reitinguid ja muid finantsteenuseid pakkuva ettevõtte Standard & Poor's nimest [26].

S&P 500 indeks katab ligikaudu 80% olemasolevast turukapitalisatsioonist ning selle mitmekülgne ülesehitus erinevate majandussektorite lõikes ja kaalumismetoodika eristab seda indeksitest nagu Dow Jonesi tööstuskeskmine või NASDAQ. Paljud peavad seda USA aktsiaturu parimaks kajastuseks ja USA majanduse suunaandjaks [27]. Just laiapõhjaline aktsiate valik ja nende kauplemine suurtel aktsiabörsidel oli üheks määravaks teguriks eksperimendi lähteandmete valikul.

Vaadeldava viie aasta jooksul on toimunud indeksi koosseisus mõningaid muudatusi, mistõttu sisaldab esialgne andmebaas infot 505 aktsia hinnaliikumise kohta. Küll aga puudub sellistel väärtpaberitel, mis on indeksist mingi hetk välja arvatud ning neil, mis asemele võetud, täielik hinnaajalugu. Sellised indeksi komponendid on autori poolse kitsendusena algandmete hulgast välja arvatud, kuna andmete puudumine on erijuht, mis ei kuulu käesoleva töö skoopi.

Algandmete loetakse R-i sisse `read.csv` käsuga. Seejärel moodustatakse andmetest esmalt kahemõõtmeline maatriks, mille veergude päises on aktsiate nimetused ning ridadel vastavate aktsiate sulgemishinnad iga kauplemispäeva kohta vaadeldavas ajavahemikus (Joonis 14).

```
data = within(data.frame(date = as.Date(data$date), close =
as.numeric(data$close), name = as.character(data$Name)), {x =
as.numeric(factor(name))})

reordered_data = subset(data[order(data$x, decreasing = F), ],
select=-c(x))

reshaped_data = reshape(reordered_data, timevar = "name", idvar =
"date", direction = "wide")
```

Joonis 14. Maatriksi moodustamine.

Järgmisena eemaldatakse mittetäieliku ajalooga aktsiate info (Joonis 15), mille tulemusena jäi näidisportfelli 470 aktsiat (Joonis 16).

```
clear_data <- reshaped_data[ , colSums(is.na(reshaped_data)) == 0]
```

Joonis 15. Mittetäieliku ajalooga aktsiate eemaldamine.

Filter | Cols: « < 1 - 50 > »

	date	close.A	close.AAL	close.AAP	close.AAPL	close.ABBV	close.ABC	close.ABT	close.ACN	close.ADBE	close.ADI	close.ADM
71612	2013-02-08	45.08	14.75	78.90	67.8542	36.25	46.89	34.41	73.31	39.120	45.700	30.22
71613	2013-02-11	44.60	14.46	78.39	68.5614	35.85	46.76	34.26	73.07	38.640	46.080	30.28
71614	2013-02-12	44.62	14.27	78.60	66.8428	35.42	46.96	34.30	73.37	38.890	46.270	30.81
71615	2013-02-13	44.75	14.66	78.97	66.7156	35.27	46.64	34.46	73.56	38.810	46.260	31.16
71616	2013-02-14	44.58	13.99	78.84	66.6556	36.57	46.77	34.70	73.13	38.610	46.540	31.40
71617	2013-02-15	42.25	14.50	79.00	65.7371	37.58	46.60	35.08	74.16	38.635	46.175	32.57
71618	2013-02-19	43.01	14.26	80.72	65.7128	38.19	47.22	34.82	75.40	38.995	47.010	33.08
71619	2013-02-20	42.24	13.33	79.50	64.1214	38.61	46.61	34.52	75.00	38.770	45.790	32.50
71620	2013-02-21	41.63	13.37	79.06	63.7228	38.78	46.48	34.26	73.85	38.340	45.120	32.11
71621	2013-02-22	41.80	13.57	79.21	64.4014	38.46	46.95	34.55	74.80	38.550	45.520	32.10
71622	2013-02-25	41.29	13.02	78.36	63.2571	37.37	46.18	34.27	73.91	38.110	44.770	31.70
71623	2013-02-26	40.97	13.26	77.15	64.1385	37.09	46.57	34.06	73.83	38.590	45.000	31.83
71624	2013-02-27	41.73	13.41	77.30	63.5099	36.73	47.03	34.26	74.41	39.600	45.380	32.00
71625	2013-02-28	41.48	13.43	76.34	63.0571	36.92	47.20	33.79	74.36	39.310	45.220	31.86
71626	2013-03-01	41.93	13.61	76.37	61.4957	37.81	47.98	33.60	74.82	39.830	45.230	31.97
71627	2013-03-04	42.03	13.90	77.02	60.0071	38.24	48.25	34.31	75.24	40.460	45.300	32.02

Showing 1 to 19 of 1,259 entries, 471 total columns

Joonis 16. Alusandmete maatriks.

Autori hinnangul on tegemist piisavalt mahuka andmehulgaga, mis sobib hästi ilmestama suure andmehulga komplekssust andmete töötlemisel ja tulemuse visualiseerimisel. Andmete hulk oli teiseks kriteeriumiks alusandmete valikul.

3.2 Korrelatsioonimaatriksite genereerimine

Korrelatsioonimaatriksite moodustamiseks leitakse esmalt igale aktsiale päevased tootlused, kasutades valemit:

$$tootlus = \frac{P_1 - P_0}{P_0},$$

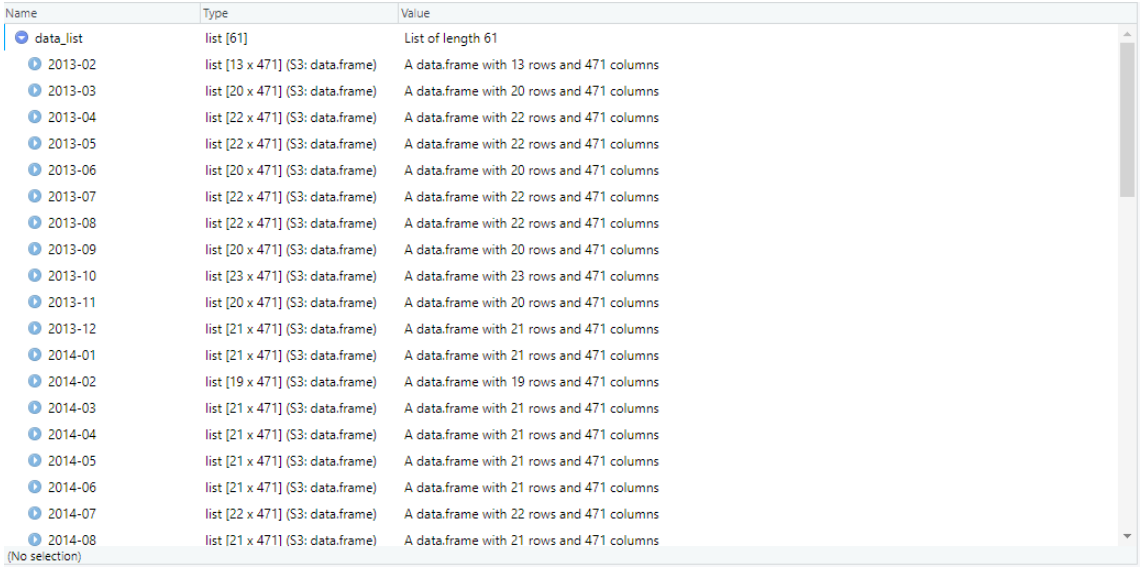
kus P_1 on aktsia hind päeval, mille kohta tootlust arvutatakse ning P_0 aktsia hind sellele eelnenud tööpäeval. Seejärel jagatakse andmed kuude kaupa alamhulkadeks, mille tulemusena moodustatakse algandmete maatriksist list (Joonis 17).

```
daily_returns <- data.frame(clear_data[-1,1], apply(clear_data[,-1],
2, function(x) diff(x)/head(x,-1)))

data_list = split(daily_returns, format(daily_returns$date, "%Y-%m"))
```

Joonis 17. Aktsiate päevaste tootluste arvutamine ja alamhulkadeks jagamine.

Vaadeldav periood 08.02.2013−07.02.2018 koosneb 61 kuu päevastest tootlustest (Joonis 18).



Name	Type	Value
data_list	list [61]	List of length 61
2013-02	list [13 x 471] (S3: data.frame)	A data.frame with 13 rows and 471 columns
2013-03	list [20 x 471] (S3: data.frame)	A data.frame with 20 rows and 471 columns
2013-04	list [22 x 471] (S3: data.frame)	A data.frame with 22 rows and 471 columns
2013-05	list [22 x 471] (S3: data.frame)	A data.frame with 22 rows and 471 columns
2013-06	list [20 x 471] (S3: data.frame)	A data.frame with 20 rows and 471 columns
2013-07	list [22 x 471] (S3: data.frame)	A data.frame with 22 rows and 471 columns
2013-08	list [22 x 471] (S3: data.frame)	A data.frame with 22 rows and 471 columns
2013-09	list [20 x 471] (S3: data.frame)	A data.frame with 20 rows and 471 columns
2013-10	list [23 x 471] (S3: data.frame)	A data.frame with 23 rows and 471 columns
2013-11	list [20 x 471] (S3: data.frame)	A data.frame with 20 rows and 471 columns
2013-12	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns
2014-01	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns
2014-02	list [19 x 471] (S3: data.frame)	A data.frame with 19 rows and 471 columns
2014-03	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns
2014-04	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns
2014-05	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns
2014-06	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns
2014-07	list [22 x 471] (S3: data.frame)	A data.frame with 22 rows and 471 columns
2014-08	list [21 x 471] (S3: data.frame)	A data.frame with 21 rows and 471 columns

(No selection)

Joonis 18. Päevasete tootluste list.

Juhul, kui esimese ja/või viimase kuu andmeid on kuude keskmisest andmemahust vähem, siis arvatakse see/need edaspidisest vaatlusest välja, millega luuakse kuude lõikes võrdsed eeldused korrelatsioonikordajate leidmiseks (Joonis 19).

```
if (nrow(data_list[[1]]) < ave(sapply(data_list, NROW))) {
  data_list[[1]] <- NULL
}

if (nrow(data_list[[length(data_list)]]) < ave(sapply(data_list,
NROW))) {
  data_list[[length(data_list)]] <- NULL
}
```

Joonis 19. Esimese ja viimase kuu andmete eemaldamine.

Ka antud töös tekkis olukord, kus esimesel ja viimasel kuul (veebruar 2013 ja veebruar 2018) ei tekkinud andmetöötluse käigus teiste kuudega võrreldav hulk tootlusi ning need eemaldati edasisest analüüsist. Seega hõlmab käesoleva töö praktiline osa ajavahemiku märts 2013 kuni jaanuar 2018 k.a. ehk kokku 59 kuu tootluseid.

Järgmiseks eemaldatakse iga kuu esimene rida, sest esimese kuu puhul sisaldab see vigast arvutustulemust *N/A* ning kõikidel ülejäänud kuudel kuu esimese päeva tootlust eelmise kuu viimase päeva suhtes, mis ei ole soovitud. Lisaks eemaldatakse ka kuupäevade veerud, kuna edaspidises analüüsis neid enam tarvis ei ole (Joonis 20).

```
new_data_list = lapply(data_list, function(x) x[-1,-1])
```

Joonis 20. Maatriksite esimese rea ja kuupäevade veergude eemaldamine.

Päevaste tootluste põhjal leitakse iga aktsiapaari kohta igal kuul konkreetse kuu baasil Pearsoni korrelatsioonikordaja (r), mis mõõdab kahe objekti vahel lineaarset seost. Seost nimetatakse lineaarseks, kui ühe tunnuse muutus on seotud teise tunnuse proportsionaalse muutusega [28].

Pearsoni kordaja on kahe pideva muutuja vahelise seose tugevuse mõõt, mille väärtused jäävad -1 ja +1 vahele. Väärtus +1 on muutujate vahelise ideaalse positiivse seose tulemus. Positiivsed korrelatsioonid näitavad, et mõlemad muutujad liiguvad samas suunas. Seevastu väärtus -1 tähistab ideaalset negatiivset seost. Negatiivsed seosed näitavad, et ühe muutuja suurenedes teine väheneb ehk et need on pöördvõrdeliselt seotud. Väärtus 0 viitab muutujate vahelise korrelatsiooni puudumisele [29].

Seose tugevuse hindamisel on aluseks võetud Rowntree [30] skaala, mille järgi väljendab $0 \leq |r| \leq 0{,}2$ olematut või väga nõrka, $0{,}2 < |r| \leq 0{,}4$ nõrka, $0{,}4 < |r| \leq 0{,}7$ keskmist või mõõdukat, $0{,}7 < |r| \leq 0{,}9$ tugevat ning $0{,}9 < |r| \leq 1$ väga tugevat seost.

Joonisel 22 toodud tulemuse saamiseks leitakse korrelatsioonikordaja tervele listile korraga (Joonis 21).

```
cor_list = lapply(new_data_list, cor, method = "pearson")
```

Joonis 21. Korrelatsioonikordajate leidmine.

Name	Type	Value
cor_list	list [59]	List of length 59
2013-03	double [470 x 470]	1.000000 0.364849 -0.050739 -0.201890 0.239684 0.027383 0.364849 1.000000 ...
2013-04	double [470 x 470]	1.000000 0.400530 0.474722 0.275822 0.162238 0.219920 0.400530 1.000000 ...
2013-05	double [470 x 470]	1.000000 0.274073 0.732473 -0.476792 0.357904 0.382790 0.274073 1.000000 ...
2013-06	double [470 x 470]	1.00000 0.39206 0.25546 0.50513 0.66841 0.34134 0.39206 1.00000 0.57322 ...
2013-07	double [470 x 470]	1.00e+00 -5.50e-02 1.81e-01 -1.18e-01 4.79e-02 1.68e-01 -5.50e-02 1.00e+00 ...
2013-08	double [470 x 470]	1.000000 -0.047056 0.394383 0.150344 0.089273 -0.070207 -0.047056 1.000000 ...
2013-09	double [470 x 470]	1.000000 0.378976 0.347874 -0.011176 0.157454 0.402332 0.378976 1.000000 ...
2013-10	double [470 x 470]	1.000000 0.220222 0.449139 -0.052177 0.670614 0.889887 0.220222 1.000000 ...
2013-11	double [470 x 470]	1.00e+00 1.79e-01 3.85e-01 2.49e-02 3.66e-01 2.12e-01 1.79e-01 1.00e+00 ...
2013-12	double [470 x 470]	1.00e+00 -1.12e-01 -6.11e-02 -1.33e-01 3.58e-01 5.05e-01 -1.12e-01 1.00e+00 ...
2014-01	double [470 x 470]	1.000000 0.268371 0.530062 0.137643 0.222857 0.686690 0.268371 1.000000 ...
2014-02	double [470 x 470]	1.00e+00 5.40e-01 1.38e-01 2.34e-02 1.40e-01 1.92e-01 5.40e-01 1.00e+00 ...
2014-03	double [470 x 470]	1.00e+00 5.52e-01 4.02e-01 7.06e-02 6.29e-01 5.10e-01 5.52e-01 1.00e+00 ...
2014-04	double [470 x 470]	1.000000 0.739039 0.861978 0.254189 0.612632 0.613923 0.739039 1.000000 ...
2014-05	double [470 x 470]	1.000000 0.719839 -0.114721 0.402106 0.020049 0.294531 0.719839 1.000000 ...
2014-06	double [470 x 470]	1.000000 0.359627 0.238642 0.204988 0.195629 -0.114970 0.359627 1.000000 ...
2014-07	double [470 x 470]	1.000000 0.373485 0.092157 0.414119 0.363729 0.191164 0.373485 1.000000 ...
2014-08	double [470 x 470]	1.00e+00 4.39e-01 2.64e-01 2.92e-01 1.84e-01 2.49e-01 4.39e-01 1.00e+00 ...

(No selection)

Joonis 22. Pearsoni korrelatsioonikordajad.

Tugev seos aktsiapaari vahel tähendab, et suure tõenäosusega liiguvad mõlema aktsia hinnad samal ajal samas suunas ja tugeva negatiivse korrelatsiooni korral vastassuunas. Esimesel juhul toob kaasa sellise aktsiapaari omamine portfellis kasumi/kahjumi võimendumise ning teisel juhul riski maandamise.

Autori poolse eelduse kohaselt toimub üldine liikumine aktsiaturgudel valdavalt samas suunas ning seetõttu võiks olla ka näidisportfelli aktsiate hinnaliikumine enamjaolt samasuunaline. Selle eelduse kontrollimiseks vaadeldakse edasises analüüsis positiivseid ja negatiivseid korrelatsiooniseosed eraldi. Negatiivsete seoste hulgas leiduvad tõepoolest vaid mõned üksikud väga tugevad seosed, mille uurimisel käesoleva töö skoobi raames poleks mõtet. Seetõttu vaadeldakse negatiivsete seoste hulgas lisaks väga tugevale ka tugevat seost ehk $-1 < r \leq -0{,}7$. Seevastu on positiivsete korrelatsioonide seas piisaval hulgal väga tugeva seosega aktsiate paare, mistõttu on võetud sellele hulga alampiiriks 0,9.

Eelnevat arvesse võttes asendatakse korrelatsioonimaatriksites vastavalt kas tugevad või väga tugevad seosed väärtusega 1 ja kõik ülejäänud seosed väärtusega 0. Kuna

korrelatsioon iseendaga (korrelatsioonikordaja väärtuseks on 1) on antud analüüsis ebavajalik info, siis ka need asendatakse väärtusega 0 (Joonis 23).

```
for (k in 1:length(cor_list)) {
  cor_matrix_list[[k]] = as.matrix(cor_list[[k]])
  diag(cor_matrix_list[[k]]) <- 0
  cor_matrix_list[[k]][cor_matrix_list[[k]] > 0.9] <- 1
  cor_matrix_list[[k]][cor_matrix_list[[k]] <= 0.9] <- 0
}
```

Joonis 23. Seoste asendamine väärtustega 1 ja 0.

Lisaks korrelatsiooni tugevuse hindamisele vaadeldakse seosete esinemise sagedust ka läbi aja. Üksikuid tugevaid korrelatsioone aktsiate paari vahel saab vaadelda kui juhuslikke suurusi ehk müra. Kuna loodus- ja inseneriteadustes peetakse sobilikuks olulisuse nivoo suuruseks $\beta = 0{,}05$ [31], siis sellest lähtuvalt on alla selle nivoo jäävad seosed edasisest analüüsist välja jäetud (Joonis 24).

```
for (i in 1:nrow(cor_matrix_list[[k]])) {
  for (j in 1:ncol(cor_matrix_list[[k]])) {
    sum = 0
    for (k in 1:length(cor_matrix_list)) {
      sum = sum + cor_matrix_list[[k]][[i,j]]
    }
    if (sum < length(cor_matrix_list)*0.05) {
      for (k in 1:length(cor_matrix_list)) {
        cor_matrix_list[[k]][[i,j]] <- 0
      }
    }
  }
}
```

Joonis 24. Müra eemaldamine.

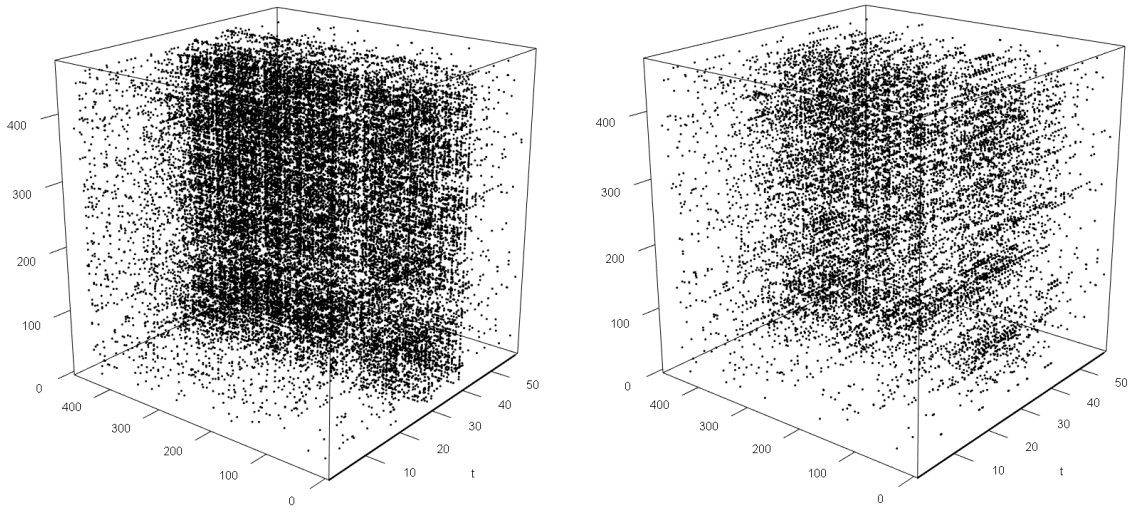
Müra eemaldamise järgselt selgus, et ühtegi pidevat ja negatiivset tugevat seost läbi aja näidisportfelli aktsiate vahel ikkagi ei eksisteerinud, ning seetõttu keskendutakse edaspidi vaid positiivsete väga tugevate korrelatsiooniseoste uurimisele.

Andmete korrastamise lõpptulemusena saadakse 59 binaarset korrelatsioonimaatriksit, mille paigutamisel ajalises järgnevuses üksteisega kohakuti (Joonis 3) saame kolmemõõtmelise aegruumi kuubi, mille telgedel *x* ja *y* on portfelli kuuluvad aktsiad ning *z*-teljel aeg. Kuubi elementideks on aktsiapaaride omavahelised korrelatsiooniseosed erinevatel ajahetkedel.

Joonisel 25 on kujutatud algandmete töötluse tulemusena saadud aegruumi kuup 470 aktsia omavahelistest paarikaupa positiivsetest korrelatsiooniseostest läbi aja koos müra

(vasakul) ja ilma mürata (paremal), mida järgnevalt järjestama hakatakse. Aktsiate suure arvu tõttu ei kuvata joonisel aktsiate nimesid, vaid neid kujutatakse skaalal 0–470 ning ajatelg *t* moodustub 59-st järjestikusest kuust.

Visualiseerimise rakendustes saab genereeritud kuupi erinevatesse asenditesse pöörata. Käesolevas töös on parema vaadeldavuse huvides asetatud aegruumi kuup nii, et ajahetke kihid (*t*) paiknevad *yz*-tasapinnas selliselt, et ajavahemiku kõige varasem kuu asub esiplaanil.



Joonis 25. Positiivsed korrelatsiooniseosed läbi aja koos müraga (vasakul) ja ilma mürata (paremal).

Alusandmete eeltöötluse tulemusena on 505 aktsia 5 aasta päevastest sulgemishindadest saadud aegruumi kuup 470 aktsia omavahelistest väga tugevatest positiivsetest korrelatsiooniseostest 59-l järjestikusel ajahetkel. Järgneva etapina rakendatakse saadud kuubile ühte võimalikku järjestamise algoritmi ning visualiseeritakse selle tulem.

4 Aegruumi kuubi järjestamise eksperiment

Käesoleva töö raames läbi viidud eksperimendi üheks eesmärgiks on visualiseerida aktsiate omavaheliste korrelatsioonide muutumist ajas. See aga tähendab, et kolmemõõtmelises aegruumi kuubis säilitatakse korrelatsioonimaatriksite kihtide ajaline järgnevus ja ümber järjestatakse vaid aktsiatega seotud read ja veerud. Antud peatükis tutvustatakse järjestamiseks kasutatud algoritme ja püstitatakse hüpotees nende parimaks rakendamiseks ning kirjeldatakse järjestamise eksperimendi läbiviimist ja tulemuse visualiseerimist.

4.1 Kasutatud järjestusalgoritmid

Aegruumi kuubi järjestamiseks ei kasutata käesolevas töös klasterdamise algoritme, mis ei võimalda säilitada kuubi kihtide ajalist järgnevust. Klasterdamise asemel leitakse kuubi järjestus järgnevat meetodit rakendades. Kuubi ühele ajahetke kihile (edaspidi baaskiht) rakendatakse järjestamise algoritmi ja ülejäänud kihid järjestatakse ümber vastavalt baaskihi (kahemõõtmeline maatriks) ümberjärjestatud ridade ja veergude järjekorrale. Baaskihi järjestamiseks kasutatakse käesolevas töös tarkvaras R sisalduva *seriation* paketi järjestusalgoritme *Bond Energy Algorithm* (BEA) ning *Principal Component Analysis* (PCA). Nimetatud algoritme on kasutatud kuna need ja nende modifikatsioonid on sobilikud just maatrikskujul olevate andmete järjestamiseks [32].

BEA algoritmi eesmärk on järjestusprotsessi käigus korrastada andmeid selliselt, et igal andmepunktil oleks võimalikult palju nullist erinevaid naaberelemente. Kahemõõtmelises maatriksis mõeldakse nende naaberelementide all elemente, mis paiknevad vahetult kõrvalreas või kõrvalveerus. Sellist maksimaalsete naaberelementide olemasolu mõõdetakse vastava efektiivsusmõõdikuga (ingl *Measure of Effectiveness*, edaspidi ME), millest tuleb täpsemalt juttu valideerimise alapeatükis. Kuna ridade ümberreastamine ei mõjuta vastavates veergudes olevate andmete panust ME mõõdiku väärtusele ja vastupidi, siis protsessi käigus järjestatakse ridu ja veerge vaheldumisi, kuni maatriksi ME mõõdik annab maksimaalse tulemuse [32].

BEA algoritmi rakendamise tulemus sõltub, millisest reast või veerust järjestamist alustatakse. Kuna see valitakse juhuslikult, siis parema tulemuse saamiseks tuleks algoritmi rakendada mitmekordselt [32].

PCA algoritm lähtub eesmärgist minimeerida nullist erinevate elementide vahelisi eukleidilisi kaugusi, alustades esimesest elemendist. Ridade ja veergude ümberreastamisi jätkatakse seni, kuni vastav stressimõõdik on saavutanud võimaliku miinimumi. Algoritmi rakendamise tõhusust näitav stressimõõdik arvutatakse iga maatriksi elemendi kohta antud elemendi ning kõikides suundades esimese nullist erineva elemendi ruutude kauguse summana [32].

Kuna kogu aegruumi kuup järjestatakse ümber vastavale sellele, kuidas järjestuvad baaskihi read ja veerud, siis on baaskihi valik olulise tähtsusega. Parima aegruumi kuubi järjestamise tulemuse saamiseks on vaja teha kindlaks, milline kiht on baaskihi valikuks parim. Eksperimendi käigus teostati järjestamised kõikide baaskihtide põhjal kasutades eelpool nimetatud järjestusalgoritme (BEA ja PCA) ning valideerides tulemusi mitme erineva mõõdiku vastu.

Kuna igakordselt kõikide kihtide järgi järjestamine on ressursimahukas, siis püstitab käesoleva töö autor parima baaskihi määramiseks hüpoteesi, mida testitakse eksperimendi käigus ning mille tulemust hinnatakse vastu valideeritud järjestustulemusi.

Parima baaskihi määramise hüpotees. Binaarsete elementidega aegruumi kuubi järjestamise aluseks on parimaks baaskihiks kiht, mis on teiste kihtidega maksimaalselt sarnane. Kihte võib pidada sarnaseks, kui nad omavad võimalikult sarnast nullist erinevate väärtustega elementide jaotust.

Olgu aegruumi kuup $\mathcal{X} = \{x_{ijk}\} \in \mathbb{R}^{n \times \mathrm{n} \times \mathrm{m}}$. *l*-nda kihi $(1 \leq l \leq m)$ sarnasustegur on antud kihi iga elemendi sarnasustegurite summa. Elemendi sarnasustegur näitab omakorda kihtide arvu, kus lisaks *l*-ndale kihile leidub veel samas reas ja veerus element 1. Matemaatiliselt saab binaarsete andmete korral kontrollida nullist erineva elemendi olemasolu korrutamise teel. Seega saab kihi sarnasusteguri avaldada järgmiselt:

$$s_l = \sum_{i,j}^{n} s_{ij} = \sum_{i,j}^{n} \sum_{\substack{k \\ k \neq l}}^{m} x_{ijl} \cdot x_{ijk.}$$

Hüpoteesi kohaselt on parima baaskihi sarnasustegur sarnasustegurite seast maksimaalne, st $max\{s_l\}$, kus $1 \leq l \leq m$, kusjuures selliseid kihte võib olla rohkem kui üks. Kui järjestusprotsessis baaskihi valimisel ilmneb, et maksimaalse sarnasusteguriga kihte on rohkem kui üks, siis tuleb tegevuskava valikul lähtuda konkreetse eksperimendi nõuetest. Kui täpsus on olulisem kui ressurss, siis testida läbi kõik maksimaalse sarnasusteguriga kihid, vastasel juhul võib valida baaskihi maksimaalsete hulgast vabalt.

4.2 Järjestamine ja tulemuste visualiseerimine

Kuubi järjestamisel läbitakse realisatsioonis järgnevad etapid:

1. valitakse baaskiht,
2. baaskihile rakendatakse järjestamise algoritmi,
3. tuvastatakse baaskihi uus ridade ja veergude järjekord,
4. kõik ülejäänud kuubi kihid järjestatakse ümber vastavalt baaskihi uuele ridade ja veergude järjekorrale,
5. iga ümberjärjestatud pildi kohta salvestatakse töökataloogi PNG formaadis pilt,
6. järjestuse andmetest tekitatakse R-s eraldi andmestik (vajalik visualiseerimiseks),
7. järjestuse kõikide kihtide piltidest moodustatakse pakkimise tulemusena MP4 formaadis videofail,
8. pildid kustutatakse,
9. arvutatakse järjestuse ME.

Kõikide kihtide järgi järjestuse leidmiseks korratakse etappe 1–9 seni, kuni kuubi iga ajahetke tähistav kiht on olnud korra baaskihiks. Iteratsioonide lõpuks on töökataloogi salvestatud 59 videofaili ning iga järjestuse kohta on arvutatud ME, mis on vajalikud valideerimise faasis. Videofailide loomiseks kasutatavad pildid kustutatakse, kuna edasises analüüsis neid enam tarvis ei ole.

Iteratsiooni etappide koodiread, milles on näitena rakendatud PCA algoritmi, on Lisas 2 toodud lähtekoodis tähistatud vastavate etapi numbritega.

Kõikide kihtide järgi järjestamise alternatiivina leitakse ka hüpoteetiliselt parim baaskiht alapeatükis 4.1 püstitatud hüpoteesi arvestades. Kuna käesolevas eksperimendis on tegemist binaarsete maatriksitega, siis valemi rakendamiseks piisab, kui vaadeldava kihi iga nullist erineva elemendi korral summeeritakse ülejäänud kihtides samal positsioonil

asuvate elementide väärtused ning liidetakse saadud summad omavahel, saades nii kihi sarnasusteguri. Hüpoteesi kohaselt on parimaks baaskihiks kiht, mille sarnasustegur on kihtide sarnasustegurite seas maksimaalne. Joonisel 26 on toodud programmikood, mida rakendatud suurima sarnasusteguriga kihi leidmiseks. Antud programmikoodi näide ei arvesta hetkel võimalusega, et aegruumi kuubil võiks olla mitu maksimaalse väärtusega sarnasusteguriga kihti, sest antud eksperimendis oli see juba varasemalt testitud, et selliseid kihte on üks. Eksperimenteeritava aegruumi kuubi puhul osutus hüpoteetiliselt parimaks kihiks kiht nr 30.

```
#järjestamiseks parima kuu leidmine
best_sum = 0
best_month = 0
for (k in 1:length(cor_matrix_list)) {
  sum_all = 0
  for (i in 1:nrow(cor_matrix_list[[k]])) {
    for (j in 1:ncol(cor_matrix_list[[k]])) {
      elem_sum = 0
      if (cor_matrix_list[[k]][i,j] != 0) {
        for (l in 1:length(cor_matrix_list)) {
          elem_sum = elem_sum + cor_matrix_list[[l]][i,j]
        }
        elem_sum = elem_sum - cor_matrix_list[[k]][i,j]
      }
      sum_all = sum_all + cor_matrix_list[[k]][i,j] * elem_sum
    }
  }
  if (sum_all > best_sum) {
    best_sum = sum_all
    best_month = k
  }
}
```

Joonis 26. Hüpoteetiliselt parima kihi leidmine.

Järjestatud kuubi visualiseerimiseks on käesolevas töös kasutatud valdavalt R-i. Küll aga on välja toodud ka mõned näited rakendusega Cubix [33] visualiseeritud andmehulkadest. R-i kasuks langes valik kahel põhjusel:

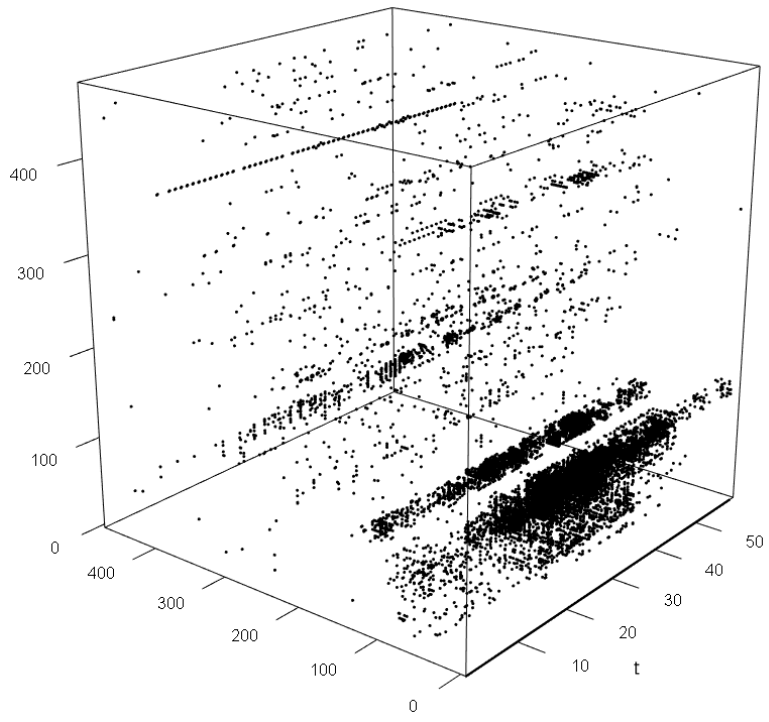
1. Visualiseerimiseks on mugav kasutada andmete töötlemisega sama keskkonda, sest sel juhul ei pea andmeid ühest keskkonnast teise eksportima.

2. R-s ei tekkinud andmete suurest mahust mingisuguseid tõrkeid. Cubix seevastu on tänaseks tehniliselt aegunud ja sisendfailivormingu suhtes väga tundlik. Samuti

on maksimaalne sisendfailisuurus seotud Java mälujaotusega, mistõttu jäi see üle 100 000 KB suuruse faili importimisega hätta [34].

Mõlemad programmid (Cubix küll oma sisendandmete mahu kitsendustega) võimaldavad kolmemõõtmelist kuubikut arvutiekraanil keerata ja pöörata oma äranägemise ja soovi järgi, mille tulemusena on võimalik vaatlejal paremini märgata seosete ajaloolisi muutuseid, mis muidu välja ei paista. Cubix-i peale võiks väiksemate andmemahtude korral kindlasti mõelda, kuna see võimaldab lihtsa faili sisselugemise vaevaga vaadelda ilusat 3D kujutist ning lisaks saab sellega kergesti vaadelda ka seda, mis toimub kujutise sees. R eeldab selleks aga heal tasemel programmeerimisoskust.

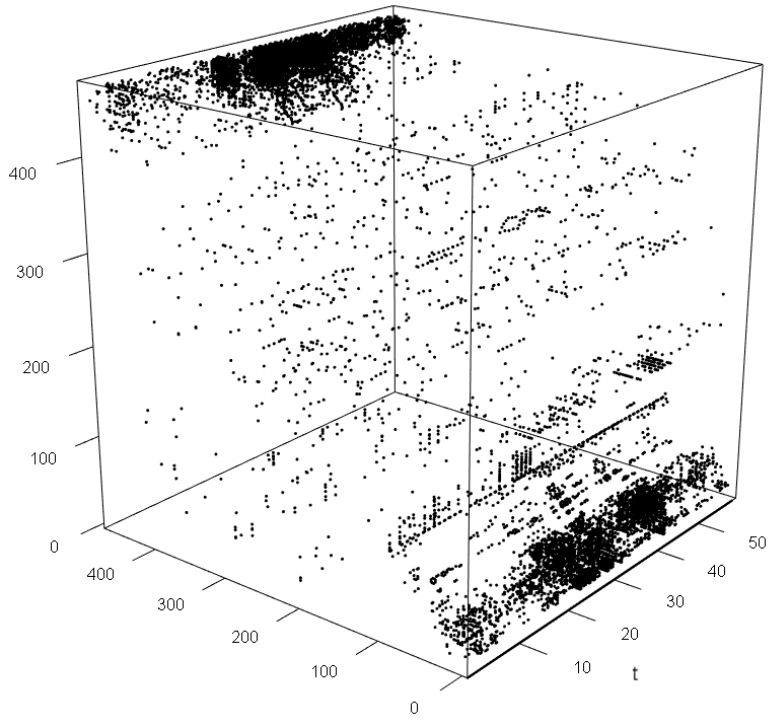
Joonisel 27 on kujutatud hüpoteetiliselt parima kihi järgi järjestatud aegruumi kuupi, mille baaskiht on järjestatud BEA algoritmi abil. Visualiseerimine teostatakse programmi koodis funktsiooniga `plot3d` ning ajatelje sätteid täiendatakse `axes3d` abil.



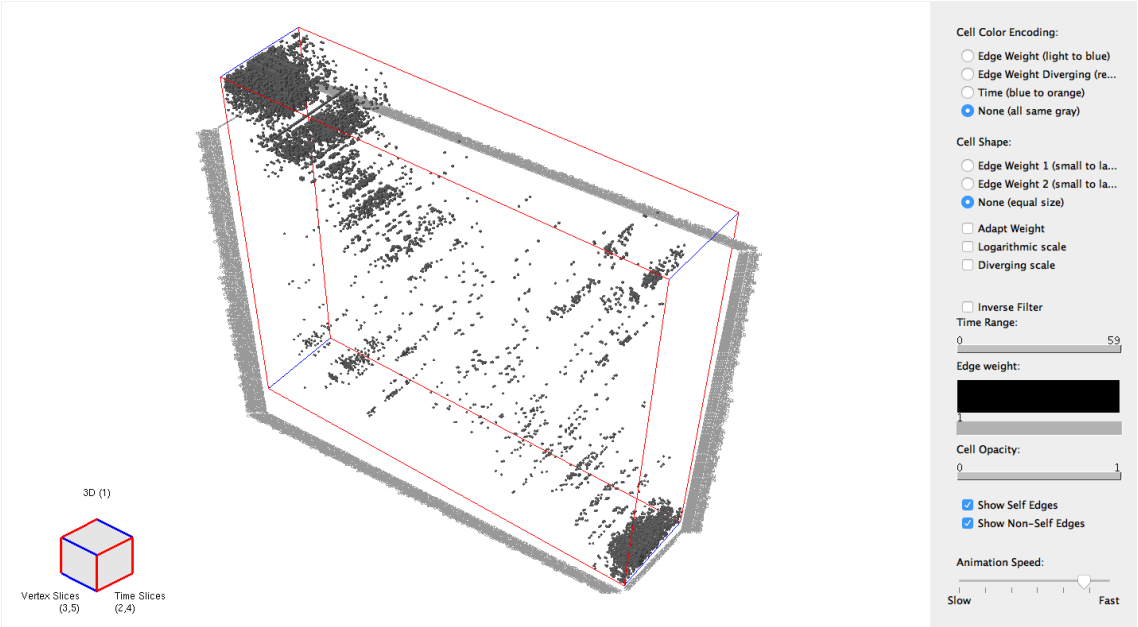
Joonis 27. BEA algoritmiga järjestatud kuubik hüpoteetiliselt parima baaskihi järgi.

Kui toodud joonisel on andmete koondumine visuaalselt näha, siis müraga koos andmete koondumist ei toimunud. Seetõttu on oluline enne järjestamisega alustamist üksikud juhuslikud korrelatsioonid andmete hulgast eemaldada.

Joonistel 28 ja 29 on kujutatud samuti hüpoteetiliselt parima kihi järgi järjestatud kuupi, kuid neil on baaskihile rakendatud PCA algoritmi. Visualiseerimine on joonisel 28 teostatud tarkvaraga R ning joonisel 29 programmiga Cubix.



Joonis 28. PCA algoritmiga järjestatud kuubik hüpoteetiliselt parima baaskihi järgi.



Joonis 29. Cubix kasutajaliidese visuaal.

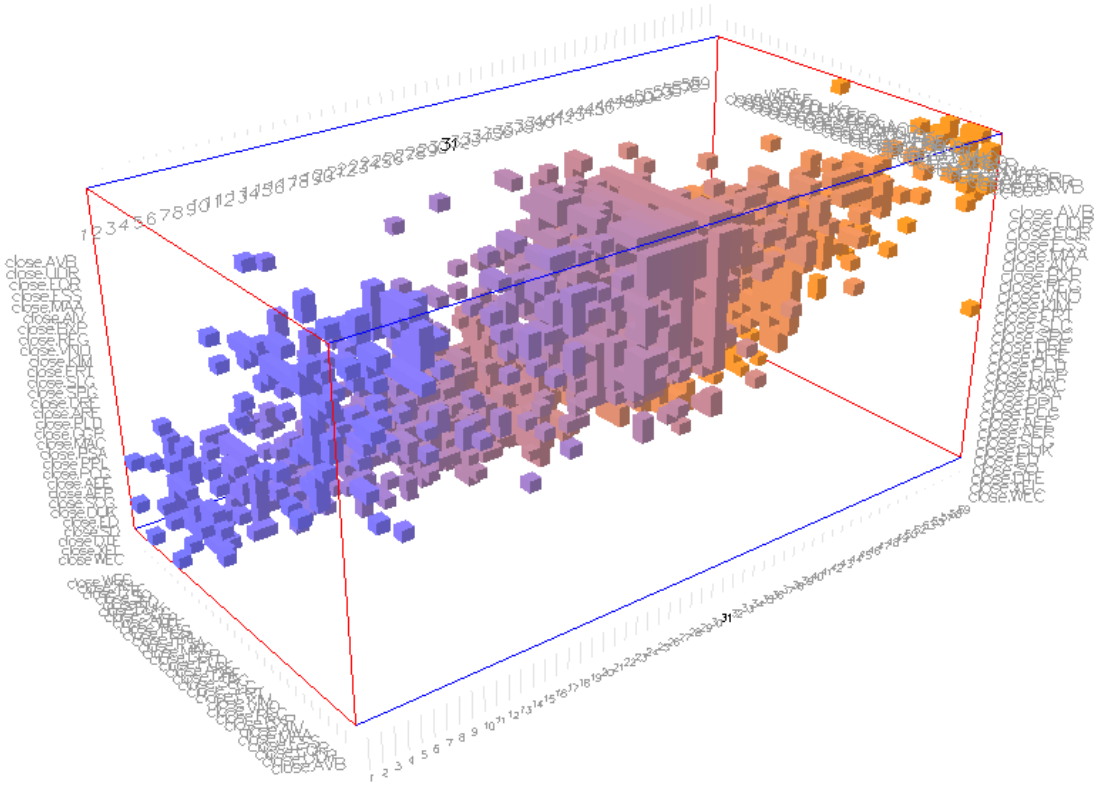
Nii R kui ka Cubix annavad järjestatud aegruumi kuubist üldiselt hea visuaalse ülevaate. Kui aga on soov täpsemalt teada, millised aktsiad on need, mis kuubi erinevatesse ruumiosadesse on koondunud, siis selleks soovitab töö autor Cubix-it. Sest nagu juba eelpool korra mainitud, siis Cubix ei eelda programmeerimise oskust ning sellega on äärmiselt lihtne teostada erinevaid huvipakkuvaid vaatlusi. Küll aga võib selle uurimistöö raames loodud rakenduse kasutamisel esineda tõrkeid, sest uuendusi pole sellele tehtud.

Näiteks saab modifitseerida R-i koodi nii, et suurest andmestikust eraldatakse väiksem andmehulk näiteks 30 x 30 läbilõike jaoks kohast, kuhu korrelatsioonipunktid koondunud on (Joonis 30).

```
if (l == best_month) {
  month_cor <- month_cor[1:30,1:30]
}
```

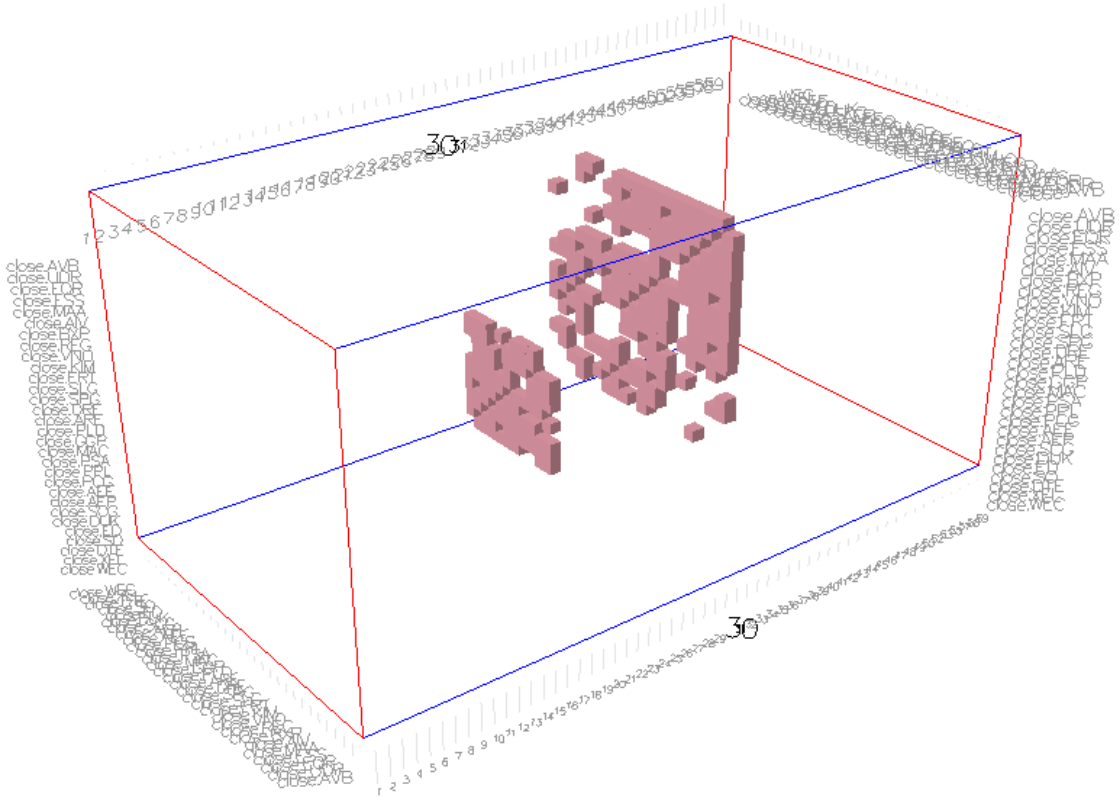
Joonis 30. 30x30 andmestiku eraldamine.

Vähendatud mahus andmestiku Cubix-isse laadimise järgselt kuvatakse joonisel 31 toodud kujutis, kus soovi korral on võimalik ühe nupuvajutusega ajalist muutust kujutada värviskaalal sinisest oranžini.



Joonis 31. Kuubi 30x30 läbilõige Cubix-is.

Soovides aga uurida hüpoteetiliselt parima kihi nr 30. algoritmi poolt tekitatud „mustrit“, siis ka see on kergesti tehtav. Cubix-i kasutajaliideses piisab kursoriga vastava kihi peale liikumisest, kui tulemus juba kuvatakse (Joonis 32).



Joonis 32. Kuubi 30x30 läbilõike kiht nr. 30.

Cubix võimaldab toodud näidetest veelgi detailsemaid vaateid (näiteks erinevad laotused, üksikute huvipakkuvate punktide selekteerimine jne) ja analüüsi (andmeid on võimalik eksportida neljas erinevas formaadis), kuid see võiks autori hinnangul olla käesoleva töö üheks võimalikuks edasiarenduseks näiteks majandusteaduste valdkonnas.

Aegruumi kuubi järjestamise eksperiment osutus visuaalse vaatluse hinnangu järgselt edukaks. Järgnevalt hinnatakse eksperimendi edukust ka valideerimismõõdikute abil.

5 Tulemuste valideerimine ja analüüs

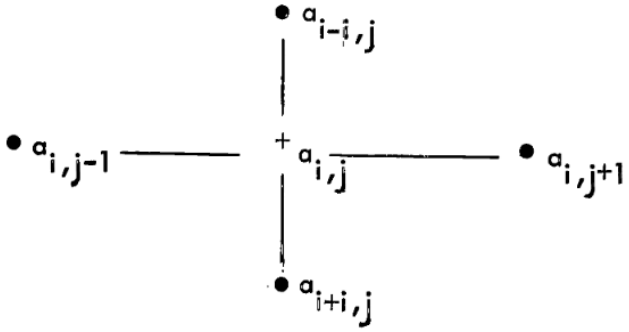
Sõltumata järjestamiseks kasutatud järjestusalgoritmist rakendatakse käesolevas töös tulemuste valideerimiseks kahte erinevat mõõdikut. Mõõdikute väärtuste järgi saadakse baaskihtide parim järjestus ning neid järjestusi analüüsides saab anda hinnangu eksperimendi tulemustele. Eksperimendi valideeritud tulemuste põhjal hinnatakse ka varasemalt autori poolt püstitatud hüpoteesi paikapidavust.

5.1 Valideerimise mõõdikud

Naaberelementide seoste (antud juhul väärtuse) olemasolu saab mõõta efektiivsus-mõõdikuga *Measure of Effectiveness* (ME), mis on algselt defineeritud McCormick, Deutsch, Martin ja Schweitzer [35] poolt kahemõõtmelise maatriksi jaoks alljärgnevalt. Olgu seoste maatriks mõõtmetega $M \times N$, millel on mittenegatiivsed elemendid a_{ij}, ja kus a_{ij} on leitav valemiga:

$$a_{ij} = \frac{1}{2}\left[a_{i+1,j} + a_{i-1,j} + a_{i,j+1} + a_{i,j-1}\right],$$

tingimusel $a_{0,j} = a_{i,0} = a_{M+1,j} = a_{i,N+1} = 0$. Jooniselt 33 nähtub, et a_{ij} on oma horisontaalsete ja vertikaalsete lähimate naabrite a_{ij} poolsumma [35].



Joonis 33. 2D ME näitlikustav kujutis.

Poolsumma seetõttu, kuna iga kahe elemendi vahelist seost hinnatakse mõlema elemendi vaatest. Seega saab ME defineerida järgmiselt [35]:

$$ME = \sum_{i,j} a_{ij}\, \mathrm{a}_{ij}.$$

ME võrdub kõigi maatriksi vertikaalsete ja horisontaalsete seoste tugevuste summaga, kus kahe horisontaalselt või vertikaalselt külgneva elemendi vahelise seose tugevus on määratletud kui elementide korrutis [35].

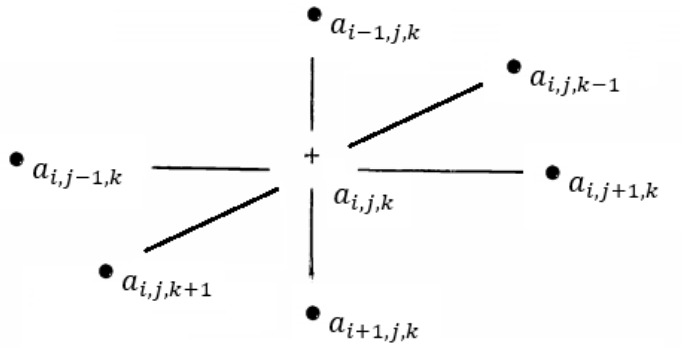
Maatriksi maksimaalse "tükilisuse" saamiseks on vaja maksimeerida ME kõikides maatriksi ridade ja veergude permutatsioonides, st

$$\max_{\substack{\text{kõikide ridade permutatsioonid} \\ \text{ja veergude permutatsioonid}}} \left\{ ME = \sum_{i,j} a_{ij}\, \mathrm{a}_{ij} \right\}$$

[35]. Eelnevale tuginedes konstrueeritakse ME kolmemõõtmelise andmehulga jaoks. Selleks täiendatakse kahemõõtmelist maatriksit veel ühe mõõtmega, saades kuubi. Seega, olgu kuup mõõtmetega $M \times N \times P$, millel on mittenegatiivsed elemendid a_{ij}, kus a_{ij} on defineeritud kui:

$$\mathrm{a}_{ijk} = \frac{1}{2}\left[a_{i+1,j,k} + a_{i-1,j,k} + a_{i,j+1,k} + a_{i,j-1,k} + a_{i,j1,k+1} + a_{i,j,k-1}\right],$$

tingimusel $a_{0,j,k} = a_{i,0,k} = a_{i,j,0} = a_{M+1,j,k} = a_{i,N+1,k} = a_{M,j,P+1} = 0$. Analoogiliselt nähtub autori poolt täiendatud jooniselt 34, et a_{ij} on ka kolmemõõtmelises andmehulgas oma lähimate naabrite a_{ij} poolsumma.



Joonis 34. Autori poolsete täiendustega 3D ME näitlikustav kujutis.

Seega saab kolmemõõtmelise andmehulga ME defineerida järgmiselt:

$$ME = \sum_{i,j,k} a_{ijk}\, \mathrm{a}_{ijk}.$$

Kuubi maksimaalse „tükilisuse“ saamiseks on vaja maksimeerida ME kõikides kuubi ridade, veergude ja torude permutatsioonides. Kuna aga käesolevas töös tehtud kitsenduse tõttu jäetakse ajahetki tähistavad kihid esialgsesse järjestusse, siis mööda kuubi torusid permutatsioonide leidmist antud juhul ei toimu. Sellegipoolest võib indikatsiooni saamiseks mõõta aegruumi kuubi „tükilisust“ järgneva valemiga:

$$\max_{\substack{kõikide\ ridade\ permutatsioonid \\ ja\ veergude\ permutatsioonid}} \left\{ ME = \sum_{i,j,k} a_{ijk}\, \mathrm{a}_{ijk} \right\}.$$

Viimasena toodud valemit on kasutatud kuubi ME arvutamiseks iga järjestuse kohta. BEA algoritmi eesmärk on saada järjestatavas kihis ME mõõdik maksimaalseks ja selliselt järjestatud aegruumi kuubi puhul on kolmemõõtmelise ME väärtuse arvutamine igati asjakohane. PCA algoritmi rakendamisel on eesmärgiks vähendada nullist erinevate punktide omavahelisi eukleidilisi kaugusi. Sellest lähtuvalt võib öelda, et ka PCA algoritm proovib saada punktid võimalikult kõrvuti ning selliselt järjestatud kuubi peal kolmemõõtmelise ME mõõdiku rakendamine annab samuti sisendi, mille põhjal hinnata parima baaskihi valikut.

Alternatiivseks valideerimise mõõdikuks on käesolevas töös kasutatud kuubi järjestamisel moodustatud videofailide salvestusmahtude võrdlemist. Järjestusalgoritmi (nii BEA kui PCA) rakendamise järgselt salvestati töökataloogi esmalt igast kihist PNG formaadis pilt, ning kõikide kihtide piltide pakkimise tulemusena tekkis igast järjestusest MP4 formaadis fail. Pärast kõikidele võimalikele baaskihtidele mõlema järjestusalgoritmi rakendamist sisaldab töökataloog 2 x 59 videofaili. Tuginedes kokkusurumise definitsioonile [36], mille kohaselt on andmete pakkimine sisendandmete voo (lähtevoog või algandmed) teisendamine väiksema suurusega andmevooks (väljund või tihendatud voog), võib eeldada, et paremini järjestatud aegruumi kuubi kihtidest koostatud MP4 on väiksema salvestusmahuga. Ehk mida vähem muudatusi on aja jooksul aktsiate omavahelistes korrelatsioonides toimunud seda paremini õnnestub sisendandmete kokku surumine ja seda väiksem on väljundfaili maht.

Hüpoteetiliselt parimale kihile PCA algoritmi rakendades ja selle järgi järjestades on piltide maht kokku 184,35 KB ja nendest genereeritakse 31,9 KB suurune videofail, seega võib väita, et pakkimine, kui omaette mõõdik täidab oma eesmärki ja seda võib pidada usaldusväärseks.

5.2 Järjestustulemuste valideerimine ja analüüs

Tabelis 1 on toodud erinevate mõõdikute (PCA ME, PCA MP4, BEA ME, BEA MP4) põhjal tekkinud järjekorrad. Efektiivsusmõõdik ME on arvutatud ja MP4 on töökataloogi salvestatud vastava algoritmi rakendamisel, kui vastav ajahetke kiht on olnud baaskihiks.

Tabel 1. Mõõdikute põhjal koostatud järjestused.

Jrk. nr.	PCA ME	PCA MP4	BEA ME	BEA MP4
1	30	30	28	47
2	39	38	30	30
3	40	31	39	38
4	22	34	22	45
5	31	42	45	33
6	34	45	31	28
7	38	40	38	40
8	45	49	33	31
9	33	33	40	39
10	28	39	34	4
11	49	19	55	52
12	55	5	4	22
13	42	22	25	34
14	36	23	49	49
15	25	25	51	59
16	32	32	32	27
17	24	55	36	41
18	51	58	24	51
19	58	10	58	55
20	23	28	42	23
21	35	36	43	29
22	15	57	23	2
23	37	50	35	18
24	50	51	50	44
25	48	41	37	58
26	57	52	41	6
27	43	9	47	1
28	41	3	57	32
29	47	35	6	3
30	19	6	15	48

Jrk. nr.	PCA ME	PCA MP4	BEA ME	BEA MP4
31	29	15	48	26
32	52	8	46	42
33	6	7	52	35
34	27	43	26	15
35	46	24	29	24
36	4	4	19	50
37	44	59	27	36
38	26	11	54	25
39	13	37	13	12
40	17	53	17	9
41	7	48	20	10
42	14	47	11	57
43	16	46	44	56
44	8	13	21	53
45	11	18	14	20
46	54	17	7	54
47	21	20	8	14
48	20	56	16	8
49	59	21	3	5
50	3	29	59	43
51	9	12	18	21
52	18	2	9	37
53	2	27	10	7
54	5	44	2	19
55	10	14	5	46
56	53	16	53	13
57	56	26	1	11
58	12	54	56	17
59	1	1	12	16

Tabelist 1 nähtub, et hüpoteesi põhjal määratud parim baaskiht (kiht nr. 30) on parim ka PCA algoritmi kasutades ning teisel kohal BEA algoritmi kasutades. Kuna BEA algoritmi puhul valitakse rida ja veerg, millest järjestamist alustatakse, juhuslikult, siis sisaldab see mingis osas määramatust ja selle vähendamiseks peaks algoritmi rakendama mitmekordselt. BEA järjestusalgoritmi korduva rakendamise ulatust ja sellest tulenevat efekti tuleks täiendavalt uurida, kuid kuna see ei kuulunud käesoleva töö skoopi, siis seda siin ei teostatud.

Üheks levinud meetodiks erinevate järjestuste omavaheliseks võrdlemiseks on Kemeny-Snelli paariskauguste võrdlemine. Seda meetodit kasutatakse olukordades, kus võistlejad

(antud juhul baaskihid) on järjestatud mitmes erinevas paremusjärjestuses ning leitav paariskaugus näitab võrreldavates järjestustes esinevate lahkarvamuste arvu. Mida suurem on paariskaugus, seda erinevad järjestused on [37].

Üldistatud kujul saab Kemeny-Snelli paariskauguse leida järgmise valemi abil:

$$paariskaugus = 2 \cdot (1 : 1\ viigid) + (1 : 0\ võidud).$$

Viikide all mõeldakse olukorda, kus leidub võistlejate (kihtide) paar selliselt, et ühes järjestuses on üks parem ja teises teine parem. Võitude all mõeldakse olukorda, kus ühes järjestuses on üks parem ja teises järjestuses ollakse viigis. Kuna antud eksperimendis ei saa järjestuse siseselt viike tekkida, siis see osa valemist ei ole relevantne [37].

Konkreetse järjestusega seotud paariskauguste summa annab koondkauguse, mille põhjal saab välja tuua parima järjestuse. Paariskauguste ja koondkauguste arvutused on teostatud R-s, kasutades joonisel 35 toodud koodi [37].

```
setwd("…")
library(ConsRank)
A <- read.table(file ="eelistused.txt", row.names =1)
B <- as.matrix(A)
KS = kemenyd(B)
C <- data.matrix(KS)
write.table(C,file ="paariskaugused.txt")
D <- rowSums(C)
write.table(D,file ="koondkaugused.txt")
```

Joonis 35. Paariskauguste arvutamine.

Saadud paariti võrdlustulemused ning summeerimisel leitud koondkaugused on toodud tabelis 2.

Tabel 2. Järjestuste koondkaugused.

	PCA ME	PCA MP4	BEA ME	BEA MP4	Koondkaugus
PCA ME	0	746	248	1054	2048
PCA MP4	746	0	830	1144	2720
BEA ME	248	830	0	986	2064
BEA MP4	1054	1144	986	0	3184

Tabelist 2 ilmneb, et parimaks järjestuseks võib pidada PCA ME oma, sellele järgneb väiksese vahega BEA ME, ning kõige halvemaks osutus BEA MP4 järgi koostatud järjestus.

Neljast järjestusest kahel, millest üks oli ka valideerimise tulemusena parim järjestus, osutus hüpoteesi põhjal leitud baaskiht parimaks ning ülejäänud kahel juhul tuli teisele kohale. Arvestades kihtide arvu võib öelda, et parima baaskihi hüpotees peab paika vähemalt 95%-lise usaldusnivoo korral.

Valideerimise kokkuvõtteks võib öelda, et kuigi PCA algoritm andis parema tulemuse, siis ei saa kindlalt väita, et see on ainult stabiilsemast algoritmist tingitud. PCA algoritmi paremust kinnitab ka autori visuaalne hinnang järjestatud kuubile. Kui järjestuse tulemuse osas on 95%-line usaldusnivoo piisav, siis käesoleva töö autor soovitab alustada kuubi järjestamist parima baaskihi leidmisest, sest ressursside sääst, mis sellega saavutatakse, on kaalukam argument, kui väike tulemuse ebatäpsus. Autori hinnangul sobib ME kuubi järjestuse mõõdikuks paremini kui MP4, sest see toimis mõlema algoritmi puhul ning seetõttu võib pidada seda usaldusväärsemaks. MP4 mittesobivus võib olla tingitud asjaolust, et minimaalsed muudatused järjestikustel kihtidel ei tähenda alati parima järjestuse olemasolu nendes kihtides antud eksperimendi eesmärki arvestades.

5.3 Eksperimendi tulemuste analüüs ning järeldused

Käesoleva töö raames läbi viidud eksperimendi eesmärgiks oli valitud andmekogu (S&P 500 aktsiate) põhjal binaarsete korrelatsioonimaatriksite genereerimine ning nende järjestamine selliselt, et säiliks andmete ajaline järgnevus, kuid nende omavahelised olemuslikud seosed (korreleerumised) tuleks esile läbi tulemuse visualiseerimise. Kuna antud eksperiment on oma ülesehituselt pigem heuristiline, siis on eksperimendi analüüsimiseks kasutatud heuristiliste meetoditele omaseid kompromisskriteeriume [38].

Optimaalsus ja täielikkus. Eksperimendis kasutati aegruumi kuubi järjestamiseks meetodit, kus järjestati ümber üks kiht, kasutades kahemõõtmelise maatriksi jaoks loodud järjestusalgoritmi ning sellega koos järjestus ümber terve kuup. Kuubi järjestamisel on kitsenduseks ajalise järgnevuse säilitamine, mis ei võimalda klasterdamise algoritme ilma modifikatsioonideta kasutada. Eksperimendis kasutatud meetodi optimaalsemaks muutmiseks püstitati hüpotees, kuidas leida kiiremini soovitud lahendus. Järjestustulemuste valideerimine kinnitas püstitatud hüpoteesi piisaval usaldusnivool. Nii aegruumi kuubi järjestamise õnnestumine kui ka täiendava lihtsustava meetodi lisamine antud käsitlusse lubab töö autoril kinnitada, et eksperiment õnnestus.

Täpsus ja ressurss. Kuigi saadud järjestustulemusi sai omavahel võrreldud ja valideeritud, siis ei saa lõplikult väita, et konkreetsel järjestusmeetodil või eksperimendil on kindel täpsusaste. Tulemuse usaldusvahemiku välja töötamine ei kuulunud ka käesoleva töö eesmärkide hulka. Samas saab väita, et eksperiment suutis leida aegruumi kuubi järjestamiste paremusjärjestused, mida toetasid nii mõõdikud kui visuaalne vaatlus. Suure tõenäosusega saaks täpsust suurendada üle järjestamistega, kuid sellisel juhul kasutatavate algoritmide valik ja ulatus jääb juba järgnevate eksperimentide koosseisu.

Nii nagu ka täpsuse kriteeriumi puhul tuleb reaalset andmetöötlust läbi viies arvestada kehtestatud nõuete ja piirangutega. Ühe kriteeriumi suurendamine käib üldjuhul mõne muu kriteeriumi arvelt. Selleks, et saaks üles ehitada optimaalse aja- ja muu ressursikuluga protsessi, on vaja kindlasti teada võimalikke kokkuhoiu kohti, mis ei käiks ülemäära teiste kriteeriumite arvelt. Antud eksperimendis välja pakutud ja testitud hüpotees, kuidas vältida kõikide võimaluste läbi testimist, on kindlasti üheks selliseks kohaks, mida kasutada saaks.

Kui järjestuseksperimendi üheks eesmärgiks oli visuaalse lahendi genereerimine suurest andmehulgast, siis kaasnevaks tulemuseks võib lugeda ka andmetöötluse ressurssi kokku hoidva andmehulga tekke. Edasiste analüüside ja/või arvutuste teostamine andmekoguga, kus väärtused on kokku koondatud, kaasab vähem ressurssi, kui hõreda järjestamata (ehk hajusa) andmekoguga opereerides.

Järgnevalt on käesoleva töö autor välja toonud eksperimendi käigus esile kerkinud probleemsed kohad ja nende võimalikud lahendused.

Probleem 1. Analüüsimiseks suur andmete hulk suurendab ka visualiseerimise keerukust. Visuaalne interpretatsioon peaks olema lihtne, informatiivne ja looma lisandväärtust.
Lahendus 1. Probleemi lahendamiseks saab pärast järjestamist huvipakkuvaid segmente eraldiseisevalt analüüsida ja visualiseerida.

Probleem 2. Kui eksperimendis on oluline järjestuse täpsust tõsta, siis võib järjestamise algoritmi ebastabiilsus lõpptulemust negatiivselt mõjutada.
Lahendus 2. Probleemi lahendamiseks tuleks minimeerida volatiilsust järjestusalgoritmi sees. Selleks tuleks näiteks BEA algoritmi korduvalt rakendada, kuid ei ole teada, millal tulemus stabiliseerub. Nimetatud analüüs ei olnud käesoleva töö osa.

Probleem 3. Visualiseerimise teostamine rakendusega R eeldab heal tasemel R-i oskust, mille puudumine võib detailsete visuaalide koostamisel takistuseks osutuda.
Lahendus 3. Probleemi vältimiseks saab kasutada kombineeritult mitut tarkvara või tellida R-i sisesed valmisrakendused programmeerimiskeele R valdajatelt.

Probleem 4. Cubix ei suuda teostada visualiseerimist, kui sisendandmete maht on suur.
Lahendus 4. Probleemi vältimiseks saab kasutada kombineeritult mitut või siis hoopis mõnda sobivamat visualiseerimise rakendust.

Magistritöö peamised järeldused:

- Korrelatsiooniseoste esitamine binaarsel kujul on sobilik nii aegruumi kuubi järjestamise kui ka visualiseerimise sisendiks. Juhuslike suuruste ehk müra eemaldamine on järjestuse koondumiseks ja informatiivseks visuaaliks oluline. Teistsuguse algandmete hulga puhul võib analüüsi käigus tekkida ka nii positiivsete kui negatiivsete korrelatsioonide kuupi, mida siis vastavalt konkreetse analüüsi eesmärkidele tuleks eraldi uurida.
- Aegruumi kuupi on võimalik järjestada kiht-kihi haaval, rakendades kuubi ühele kihile maatriksi järjestamise algoritmi, millega koos järjestuvad ümber ka ülejäänud kihtide read ja veerud. Sellisel juhul säilib ka kuubi ajaline järjestus.
- Järjestatud aegruumi kuubi visualiseerimine võimaldab korrelatsiooniseoste olemasolu ja muutumist aja jooksul tuvastada. Küll aga eeldab visuaalide viimistlemine R-s keskmisest paremat R-i oskust, Cubix on seevastu kergemini õpitav ja kasutatav (aja faktor). Samas on visualiseerimise aspektist R-l tugev eelis andmemahtude osas.

Käesoleva töö ühe võimaliku edasiarendusena võiks uurida, kas mõnda klasterdamise algoritmi saab modifitseerida selliselt, et kuubi üks telg säilitaks oma järjestuse. S&P 500 asemel võiks andmestiku võtta näiteks krüptorahade valdkonnast, kuid samas ei ole meetodi kasutamine vaid finantsandmetega piiratud.

Lisaks on ka visualiseerimise osas võimalikke tulevikusuundasid kindlasti palju – nii tarkvara arenduse, disaini kui andmeanalüüsi valdkonnas.

6 Kokkuvõte

Käesoleva töö eesmärgiks oli välja töötada üks võimalik meetod suurte korrelatsioonimaatriksite rea jada järjestamiseks ning lõpptulemi visualiseerimiseks. Töö eesmärgist lähtuvalt valiti eksperimendi andmeteks S&P 500 indeksisse kuuluvad aktsiad, mille omavaheliste väga tugevate positiivsete korrelatsiooniseosete põhjal 59-l järjestikusel kuul moodustati aegruumi kuup. Meetodi, mida kuubi järjestamiseks rakendati, peamine idee seisnes ühele kihile (maatriksile) järjestusalgoritmi rakendamises, millega koos järjestusid ümber ülejäänud kuubi kihtide read ja veerud. Sellisel viisil järjestades säilis kuubi kihtide ajaline järgnevus, mis oli antud töö üheks peamiseks kitsenduseks.

Parimal viisil järjestatud aegruumi kuup visualiseeriti rakendustega R ja Cubix. Visuaalse vaatluse põhjal võib väita, et käesolevas töös välja pakutud meetod töötab ja kuubi elemendid tõepoolest koonduvad läbi aja kuubi erinevatesse osadesse. Parim järjestuse tulemus sai kinnitatud ka valideerimismõõdikute poolt.

Magistritöö peamised tulemused on:

- aegruumi kuubi järjestamiseks meetodi väljatöötamine ja valideerimine;
- korrelatsiooniseoseid sisaldava järjestatud aegruumi kuubi visualiseerimine;
- aegruumi kuubi järjestamise optimeerimiseks autori poolt püstitatud hüpoteesi sõnastamine ja valideerimine.

Kõik magistritöös püstitatud eesmärgid said täidetud ning käesolevas töös tutvustatud meetodi võib võtta kasutusele lisaväärtust loovana väärtpaberiportfelli analüüsimisel või selle riskide juhtimisel.

Kasutatud kirjandus

[1] P. M. Simon ja C. Turkay, „Hunting High and Low: Visualising Shifting Correlations in Financial Markets,“ *Computer Graphics Forum,* kd. 37, nr 3, pp. 479-490, 2018.

[2] C. Samantha, K. Soramäki ja A. Laubsch, „A network-based method for visual identification of systemic risks,“ *Journal of Network Theory in Finance,* kd. 2(1), p. 67–101, 2016.

[3] E. Eessaar, „Lõputöö soovitused,“ [Online]. Available: http://staff.ttu.ee/~eessaar/loputood_soovitused.html. [Accessed April 2021].

[4] T. G. Kolda, B. W. Bader, „Tensor Decompositions and Application,“ *Society for Industrial and Applied Mathematics,* kd. 51, nr 3, p. 455–500, 2009.

[5] X. Bi, X. Tang, Y. Yuan, Y. Zhang, A. Qu, „Tensors in Statistics,“ *Annual Review of Statistics and Its Applicatio,* kd. 8, 2021.

[6] G. Golnari, Z.-L. Zhang, D. Boley, „Markov fundamental tensor and its applications to network analysis,“ *Linear Algebra and its Applications,* kd. 564, pp. 126-158, 2019.

[7] J. Li, B. Uçar, Ü. V. Çatalyürek, J. Sun, K. Barker, R. Vuduc, „Efficient and effective sparse tensor reordering,“ *ICS '19: Proceedings of the ACM International Conference on Supercomputing*, Phoenix Arizona, 2019.

[8] B. Bach, E. Pietriga, J-D. Fekete, „Visualizing dynamic networks with matrix cubes,“ *CHI '14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Toronto Ontario Canada, 2014.

[9] I. Liiv, „Seriation and matrix reordering methods: An historical overview,“ *Statistical Analysis and Data Mining ,* kd. 3, nr 2, pp. 69-139, 2010.

[10] I. Liiv, R. Öpik, J. Ubi, J. Stasko, „Visual matrix explorer for collaborative seriation,“ *WIREs Computational Statistics,* kd. 4, nr 1, pp. 85-97, 2012.

[11] L. Parsons, E. Haque, H. Liu, „Subspace clustering for high dimensional data: a review,“ *ACM SIGKDD Explorations Newsletter,* kd. 6, nr 1, 2004.

[12] A. McCallum, K. Nigram, L. H. Ungar, „Efficient clustering of high-dimensional data sets with application to reference matching,“ *KDDoo: The Second Annual International Conference on Knowledge Discovery in Data*, Boston Massachusetts USA, 2000.

[13] M. Behrisch, B. Bach, N. H. Riche, T. Schreck ja J.-D. Fekete, „Matrix Reordering Methods for Table and Network Visualization,“ *Computer Graphics Forum,* kd. 35, nr 3, pp. 693-716, 2016.

[14] G. Martia, F. Nielsenc, M. Bińkowskid ja P. Donnat, „A review of two decades of correlations, hierarchies, networks and clusteringin financial markets,“ 2019. [Online]. Available: https://arxiv.org/abs/1703.00485. [Accessed April 2021].

[15] A. Rea ja W. Rea, „Visualization of a stock market correlation matrix,“ *Physica A: Statistical Mechanics and its Applications,* kd. 400, pp. 109-123, 2014.

[16] P. J. Groenen ja P. H. Franses, „Visualizing time-varying correlations across stock markets,“ *Journal of Empirical Finance,* kd. 7, nr 2, pp. 155-172, 2000.

[17] „Bloomberg Terminal,“ [Online]. Available: https://en.wikipedia.org/wiki/Bloomberg_Terminal. [Accessed April 2021].

[18] M. Halperin, „Exploring Correlations with Bloomberg,“ [Online]. Available: https://lippincottlibrary.wordpress.com/2015/02/18/ruble-regression-exploring-correlations-with-bloomberg/. [Accessed April 2021].

[19] „ETFreplay.com,“ ETFreplay.com, [Online]. Available: https://www.etfreplay.com/correlation.aspx. [Accessed April 2021].

[20] Z. Galáž, J. Mekyska ja Z. Smékal, „Correlation analysis tool,“ Brno University of Technology, 2016. [Online]. Available: http://splab.cz/en/download/software/software-pro-korelacni-analyzu. [Accessed April 2021].

[21] B. E. Reese, D. Lofgreen, „Spatial Analysis 3D User's Guide,“ 2008. [Online]. Available: https://labs.nri.ucsb.edu/reese/benjamin/Files/SpatialAnalysisUserGuide.pdf. [Accessed April 2021].

[22] A. Sauga, „Statsionaarsed aegread,“ [Online]. Available: http://www.sauga.pri.ee/portfoolio/OkonomeetriaLoengStatsionaarsedAegread.pdf . [Accessed April 2021].

[23] S. Ko, I. Cho, S. Afzal, C. Yau, J. Chae, A. Malik, K. Beck, Y. Jang, W. Ribarsky ja D. Ebert, „A Survey on Visual Analysis Approaches for Financial Data,“ *Computer Graphics Forum,* kd. 35, nr 3, pp. 599-617, 2016.

[24] „RStudio,“ [Online]. Available: https://rstudio.com/. [Accessed April 2021].

[25] „S&P 500 stock data,“ Kaggle, [Online]. Available: https://www.kaggle.com/camnugent/sandp500. [Accessed April 2021].

[26] „S&P 500,“ Wikipedia, [Online]. Available: https://et.wikipedia.org/wiki/S%26P_500. [Accessed April 2021].

[27] „S&P 500 indeks,“ Capital Com SV Investments Ltd, [Online]. Available: https://capital.com/et/s-p-500-indeks-definitsioon. [Accessed April 2021].

[28] A. Hayes, „Linear Relationship Definition,“ Investopedia, [Online]. Available: https://www.investopedia.com/terms/l/linearrelationship.asp. [Accessed April 2021].

[29] W. Kenton, „Pearson Coefficient,“ Investopedia, [Online]. Available: https://www.investopedia.com/terms/p/pearsoncoefficient.asp. [Accessed April 2021].

[30] D. Rowntree, Statistics without tears: an introduction for nonmathematicians, London: Penguin Books, 2000.

[31] M. Pihlak, Klassikaline ja mitteparameetriline matemaatiline statistika, Tallinn: TTÜ kirjastus, 2018.

[32] „Getting Things in Order: An Introduction to the R Package seriation," [Online]. Available: https://cran.r-project.org/web/packages/seriation/vignettes/seriation.pdf. [Accessed April 2021].

[33] „Cubix," [Online]. Available: https://aviz.fr/Reseach/Cubix. [Accessed April 2021].

[34] J.-D. Fekete, *Personal communicaton*. [March 13, 2021].

[35] W.T. McCormick Jr., S.B. Deutsch, J. J. Martin, P. J. Schweitzer, „Identification of data structures and relationships by matrix reordering techniques," 1969.

[36] D. Salomon, „Data Compression," Springer, 1998. [Online]. Available: https://link.springer.com/content/pdf/bfm%3A978-1-4757-2939-9%2F1.pdf. [Accessed April 2021].

[37] I. Liiv, „ITB8802 Täppismeetodid otsustuste vastuvõtmisel," Tallinn, 2021.

[38] „Heuristic (computer science)," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Heuristic_(computer_science). [Accessed April 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Helene Lillemägi

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Aegruumi kuubi järjestamine ja selle rakendamine finantsturul", mille juhendaja on Innar Liiv
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

10.05.2021

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Programmi lähtekood

```
rm(list = ls())
setwd("…")

library(seriation)
library(reshape2)
library(data.table)
library(ggplot2)
library(rgl)
library(magick)

data <- read.csv("all_stocks_5yr.csv")

#algandmete maatriksi kujule viimine ja pooliku infoga aktsiate eemaldamine
data = within(data.frame(date = as.Date(data$date), close =
as.numeric(data$close), name = as.character(data$Name)), {x =
as.numeric(factor(name))})
reordered_data = subset(data[order(data$x, decreasing = F), ], select=-c(x))
reshaped_data = reshape(reordered_data, timevar = "name", idvar = "date",
direction = "wide")
clear_data <- reshaped_data[ , colSums(is.na(reshaped_data)) == 0]

#päevaste tootluste leidmine
daily_returns <- data.frame(clear_data[-1,1], apply(clear_data[,-1], 2,
function(x) diff(x)/head(x,-1)))
colnames(daily_returns)[1] <- "date"

#kuude lõikes jaotamine
data_list = split(daily_returns, format(daily_returns$date, "%Y-%m"))

#juhul, kui andmestiku esimene ja viimane kuu on keskmisest väiksema päevade
arvuga, siis eemaldame need
if (nrow(data_list[[1]]) < ave(sapply(data_list, NROW))) {
  data_list[[1]] <- NULL
}
if (nrow(data_list[[length(data_list)]]) < ave(sapply(data_list, NROW))) {
  data_list[[length(data_list)]] <- NULL
}

#kuupäevade ja esimese rea eemaldamine
new_data_list = lapply(data_list, function(x) x[-1,-1])

#korrelatsioonimaatriksite leidmine
cor_list = lapply(new_data_list, cor, method = "pearson")

# i - read
# j - veerud
# k – kihid
```

```
#peadiagonaalidele nullide kirjutamine ja olulisuse nivoo rakendamine
cor_matrix_list <- NULL
for (k in 1:length(cor_list)) {
  cor_matrix_list[[k]] = as.matrix(cor_list[[k]])
  diag(cor_matrix_list[[k]]) <- 0
  cor_matrix_list[[k]][cor_matrix_list[[k]] > 0.9] <- 1
  cor_matrix_list[[k]][cor_matrix_list[[k]] <= 0.9] <- 0
}

#olulisuse nivoo rakendamine e. müra eemaldamine
for (i in 1:nrow(cor_matrix_list[[k]])) {
  for (j in 1:ncol(cor_matrix_list[[k]])) {
    sum = 0
    for (k in 1:length(cor_matrix_list)) {
      sum = sum + cor_matrix_list[[k]][[i,j]]
    }
    if (sum < length(cor_matrix_list)*0.05) {
      for (k in 1:length(cor_matrix_list)) {
        cor_matrix_list[[k]][[i,j]] <- 0
      }
    }
  }
}

#järjestamiseks parima kuu leidmine
best_sum = 0
best_month = 0
for (k in 1:length(cor_matrix_list)) {
  sum_all = 0
  for (i in 1:nrow(cor_matrix_list[[k]])) {
    for (j in 1:ncol(cor_matrix_list[[k]])) {
      elem_sum = 0
      if (cor_matrix_list[[k]][i,j] != 0) {
        for (l in 1:length(cor_matrix_list)) {
          if (l != k) {
            elem_sum = elem_sum + cor_matrix_list[[l]][i,j]
          }
        }
      }
      sum_all = sum_all + elem_sum
    }
  }
  if (sum_all > best_sum) {
    best_sum = sum_all
    best_month = k
  }
}

# l - abimuutuja iga kihi järgi järjestuses leidmiseks

#järjestamine kõikide kihtide järgi
```

```
data3d <- NULL
new_data3d <- NULL
measures <- NULL
# 1. baaskihi valik, ehk igale kiht on korra baaskihiks
for (l in 1:length(cor_matrix_list)) {

  # 2. järjestamise meetodi rakendamine baaskihile
  seriate_matrix <- seriate(cor_matrix_list[[l]], method = "PCA")

  # 3. ridade ja veergude järjestus
  row_order <- get_order(seriate_matrix, dim=1)
  col_order <- get_order(seriate_matrix, dim=2)

  #ülejäänud kihtide ümberjärjestamine
  month_cor_list <- NULL
  for (k in 1:length(cor_matrix_list)) {

    # 4. ridade ja veergude järjestamine vastavalt esimese kihi korrastatud
maatriksile
    month_cor <- as.matrix(cor_matrix_list[[k]])[row_order, col_order]

    # 5. iga kuu kohta pidi salvestamine kausta
    png(paste0("PCA", sprintf("%02d", k) ,".png"))
    pimage(month_cor, main = paste0("PCA", sprintf("%02d", k)), legend.only =
FALSE)
    dev.off()

    #koostame kuudest listi ME arvutamiseks
    month_cor_list[k] <- list(month_cor)

    # 6. 3D joonise jaoks lisame kogu info ühte andmestikku
    longData <- reshape2::melt(month_cor_list[k])
    longData <- longData[longData$value!=0,]
    if (dim(longData)[1] == 0) {
      next
    } else {
      data3d <- data.frame(longData, check.names=FALSE)
      data3d <- cbind(k, data3d)
      if (k == 1) {
        new_data3d <- data3d
      } else {
        new_data3d <- rbind(new_data3d, data3d)
      }
    }
  }

  #andmestiku salvestamine
  colnames(new_data3d)[1] <- "timestep"
  colnames(new_data3d)[4] <- "correlation"
  assign(paste("new_data3d", l, sep = "_") , new_data3d)
```

```
  # 7. kausta salvestatud piltidest mp4 faili tegemine
  av::av_encode_video(list.files("V:/10 Sales Pension/Helene/Kooli
asjad/Magistritöö/New", 'PCA.+.png'), framerate = 10, output = paste("PCA",
l, ".mp4", sep = ''))

  # 8. piltide kustutamine
  file.remove(list.files("V:/10 Sales Pension/Helene/Kooli
asjad/Magistritöö/New", 'PCA.+.png'))

  # 9. järjestuse ME arvutamine
  ME = 0
  for (k in 1:length(month_cor_list)) {
    for (i in 1:length(row_order)) {
      for (j in 1:length(col_order)) {
        if (i == 1) {
          a = 0
        } else {
          a = month_cor_list[[k]][[i-1,j]]
        }
        if (j == 1) {
          b = 0
        } else {
          b = month_cor_list[[k]][[i,j-1]]
        }
        if (k == 1) {
          c = 0
        } else {
          c = month_cor_list[[k-1]][[i,j]]
        }
        if (i == length(row_order)) {
          d = 0
        } else {
          d = month_cor_list[[k]][[i+1,j]]
        }
        if (j == length(col_order)) {
          e = 0
        } else {
          e = month_cor_list[[k]][[i,j+1]]
        }
        if (k == length(month_cor_list)) {
          f = 0
        } else {
          f = month_cor_list[[k+1]][[i,j]]
        }
        ME = ME + 1/2 * month_cor_list[[k]][[i,j]] * (a + b + c + d + e + f)
      }
    }
  }

  #ME lisamine ME koondtabelisse
  measures_l <- cbind(l, ME)
```

```
  measures <- rbind(measures, measures_l)
}

#ME koondtabeli salvestamine
write.csv(measures, "measures_PCA.csv", row.names=FALSE)

#hüpoteetiliselt parima kihi 3D graafik
plot3d(new_data3d_30$Var1, new_data3d_30$Var2, new_data3d_30$timestep, type =
"p", radius = .2, xlab="", ylab="", zlab="t")
axes3d("z", lwd=2.5, z=c(0,length(cor_matrix_list)), col = "black", labels =
FALSE, tick = FALSE)

# hüpoteetiliselt parima kihi andmestiku salvestamine töökataloogi
write.csv(new_data3d_30, "data30.csv", row.names=FALSE)
```