

Tilastotieteen pro gradu -tutkielma

Järjestyslukuihin perustuva eksponentiaallinen tasoitus ja  
volatiliteetin ennustaminen

Ari Väisänen

Jyväskylän yliopisto  
Matematiikan ja tilastotieteen laitos  
14.6.2009

# Tiivistelmä

Järjestyslukuihin perustuva eksponentiaalinen tasoitus ja volatilitietin ennustaminen

Ari Väisänen

Tilastotiede

Pro gradu -tutkielma

Matemaattis-luonnontieteellinen tiedekunta

Matematiikan ja tilastotieteen laitos

Jyväskylän yliopisto

Ohjaaja: Jukka Nyblom

Kesäkuu 2009

Sivumäärä: 40 sivua + liitteet 9 sivua

**Tiivistelmä:** Tässä tutkielmassa esitetään järjestyslukuihin perustuva eksponentiaalisen tasoituksen menetelmä. Järjestyslukuihin perustuvilla menetelmillä on usein robusteja ominaisuuksia, joten yksi tutkimuksen päämäärinä on selvittää, onko uusi menetelmä yksinkertaista eksponentiaalista tasoitusta robustimpi. Menetelmiä verrataan aluksi simuloitujen lokaalisen tason mallin ja GARCH(1,1)-mallin aikasarjojen ennustamisessa. Järjestyslukuihin perustuva tasoitus toimii hyvin varsinkin poikkeavia havaintoja sisältävän lokaalisen tason mallin tapauksessa. Tämän jälkeen tutkitaan, soveltuuko uusi menetelmä osakkeiden volatilitietin ennustamiseen. Järjestyslukuihin perustuvalla tasoituksella, yksinkertaisella tasoituksella ja GARCH(1,1)-mallilla ennustetaan osakeaikasarjoista viikottaista volatilitiettiä. Menetelmiä hyvyttä tarkastellaan ennusteiden virheneliösummien ja jäännösten avulla. Uuden menetelmän tulokset volatilitietin ennustamisessa ovat vaihtelevia.

**Abstract:** This pro gradu thesis introduces a rank based exponential smoothing method for time series forecasting. Rank based methods are known to have robust qualities, so one objective in this thesis is to study if the new method is more robust than simple exponential smoothing. First simple exponential smoothing and rank based smoothing are applied to local level model and GARCH(1,1)-model simulations. The rank based method works especially well when forecasting a local level model that has outliers. Then we forecast volatility from empirical stock data. Rank based smoothing, simple exponential smoothing and a GARCH(1,1)-model are used to forecast weekly volatility. Forecasting performances are compared using sum of squared errors and residuals. Results for the rank based smoothing are inconclusive.

**Avainsanoja:** Eksponentiaalinen tasoitus, ennustaminen, GARCH-mallit, poikkeavat havainnot, robustisuus, volatilitietti.

**Keywords:** Exponential smoothing, forecasting, GARCH-models, outliers, robustness, volatility.

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>3</b>
<b>2</b>	<b>Aikasarjamalleja</b>	<b>5</b>
2.1	ARMA- ja ARIMA-mallit . . . . .	5
2.2	ARCH- ja GARCH-mallit . . . . .	6
<b>3</b>	<b>Yksinkertainen eksponentiaalinen tasoitus</b>	<b>8</b>
3.1	Taustaa eksponentiaalisesta tasoituksesta . . . . .	8
3.2	Yksinkertaisen tasoituksen perusteet . . . . .	9
3.3	Yksinkertaisen tasoituksen optimaalisuus . . . . .	10
3.4	Tasoitusparametrin valinta . . . . .	12
<b>4</b>	<b>Järjestyslukuihin perustuva eksponentiaalinen tasoitus</b>	<b>13</b>
<b>5</b>	<b>Volatiliteetti</b>	<b>15</b>
5.1	Volatiliteetin ennustaminen . . . . .	16
5.2	Ennusteiden vertailu . . . . .	17
5.3	Mallin hyvyys . . . . .	18
<b>6</b>	<b>Simulointikokeita</b>	<b>19</b>
6.1	Lokaalisen tason malli . . . . .	19
6.2	GARCH(1,1)-malli . . . . .	22
<b>7</b>	<b>Osakkeiden tuottojen volatiliteetin ennustaminen</b>	<b>25</b>
7.1	Viikottaisen volatiliteetin ennustaminen . . . . .	25
7.2	Ennusteet . . . . .	27
7.2.1	Patient Safety Tech . . . . .	27
7.2.2	The Coca-Cola Company . . . . .	29
7.2.3	Nokia . . . . .	29
7.3	Tulokset . . . . .	32
7.3.1	Jäännökset . . . . .	33

<b>8 Yhteenveto ja johtopäätökset</b>	<b>37</b>
<b>Lähteet</b>	<b>39</b>
<b>Liitteet</b>	<b>41</b>
<b>A R-koodit</b>	<b>41</b>
A.1 Järjestyslukuihin perustuva eksponentiaalinen tasoitus . . . . .	41
A.2 Lokaalisen tason mallin simulointi ja ennustaminen . . . . .	43
A.3 GARCH(1,1)-mallin simulointi ja volatiliteetin ennustaminen .	45
A.4 Volatiliteetin ennustaminen osakeaikasarjasta . . . . .	47

# Luku 1

## Johdanto

Aikasarjojen ennustaminen on tärkeää useilla aloilla, kuten kansantaloudessa, inventaarion kontrollissa, myynnin ennustamisessa ja osakemarkkinoilla. Ennustusmenetelmiä on useita, eikä ole olemassa yhtä metodologiaa, joka soveltuisi kaikkiin tapauksiin. Ennustusmenetelmät voidaan lajitella kolmeen ryhmään: subjektiivisiin menetelmiin, yhden muuttujan menetelmiin sekä monen muuttujan menetelmiin (Chatfield 2004, 73–74). Subjektiiviset ennusteet tehdään arvioimalla, intuitiolla ja käyttämällä hyväksi saatavilla olevaa oleellista tietoa. Yhden muuttujan menetelmissä ennusteet perustuvat vain menneisyyden havaintoihin sovitettuun malliin. Siten esimerkiksi yrityksen myynnin ennusteet perustuvat vain aikaisempiin myyntilukuihin eivätkä ota huomioon muita tekijöitä. Monimuuttujamenetelmissä ennusteet riippuvat ainakin osittain muista, selittävien muuttujien aikasarjoista.

Eksponentiaalisen tasoituksen menetelmät ovat yhden muuttujan ennustusmenetelmiä. Nämä menetelmät ovat yksinkertaisia ja helppokäyttöisiä, mutta niiden antamat ennusteet eivät välttämättä ole optimaalisia, jos aineistossa on poikkeavan pieniä tai suuria havaintoja. Robusteilla menetelmillä pyritään vähentämään tällaisten poikkeavien havaintojen kielteisiä vaikutuksia, ja useilla järjestyslukuihin perustuvilla menetelmillä on robusteja ominaisuuksia (Hettmansperger & McKean 1998).

Tässä pro gradu -tutkielmassa esitellään järjestyslukuihin perustuva eksponentiaalisen tasoituksen menetelmä. Tätä menetelmää käytetään ensin simuloitujen sarjojen ennustamiseen ja sen jälkeen osakkeiden tuottojen volatilitteen ennustamiseen. Volatilitetti kuvaa vaihtelua ja se on yksi tärkeimmistä muuttujista taloudellisessa päätöksenteossa. Eksponentiaalinen tasointus soveltuu usein hyvin osakkeiden volatilitteen ennustamiseen, mutta ennusteet voivat olla huonoja, jos aikasarjassa on poikkeavia havaintoja. Tarkoituksena on tutkia, toimiiko järjestyslukuihin perustuva tasointus tällaisissa tilanteissa paremmin.

Tutkielma etenee seuraavasti: Aluksi esitellään lyhyesti ARMA- ja ARIMA-mallit, joita tarvitaan myöhemmin käsiteltäessä eksponentiaalisen tasoituksen optimaalisuutta. Tämän jälkeen esitellään perusteet ARCH- ja GARCH-malleista, joista GARCH(1,1)-mallia käytetään tuottojen volatiliiteetin mallintamisessa. Sitten kerrotaan yksinkertaisesta eksponentiaalisesta tasoituksesta. Menetelmän perusteiden jälkeen käsitellään tarkemmin minkälaisien aikasarjojen ennustamiseen yksinkertainen tasoitus on optimaalinen, ja miten menetelmän tasoitusparametri tulisi valita. Tämän jälkeen esitellään järjestyslukuihin perustuva eksponentiaalisen tasoituksen menetelmä. Sitten kerrotaan, mitä rahoitusinstrumentin volatiliiteetilla tarkoitetaan ja miksi sen ennustaminen on tärkeää sekä näytetään kuinka tutkielmassa esitetyillä menetelmillä voidaan ennustaa volatiliiteettia.

Tämän jälkeen uutta järjestyslukuihin perustuvaa menetelmää käytetään lokaalisen mallin mukaisiin, poikkeavia havaintoja sisältäviin simuloituihin aikasarjoihin sekä GARCH(1,1)-mallilla simuloituihin aikasarjoihin. Uuden menetelmän tuottamien ennusteiden hyvyyttä verrataan erityisesti yksinkertaisen eksponentiaalisen tasoituksen ennusteisiin.

Lopuksi järjestyslukuihin perustuvaa eksponentiaalista tasoitusta käytetään volatiliiteetin ennustamiseen osakkeiden tuottojen aikasarjoista. Näitäkin ennusteita verrataan yksinkertaisen tasoituksen sekä myös GARCH(1,1)-mallin antamiin ennusteisiin. Tuloksia verrataan graafisesti ja ennustusvirheistä laskettujen suureiden avulla. Mallien sopivuutta tarkastellaan myös jäännösten avulla.

Kaikki simuloinnit, mallien sovitukset ja jäännöstarkastelut tehdään R-ohjelmistolla (<http://www.R-project.org>). Osa itse kirjoitetuista R-koodeista on liitteessä A.

# Luku 2

## Aikasarjamalleja

Tässä luvussa kerrotaan aikasarjamalleista, joita tarvitaan myöhemmän teoriaosuuden ymmärtämiseen tai joita käytetään volatilitietin ennustamiseen. Aluksi esitetään usein aikasarjojen mallintamisessa käytettävät ARMA- ja ARIMA-mallit. Tämän jälkeen kerrotaan ARCH- ja GARCH-malleista.

### 2.1 ARMA- ja ARIMA-mallit

ARMA- ja ARIMA-mallien teoriaa tarvitaan yksinkertaisen eksponentiaalisen tasoituksen optimaalisuutta käsiteltäessä ja niitä voi myös tarvita esimerkiksi ARCH- tai GARCH-malleja sovitettaessa. Nämä mallit eivät kuitenkaan sovi heteroskedastisen prosessin mallintamiseen, joten niitä ei tässä tutkielmassa käytetä volatilitietin ennustamiseen.

Aikasarjan  $y_t$  sanotaan olevan autoregressiivinen prosessi kertalukua  $p$ , AR( $p$ ), jos

$$y_t = c + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \epsilon_t,$$

missä  $c$  on vakiotermin ja  $\epsilon_t$  on sarja riippumattomia ja samoin jakautuneita satunnaismuuttujia odotusarvolla 0 ja varianssilla  $\sigma^2$ . Tällaista prosessia  $\epsilon_t$  nimitetään usein valkoiseksi kohinaksi, josta tässä tutkielmassa käytetään merkitään  $\{\epsilon_t\} \sim \text{WN}(0, \sigma^2)$ .

Aikasarjan  $y_t$  sanotaan olevan liukuvan keskiarvon prosessi kertalukua  $q$ , MA( $q$ ), jos

$$y_t = c + \epsilon_t - \theta_1 \epsilon_{t-1} - \dots - \theta_q \epsilon_{t-q}, \quad (2.1)$$

missä  $c$  ja sarja  $\epsilon_t$  määritellään kuten aiemmin.

Edelleen aikasarjan  $y_t$  sanotaan olevan autoregressiivinen liukuvan keskiarvon prosessi kertalukua  $p$  ja  $q$ , ARMA( $p, q$ ), jos

$$y_t = c + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \dots - \theta_q \epsilon_{t-q},$$

missä  $c$  ja sarja  $\epsilon_t$  määritellään kuten aiemmin.

Näitä malleja käytetään yleensä vain, kun aikasarja on stationaarinen. Aikasarjan sanotaan olevan stationaarinen, jos sen odotusarvossa ei ole systemaattista vaihtelua, eli sarjassa ei ole trendiä. Sarjassa ei myöskään saa olla jaksollista vaihtelua eikä sarjan varianssissa saa olla systemaattista vaihtelua. Stationaarisuuden matemaattinen määritelmä esitetään kirjassa *The Analysis of Time Series - An Introduction* (Chatfield 2004, 34–36). ARMA-mallit voidaan kuitenkin laajentaa sopimaan epästationaarisiiin aikasarjoihin ottamalla aikasarjasta peräkkäisiä erotuksia. Näitä malleja kutsutaan ARIMA-malleiksi, ARIMA( $p,d,q$ ), jossa  $p$  on autoregressiivisen osan kertaluku,  $d$  on peräkkäisten erotusten lukumäärä ja  $q$  on liukuvan keskiarvon kertaluku.

Näiden mallien ominaisuuksista, parametrien estimoinnista ja malleilla ennustamisesta on tietoa kirjassa *Analysis of Financial Time Series* (Tsay 2005, 32–91).

## 2.2 ARCH- ja GARCH-mallit

ARCH-mallit (autoregressive conditional heteroscedastic) ja GARCH-mallit (generalized ARCH) ovat ehdollisia heteroskedastisia malleja. Niissä prosessin varianssi voi muuttua ajan kuluessa ja niitä käytetäänkin usein esimerkiksi osakkeen tuottojen volatilititeetin mallintamiseen. Tämän kappaleen teoria perustuu kirjaan *Analysis of Financial Time Series* (Tsay 2005, 103–107).

Oletetaan, että  $y_t$  on aikasarja, jossa ei ole trendiä tai kausivaihtelua. Tällainen sarja voi olla esimerkiksi osakkeen kurssien ensimmäisen kertaluvun erotuksien luonnollisten logaritmien sarja. ARCH( $m$ )-mallissa, eli  $m$ . kertaluvun ARCH-mallissa, oletetaan että

$$y_t = \sigma_t \epsilon_t$$

ja

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \cdots + \alpha_m y_{t-m}^2,$$

missä  $\alpha_0 > 0$ ,  $\alpha_i \geq 0$  kaikille  $i > 0$  ja  $\epsilon_t$  on sarja riippumattomia ja samoin jakautuneita satunnaismuuttujia, joiden odotusarvo on 0 ja varianssi 1. Käytännössä  $\epsilon_t$  oletetaan usein normaalisti jakautuneeksi tai t-jakautuneeksi. Parametreille annetaan rajoitukset, jotta varianssi saa järkeviä arvoja. Sen halutaan esimerkiksi aina olevan nollaa suurempi. Tässä mallissa aikaisempien havaintojen suuret arvot johtavat suureen ehdolliseen varianssiin, eli suuri vaihtelu voi saada aikaan lisää suurta vaihtelua. Tällainen käytös sopii yhteen monien empiiristen finanssiaikasarjojen kanssa.



GARCH( $m,s$ )-mallissa oletetaan, että

$$y_t = \sigma_t \epsilon_t \quad (2.2)$$

ja

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \cdots + \alpha_m y_{t-m}^2 + \beta_1 \sigma_{t-1}^2 + \cdots + \beta_s \sigma_{t-s}^2, \quad (2.3)$$

missä  $\alpha_0 > 0$ ,  $\alpha_i \geq 0$ ,  $\beta_i \geq 0$  ja  $\sum_{i=1}^{\max(m,s)} (\alpha_i + \beta_i) < 1$ . Oletetaan myös, että  $\alpha_i = 0$ , kun  $i > m$  ja  $\beta_j = 0$ , kun  $j > s$ . Sarja  $\epsilon_t$  määritellään kuten ylempänä.

Molempien mallien parametrit voidaan estimoida maksimoimalla ehdollinen logaritminen uskottavuusfunktio. Sekä ARCH- että GARCH-malleilla tiedetään olevan joitakin heikkouksia (Tsay 2005, 106). Molemmissa malleissa negatiiviset ja positiiviset osakkeen arvon muutokset vaikuttavat samalla tavalla volatilitettiin. Käytännössä kuitenkin tiedetään, että näin ei tapahdu. ARCH-mallin antamat volatilitettiin ennusteet ovat myös usein liian korkeita, koska ne reagoivat hitaasti suuriin yksittäisiin osakkeen arvon muutoksiin. Volatilitettiin ennustamiseen kehitetäänkin koko ajan uusia menetelmiä.

## Luku 3

# Yksinkertainen eksponentiaalinen tasoitus

Seuraavaksi kerrotaan tarkasti yksinkertaisesta eksponentiaalisesta tasoituksesta, johon myöhemmin esitettävä järjestyslukuihin perustuva eksponentiaalinen tasoitus perustuu. Aluksi kerrotaan eksponentiaalisen tasoituksen historiasta. Tämän jälkeen käydään tarkemmin läpi yksinkertaisen eksponentiaalisen tasoituksen menetelmä, näytetään minkälaisten aikasarjojen ennustamiseen yksinkertainen tasoitus on optimaalinen ja lopuksi kerrotaan, kuinka menetelmässä tarvittava tasoitusparametri voidaan valita.

### 3.1 Taustaa eksponentiaalisesta tasoituksesta

Eksponentiaalisesti tasoitukseksi kutsutaan ennustusmenetelmien yleistä luokkaa, jossa ennustukset lasketaan yksinkertaisten päivitysyhtälöiden avulla. Tasoitus voidaan tehdä ilman parametrin mallin sovittamista aikasarjaan. Eksponentiaalisen tasoituksen menetelmiä on kehitetty useita ja ne perustuvat Brownin (1959,1963), Holtin (1957) ja Wintersin (1960) töihin. Niiden lähtökohta on, että seuraavan ajanhetken ennuste muodostetaan painotettuna summuna edellisestä havainnosta ja edellisen havainnon ennusteesta. Näissä menetelmissä uudemmilla havainnoilla on suurempi vaikutus ennusteeseen kuin vanhemmilla havainnoilla.

Tunnetuimpia eksponentiaalisen tasoituksen menetelmiä ovat yksinkertainen eksponentiaalinen tasoitus, Holtin lineaarinen tasoitus ja Holt-Winters-tasoitus. Yksinkertainen eksponentiaalinen tasoitus soveltuu parhaiten aikasarjoihin, joissa ei ole kausivaihtelua tai trendiä. Holtin lineaarinen tasoitus ottaa huomioon lineaarisen trendin, kun taas Holt-Winters-tasoitus ottaa huomioon myös kausivaihtelun. Näistä menetelmistä on kehitetty usei-

ta muunnoksia esimerkiksi tilanteisiin, joissa trendi tai kausivaihtelu ovat multiplikatiivisia tai joissa menetelmän täytyy olla robustimpi.

Eksponentiaalisen tasoituksen menetelmät ovat laajasti käytettyjä teollisuudessa ja taloudessa. Niillä on saatu hyviä tuloksia ennustuskilpailuissa (Makridakis & Hibon 2000), ja tulokset ovat usein hyviä monimutkaisempiinkin ennustusmenetelmiin verrattaessa. Näiden menetelmien etuina ovat varsinkin yksinkertaisuus ja alhaiset kustannukset.

## 3.2 Yksinkertaisen tasoituksen perusteet

Yksinkertaista eksponentiaalista tasoitusta tulisi käyttää vain aikasarjoihin, joissa ei ole kausivaihtelua tai systemaattista trendiä. Jos aikasarjassa on kausivaihtelua tai trendi, voidaan niiden ennustamiseen käyttää muita eksponentiaalisen tasoituksen menetelmiä. Kausivaihtelun tai trendin voi myös yrittää poistaa muunnoksilla, jolloin saadaan aikasarja, jonka ennustamiseen yksinkertainen eksponentiaalinen tasoitus sopii.

Tässä tutkielmassa merkintä  $\hat{y}_T(h)$  tarkoittaa ajanhetkellä  $T$  tehtyä,  $h$ :n askeleen päähän ulottuvaa ennustetta. Olkoon  $y_t$  aikasarja, joka on havaittu aikapisteissä  $t = 1, \dots, T$  ja jossa ei ole trendiä tai kausivaihtelua. Tällöin seuraavan arvon  $y_{T+1}$  eksponentiaalisen tasoituksen ennuste määritellään rekursiivisena kaavana

$$\hat{y}_T(1) = \alpha y_T + (1 - \alpha)\hat{y}_{T-1}(1), \quad 0 < \alpha < 1, \quad (3.1)$$

missä  $\alpha$  on yksinkertaiseen tasoitukseen liittyvä tasoitusparametri.

Määritelmästä seuraa, että ennuste pysyy samana, kun aikasarjaa ennustetaan useamman askeleen päähän. Jos viimeisin havainto on ajanhetkellä  $T$  ja halutaan ennuste kahden askeleen päähän, tuntematon  $y_{T+1}$  korvataan ennusteellaan  $\hat{y}_T(1)$ . Silloin ennuste saa muodon

$$\begin{aligned} \hat{y}_T(2) &= \alpha \hat{y}_T(1) - \alpha \hat{y}_T(1) + \hat{y}_T(1) \\ &= \hat{y}_T(1). \end{aligned}$$

Yleisesti  $\hat{y}_T(h) = \hat{y}_T(1)$ .

Malli voidaan esittää myös muodossa, jossa ennustetta korjataan edellisen ennusteen ennustusvirheellä

$$\hat{y}_T(1) = \hat{y}_{T-1}(1) + \alpha[y_T - \hat{y}_{T-1}(1)], \quad 0 < \alpha < 1. \quad (3.2)$$

Kaavasta (3.2) nähdään, että ennuste reagoi aina jäljessä datassa mahdollisesti olevaan trendiin. Seuraavaa ennustetta korjataan vain tietyllä prosentilla aikaisemmasta ennustusvirheestä.

Suoralla sijoituksella nähdään, että ennuste on painotettu keskiarvo aikaisemmista havainnoista. Uudemmat havainnot saavat isommat kertoimet kuin vanhemmat havainnot

$$\begin{aligned}\hat{y}_T(1) &= \alpha y_T + \alpha(1 - \alpha)y_{T-1} + (1 - \alpha)^2 \hat{y}_{T-2}(1) \\ &= \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots\end{aligned}$$

Menetelmää voidaan soveltaa, kun käytössä on vähintään kaksi havaintoa ja  $\hat{y}_1(1)$ :lle on valittu sopiva arvo. Yleensä valitaan  $\hat{y}_1(1) = y_1$ . Alkuarvon valintaan voidaan kuitenkin käyttää myös muita menetelmiä. Yksi tapa on laskea muutaman ensimmäisen havainnon keskiarvo ja käyttää sitä alkuarvona. Mitä pienempi tasoitusparametrin arvo valitaan, sitä enemmän alkuarvon valinta vaikuttaa tuleviin ennusteisiin (Makridakis, Wheelwright & Hyndman 1998, 151). Kun tasoitusparametri saa arvoja, jotka ovat lähellä nollaa, ennuste ei reagoi voimakkaasti muutoksiin. Kun  $\alpha$  saa arvoja, jotka ovat lähellä ykköstä, tasoitusta ei tapahdu paljon ja tapahtuneet muutokset saavat isot painokertoimet.

### 3.3 Yksinkertaisen tasoituksen optimaalisuus

Yksinkertaisen tasoituksen sanotaan olevan optimaalinen, jos sen antama yhden askeleen ennuste on sama kuin ennuste, joka saataisiin johdettua aikasarjan taustalla olevasta oletetusta mallista ottamalla siitä ehdollinen odotusarvo  $E(y_{T+1} | y_T, y_{T-1}, \dots, y_1)$ .

Olkoon aikasarjan taustalla oleva malli muotoa (Chatfield 2004, 77)

$$y_t = \mu + \alpha \sum_{j < t} \epsilon_j + \epsilon_t,$$

missä  $\{\epsilon_t\} \sim \text{WN}(0, \sigma^2)$ . Sarja  $y_t$  on epästationaarinen, mutta peräkkäiset erotukset  $(y_{t+1} - y_t)$  muodostavat stationaarisen MA(1)-prosessin (2.1)

$$\begin{aligned}y_t - y_{t-1} &= \mu + \alpha \sum_{j < t} \epsilon_j + \epsilon_t - (\mu + \alpha \sum_{j < t-1} \epsilon_j + \epsilon_{t-1}) \\ &= \epsilon_t + (\alpha - 1)\epsilon_{t-1}.\end{aligned}\tag{3.3}$$

Siten  $y_t$  on ARIMA(0,1,1)-prosessi. Seuraavaksi näytetään, että yksinkertainen tasoitus on optimaalinen tällaisen prosessin ennustamiseen.

Yhtälö (3.3) voidaan esittää muodossa  $y_t = y_{t-1} + \epsilon_t + (\alpha - 1)\epsilon_{t-1}$ . ARIMA-mallin ennuste saadaan tästä yhtälöstä korvaamalla sarjan  $\epsilon_t$  tulevat arvot nolllalla, sarjan  $y_t$  tulevat arvot niiden ehdollisilla odotusarvoilla ja sarjojen

$\epsilon_t$  ja  $y_t$  nykyiset ja menneisyyden arvot havaituilla arvoillaan. Aluksi huomataan, että

$$\hat{y}_{T-1}(1) = y_{T-1} + (\alpha - 1)\epsilon_{T-1}.$$

Tästä saadaan

$$y_T - \hat{y}_{T-1}(1) = y_T + \epsilon_T + (\alpha - 1)\epsilon_T - y_{T-1} - (\alpha - 1)\epsilon_{T-1} = \epsilon_T.$$

Tätä käyttämällä yhden askeleen ennuste saadaan muotoon

$$\begin{aligned}\hat{y}_T(1) &= y_T + (\alpha - 1)\epsilon_T \\ &= y_T + (\alpha - 1)[y_T - \hat{y}_{T-1}(1)] \\ &= \alpha y_T + (1 - \alpha)\hat{y}_{T-1}(1),\end{aligned}$$

joka on samaa muotoa kuin yksinkertainen eksponentiaallinen tasoitus.

Yksinkertainen eksponentiaallinen tasoitus on optimaalinen myös lokaalisen tason mallille

$$\begin{aligned}y_t &= \mu_t + \epsilon_t, \\ \mu_t &= \mu_{t-1} + \eta_t, \quad t = 1, 2, \dots,\end{aligned}$$

missä  $\{\epsilon_t\} \sim \text{WN}(0, \sigma_\epsilon^2)$ ,  $\{\eta_t\} \sim \text{WN}(0, \sigma_\eta^2)$  ja sarjat  $\epsilon_t$  ja  $\eta_t$  ovat keskenään riippumattomia. Tässä  $y_t = \mu_t + \epsilon_t$  on havaintoyhtälö ja  $\mu_t = \mu_{t-1} + \eta_t$  tilayhtälö. Kun  $T$  ei ole pieni, mallille voidaan Kalmanin suotimen avulla näyttää, että seuraavan tilan ennuste on muotoa

$$\hat{\mu}_T(1) = \hat{\mu}_{T-1}(1) + \alpha\epsilon_T,$$

missä  $\alpha = (-q + \sqrt{q^2 + 4q})/2$  (Harvey 1993, 127). Tässä  $q = \sigma_\eta^2/\sigma_\epsilon^2$ , jota kutsutaan myös signaali-kohinasuhteeksi. Tilan ennuste vastaa siis yksinkertaista eksponentiaalista tasoitusta. Kyseinen lokaalisen tason malli sopii yksinkertaisuudestaan huolimatta useisiin aikasarjoihin.

Voidaan myös näyttää, että yksinkertainen eksponentiaallinen tasoitus on optimaalinen esimerkiksi tila-avaruusmallille, jossa on vain yksi virhetermi

$$\begin{aligned}y_t &= \mu_{t-1} + \epsilon_t, \\ \mu_t &= \mu_{t-1} + \alpha\epsilon_t, \quad t = 1, 2, \dots\end{aligned}$$

On myös mahdollista muodostaa malleja, joissa varianssi ei ole vakio, ja joihin yksinkertainen eksponentiaallinen tasoitus on optimaalinen (Chatfield, Koehler, Ord & Snyder 2001, 150–154).

Yllä esitetyt ominaisuudet selittävät sitä, miksi yksinkertainen eksponentiaallinen tasoitus on käyttökelpoinen monissa tilanteissa.

### 3.4 Tasoitusparametrin valinta

Tasoitusparametrin  $\alpha$  valintaan ei ole formaalisti oikeaa menetelmää. Sen valinta riippuu aikasarjan ominaisuuksista ja valitsemiseen voi käyttää omaa harkintaa. Yleensä arvo valitaan väliltä 0.05–0.30 (Harvey 1993, 111), jolloin ennusteeseen vaikuttavat useat aikaisemmat havainnot.

Jos käytettävissä on riittävästi havaintoja, voidaan tasoitusparametrikksi valita arvo, joka minimoi yhden askeleen ennusteiden virheneliösumman (SSE). Virheneliösumma on muotoa

$$\sum_{t=2}^T e_t^2, \quad (3.4)$$

missä  $e_t = y_t - \hat{y}_{t-1}(1)$ . Ennusteiden virheneliösumman minimointi on tavallisin tapa löytää tasoitusparametri, koska summa on  $\alpha$ :n suhteen derivoituva funktio ja sen minimoiva  $\alpha$  on yleensä helppo löytää.

Muidenkin ennustusvirheiden mittojen optimoiminen on mahdollista. Tasoitusparametrin arvon voi yrittää löytää esimerkiksi minimoimalla lausekkeen  $\sum_{t=2}^T |e_t|$ , missä  $e_t$  on määritelty kuten edellä.

On myös mahdollista antaa tasoitusparametrin arvon muuttua ajan kuluessa, jolloin menetelmä reagoi paremmin muutoksiin aikasarjassa. Tätä menetelmää ei kuitenkaan käsitellä tarkemmin tässä tutkielmassa.

## Luku 4

# Järjestyslukuihin perustuva eksponentiaallinen tasoitus

Seuraavaksi esitellään järjestyslukuihin perustuva eksponentiaallinen tasoitus (Nyblom 2005). Tässä menetelmässä yksinkertaista eksponentiaalista tasoitusta sovelletaan havaintojen sijasta niiden järjestyslukuihin. Menetelmän R-koodi on liitteessä A.1.

Oletetaan, että käytössä on aikasarja  $y_t$ ,  $t = 1, \dots, T$ . Näistä havainnoista saadaan järjestysluvut  $r_{1,T}, r_{2,T}, \dots, r_{T,T}$ , missä  $r_{i,T}$  on havainnon  $y_i$  järjestysluku joukossa  $y_1, \dots, y_T$ . Tässä havainnon järjestysluku on keskijärjestysluku, joka saadaan kaavasta

$$r_{i,T} = \frac{1}{2} + \sum_{t=1}^T [I(y_t < y_i) + \frac{1}{2}I(y_t = y_i)], \quad (4.1)$$

jossa

$$I(y_t < y_i) = \begin{cases} 1 & , \text{ kun } y_t < y_i \\ 0 & , \text{ kun } y_t \geq y_i \end{cases}$$

ja

$$I(y_t = y_i) = \begin{cases} 1 & , \text{ kun } y_t = y_i \\ 0 & , \text{ kun } y_t \neq y_i. \end{cases}$$

Nyt yksinkertaisen eksponentiaalisen tasoituksen menetelmää (3.1) sovelletaan järjestysluvun ennusteeseen

$$\hat{r}_{T,T}(1) = \alpha r_{T,T} + (1 - \alpha)\hat{r}_{T-1,T-1}(1), \quad 0 < \alpha < 1.$$

Tässä tulevan havainnon järjestyslukua ennustetaan edeltävän havainnon järjestysluvulla ja edeltävän havainnon järjestysluvun ennusteella. Merkintä  $\hat{r}_{T,T}(1)$  tarkoittaa siis havainnon  $y_{T+1}$  ennustettua järjestyslukua joukossa

$y_1, \dots, y_{T+1}$ . Saatu järjestysluvun ennuste ei yleensä ole kokonaisluku. Jos  $\hat{r}_{T-1, T-1}(1) \leq T-1$ , niin  $\hat{r}_{T, T}(1) < T$ . Täten induktiolla voidaan näyttää, että jos  $\hat{r}_{1, 1}(1) = 1$ , niin  $\hat{r}_{T, T}(1) < T$  kaikille  $T = 2, 3, \dots$ .

Järjestysluvun ennusteesta täytyy vielä saada itse aikasarjan ennuste  $\hat{y}_T(1)$ . Aloitetaan kirjoittamalla saatu järjestysluvun ennuste muotoon  $\hat{r}_{T, T}(1) = k + \lambda$ , missä  $k$  on  $\hat{r}_{T, T}(1)$ :tä lähinnä oleva pienempi tai yhtä suuri kokonaisluku ja  $\lambda$  sitä vastaava desimaaliosa, jolloin  $0 \leq \lambda < 1$  ja  $k < T$ . Nyt ennuste on muotoa

$$\hat{y}_T(1) = (1 - \lambda)y_{k, T} + \lambda y_{k+1, T}, \quad (4.2)$$

missä merkinnät  $y_{1, T} \leq \dots \leq y_{T, T}$  tarkoittavat havaintojen  $y_1, y_2, \dots, y_T$  järjestettyjä arvoja.

Menetelmän päivitys tapahtuu seuraavalla tavalla: Kun havaitaan uusi arvo  $y_T$ , täytyy seuraavan havainnon  $y_{T+1}$  järjestysluvun ennustetta varten olla käytössä havainnon  $y_T$  järjestysluku  $r_{T, T}$ , joka saadaan kaavasta (4.1), ja tämän järjestysluvun ennuste  $\hat{r}_{T-1, T-1}(1)$ . Tämän lisäksi täytyy päivittää järjestetty sarja  $y_{1, T} \leq \dots \leq y_{T, T}$ ,  $T = 1, 2, \dots$ . Kun uusi arvo  $y_{T+1}$  saapuu, sen paikka päivitettyssä sarjassa saadaan kaavasta

$$r_{T+1, T+1} = 1 + \sum_{t=1}^T I(y_t < y_{T+1}),$$

missä  $I(y_t < y_T)$  määritellään kuten edellä.



# Luku 5

## Volatiliteetti

Tässä luvussa kerrotaan, mitä volatiliteetilla tarkoitetaan ja miksi sen ennustaminen on tärkeää. Sen jälkeen esitetään, kuinka yksinkertaisella tasoituksella, järjestyslukuihin perustuvalla tasoituksella ja GARCH(1,1)-mallilla voidaan ennustaa volatiliteettia.

Volatiliteetiksi sanotaan usein vaihtelua mittaavaa tunnuslukua. Tässä esitelmässä volatiliteetti on rahoitusinstrumentin, kuten osakkeen, lainan tai option, logaritmisien tuottojen keskihajonta. Logaritminen tuotto saadaan jakamalla rahoitusinstrumentin arvo edeltävän ajanhetken arvolla ja laskemalla suhteen luonnollinen logaritmi. Volatiliteettia pidetään riskin mittarina ja sen ennustaminen on tärkeää esimerkiksi sijoittajille ja markkinoiden tutkijoille. Se on noussut keskeiseen asemaan johdannaismarkkinoiden voimakkaan kasvun takia. Johdannainen on rahoitusinstrumentti, jonka arvo perustuu johonkin toiseen rahoitusinstrumenttiin. Volatiliteetin ennusteet ovat tärkeitä esimerkiksi optioiden hinnoittelussa. Optio on sopimus, jossa option asettaja lupaa myydä tai ostaa kohteen sovittuun hintaan tiettyinä hetkenä tai ajanjaksona. Useimmiten optio on sitä kalliimpi, mitä suurempi volatiliteetti kohteella on ennen maturiteettia, eli option voimassaoloaika (Vuorenmaa 2008, 3). Tämä johtuu siitä, että yleensä optiolla on isompi todennäköisyys tuottaa voittoa, jos option kohteen arvo vaihtelee voimakkaasti.

Rahoitusinstrumentin volatiliteettia ei havaita suoraan. Jos käytettävissä on esimerkiksi osakkeen päivittäisten logaritmisien tuottojen aikasarja, päivittäistä volatiliteettia ei havaita, koska yhdeltä päivältä saadaan vain yksi havainto. Tämän takia volatiliteetin ennustamiseen käytettävien menetelmien hyvyttä on vaikea vertailla. Volatiliteetille on kuitenkin mahdollista laskea estimaatti, jos käytössä on yhden päivän ajalta useampia havaintoja. Voidaan myös tarkastella esimerkiksi viikottaista volatiliteettia, jolloin viikottaisen volatiliteetin estimaatti saadaan laskettua päivittäisistä tuotoista. Tässä tutkielmassa osakeaikasarjoista ennustetaan viikottaista volatiliteettia

ja saatuja ennusteita verrataan päivittäisistä arvoista laskettuihin viikottaisen volatiliteetin estimaatteihin. Volatiliteetin estimaattina käytetään myös viikottaisia shokkeja, joista kerrotaan myöhemmin.

Vaikka volatiliteettia ei havaita suoraan, sillä on tiettyjä ominaisuuksia, joita voidaan havaita tuottoja tarkastellessa (Tsay 2005, 98–99). Tuotoissa esiintyy volatiliteettiryhmittymiä, eli volatiliteetti on korkealla joinakin aikoina ja matalalla joinakin aikoina. Volatiliteetti muuttuu ajassa jatkuvana, eli volatiliteetin hyppäykset ovat harvinaisia. Volatiliteetin ei myöskään oleteta hajaantuvan äärettömyyteen. Tämä tarkoittaa sitä, että volatiliteetti on stationaarinen. Nämä ominaisuudet ovat tärkeitä, kun kehitetään malleja volatiliteetin mallintamiseen ja ennustamiseen.

## 5.1 Volatiliteetin ennustaminen

Tämän kappaleen merkinnät perustuvat osittain artikkeliin Volatility forecasting with smooth transition exponential smoothing (Taylor 2004, 274). Seuraavaksi kerrotaan, kuinka eri menetelmillä ennustetaan rahoitusinstrumentin tuottojen volatiliteettia eli keskihajontaa. Luvun lopussa kerrotaan vielä ennusteiden vertaamisesta ja mallin hyvyyden tarkastelusta.

Olkoon  $P_t$  rahoitusinstrumentin arvo hetkellä  $t$ . Rahoitusinstrumentin logaritminen tuotto hetkellä  $t$  on

$$u_t = \log \left( \frac{P_t}{P_{t-1}} \right).$$

Määritellään seuraavaksi logaritmisien tuottojen ehdollinen varianssi ajanhetkellä  $t$  ja ehdolla  $I_{t-1}$ , missä  $I_{t-1}$  kuvaa ajanhetken  $t - 1$  ja sitä edeltävien tuottojen sisältämää informaatiota. Tällöin ehdollinen varianssi on

$$\sigma_t^2 = \text{var}(u_t \mid I_{t-1}).$$

Tämä voidaan tulkita virhetermin  $y_t$  varianssiksi, kun määritellään

$$y_t = u_t - E(u_t \mid I_{t-1}).$$

Virhetermiä  $y_t$  kutsutaan usein hinnan shokiksi.

Nyt halutaan ennustaa shokin volatiliteettia eli ehdollista keskihajontaa. Suosituimpia menetelmiä volatiliteetin mallintamiseen ovat ARCH- ja GARCH-mallit. Tässä tutkielmassa niistä käytetään GARCH(1,1)-mallia, jolloin kaavat (2.2) ja (2.3) ovat muotoa

$$y_t = \sigma_t \epsilon_t$$

ja

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \quad (5.1)$$

missä parametrit määritellään kuten aiemmin GARCH-mallin yhteydessä. Jos käytössä ovat osakkeen hinnan shokit  $y_t$ ,  $t = 1, \dots, T$  sekä  $\hat{\alpha}_0$ ,  $\hat{\alpha}_1$  ja  $\hat{\beta}_1$  ovat GARCH(1,1)-mallin estimoidut parametrit, niin yhden askeleen ennuste varianssille on yksinkertaisesti

$$\hat{\sigma}_{T+1}^2 = \hat{\alpha}_0 + \hat{\alpha}_1 y_T^2 + \hat{\beta}_1 \hat{\sigma}_T^2.$$

Volatiliteettia voidaan ennustaa myös yksinkertaisella eksponentiaalisella tasoituksella. Näin tehdään esimerkiksi suositussa RiskMetrics-menetelmässä (Tsay 2005, 290). RiskMetrics on kokoelma malleja, joita käytetään sijoitus-salkun riskin mittaamiseen. Siinä yksinkertaisen tasoituksen ennuste varianssille on

$$\hat{\sigma}_{T,E}^2(1) = \alpha \hat{\sigma}_{T-1,E}^2(1) + (1 - \alpha) y_T^2, \quad 0 < \alpha < 1. \quad (5.2)$$

Järjestyslukuihin perustuvaa eksponentiaalista tasoitusta käytetään tässä tutkielmassa volatilitietin ennustamiseen soveltamalla menetelmää shokkien neliöihin  $y_t^2$ , jolloin niistä saatava tasoitettu arvo on varianssin ennuste kaavan (4.2) mukaisesti,

$$\hat{\sigma}_{T,R}^2(1) = (1 - \lambda) y_{k,T}^2 + \lambda y_{k+1,T}^2. \quad (5.3)$$

Volatilitietin ennusteet saadaan laskemalla varianssien ennusteista neliöjuuret.

## 5.2 Ennusteiden vertailu

Yksi tapa verrata eri menetelmien antamia ennusteita on jakaa aikasarja kahteen osaan, alustusosaan ja testiosaan. Sitten vain alustusosaa käytetään menetelmän parametrien estimointiin sekä mallin sovittamiseen. Tämän jälkeen ennusteet tehdään testiosalle. Koska testiosaa ei käytetä mallin luomiseen, nämä ennusteet ovat aitoja. Ennusteiden tarkkuutta kuvaavat suureet, kuten ennusteiden virheneliösumma (3.4), lasketaan vain testiosalle. Näiden suureiden avulla voidaan vertailla ennusteiden hyvyttä.

Yksinkertaisen eksponentiaalisen tasoituksen antamille ennusteille ei ole helppoa tapaa laskea ennustevälejä. Yksi tapa on löytää malli, jolle tasoitus on optimaalinen, kuten ARIMA(0,1,1) luvussa (3.3), ja laskea ennustevälit tämän mallin avulla. Tällä menetelmällä on kuitenkin ongelmia (Makridakis, Wheelwright & Hyndman 1998, 177–178), joten monien ohjelmien laskemien ennustevälien suhteen on hyvä olla varovainen. Tässä esitelmässä ennustevälejä ei ole laskettu minkään menetelmän ennusteille, vaan ennusteiden hyvyttä tarkastellaan graafisesti ja ennusteiden virheneliösummien avulla.

### 5.3 Mallin hyvyys

Kappaleessa (5.1) esitettyjen mallien sopivuutta aineistoon voidaan tarkastella standardoitujen jäännösten,  $\tilde{y}_t = \frac{y_t}{\hat{\sigma}_{t-1}(1)}$ , avulla. Jos malli on sopiva, jäännösten pitäisi olla valkoista kohinaa (Tsay 2005, 109). Tätä voidaan tutkia esimerkiksi tarkastelemalla jäännöksiä graafisesti sekä autokorrelaatioiden avulla. Jäännösten neliöiden välillä ei myöskään pitäisi olla autokorrelaatiota. Jäännökset voivat joissain tapauksissa saada erittäin suuria arvoja, jolloin autokorrelaatiot ovat lähellä nollaa, vaikka jäännösten välillä olisi riippuvuutta. Tämän takia tutkitaan myös jäännösten järjestyslukujen autokorrelaatioita.

# Luku 6

## Simulointikokeita

Tässä luvussa järjestyslukuihin perustuvaa eksponentiaalista tasoitusta käytetään simuloituihin aikasarjoihin. Ensin simuloidaan lokaalisen tason mallia, johon asetetaan satunnaisiin paikkoihin poikkeavia havaintoja. Näistä arvoista ennustetaan latenttia prosessia, jossa poikkeavia havaintoja ei ole. Toisessa kokeessa simuloidaan GARCH(1,1)-mallilla aikasarjoja eri parametrien arvoilla ja simuloiduista arvoista ennustetaan prosessin volatilitteettia. Uuden menetelmän toimivuutta verrataan molempien simulointikokeiden yhteydessä yksinkertaiseen eksponentiaaliseen tasoitukseen.

### 6.1 Lokaalisen tason malli

Simuloidaan lokaalisen tason mallia

$$\begin{aligned}y_t &= \mu_t + \epsilon_t, \\ \mu_t &= \mu_{t-1} + \eta_t, \quad t = 1, 2, \dots,\end{aligned}$$

missä  $\{\epsilon_t\} \sim \text{WN}(0, \sigma_\epsilon^2)$ , mutta jossa voi olla poikkeavia arvoja, ja  $\{\eta_t\} \sim \text{WN}(0, \sigma_\eta^2)$  sekä sarjat  $\epsilon_t$  ja  $\eta_t$  ovat keskenään riippumattomia (Nyblom 2005). Aiemmin luvussa (3) osoitettiin, että yksinkertainen eksponentiaalinen tasoitus on optimaalinen ennustettaessa lokaalisen tason mallin mukaista aikasarjaa yksi askel eteepäin. Tässä simulointikokeessa verrataan yksinkertaisen tasoituksen ja järjestyslukuihin perustuvan tasoituksen toimivuutta, kun prosessissa on poikkeavia havaintoja. Aikasarjoja simuloidaan eri  $q$ :n arvoilla, missä  $q = \sigma_\eta^2 / \sigma_\epsilon^2$ , ja myös poikkeavien arvojen lukumääriä vaihdellaan. Simulointeja suoritetaan yksi jokaista  $q$ :n arvon ja poikkeavien havaintojen lukumäärän yhdistelmää kohti. Poikkeavat  $\epsilon_t$ :t saavat arvoja  $\pm 20\sigma_\epsilon$  ja niiden sijainnit ovat satunnaisia. Simuloiduissa aikasarjoissa on 100 havaintoa.

Ennusteet tehdään aikasarjan  $y_t$  pohjalta ja ennusteiden hyvyyttä verrataan latenttiin prosessiin  $\mu_t$ . Tämä tehdään tarkastelemalla molempien menetelmien virheneliösummia, jotka lasketaan kaavalla  $\sum_{t=2}^{100} [\mu_t - \hat{y}_{t-1}(1)]^2$ . Yksinkertaisen tasoituksen tasoitusparametri saadaan minimoimalla lauseke  $\sum_{t=2}^{100} [y_t - \hat{y}_{t-1}(1)]^2$  käyttämällä kaikkia havaintoja. Järjestyslukuihin perustuvan tasoituksen tasoitusparametri saadaan minimoimalla järjestyslukujen ennusteiden virheneliösumma, jossa havainnon järjestysluvusta vähennetään järjestysluvun ennuste ja lasketaan tämän neliö. Simuloitua sarjaa ei siis tässä jaeta alustus- ja testiosaan, näin tehdään vasta osakeaikasarjoja käsiteltäessä. Tasoitusparametrien arvoja ei esitetä tässä jokaiselle simuloinnille erikseen. Jos samalla  $q$ :n arvolla ja poikkeavien havaintojen määrällä tehdään useita simulointeja, saadut tasoitusparametrien arvot voivat vaihdella simulointikertojen välillä huomattavasti. Kaikkia simulointeja tarkastelemalla voidaan sanoa, että yksinkertaisen tasoituksen tasoitusparametri saa arvoja väliltä 0.01 – 0.80 ja järjestyslukuihin perustuvan tasoituksen parametri on välillä 0.20 – 0.80.

Olkoon nyt  $SSE_E$  yksinkertaisen eksponentiaalisen tasoituksen virheneliösumma ja  $SSE_R$  järjestyslukuihin perustuvan tasoituksen virheneliösumma. Suhde  $SSE_E/SSE_R$  ilmoitetaan taulukossa 6.1. Luvut voidaan tulkita uuden menetelmän tehokkuudeksi verrattuna yksinkertaiseen tasoitukseen. Kun suhde on yli 1, järjestyslukuihin perustuva menetelmä on virheneliösummalla mitattuna tuottanut tarkempia ennusteita kuin yksinkertainen tasoitus.

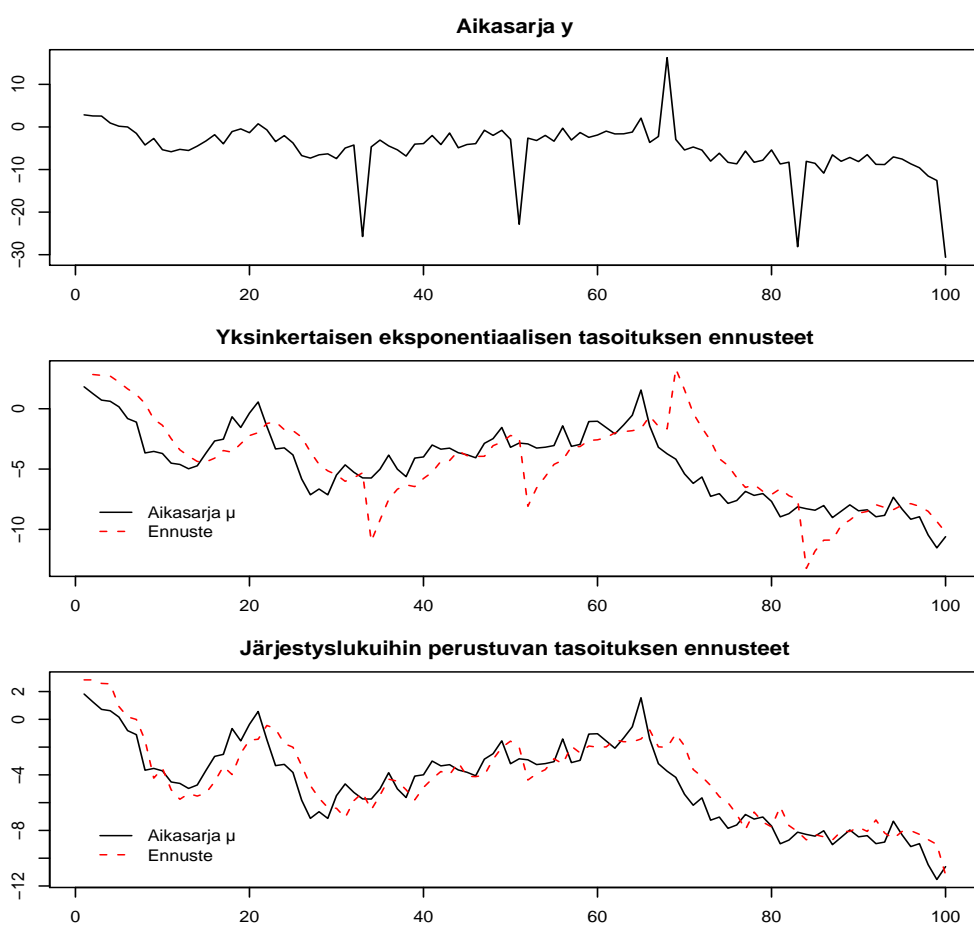
Taulukosta nähdään, että uusi menetelmä on jonkin verran tehottomampi, kun simuloitussa sarjassa ei ole poikkeavia havaintoja. Kun sarjassa on poikkeavia havaintoja ja  $q$  on 0.1, 1 tai 5, uusi menetelmä on yksinkertaista tasoitusta tehokkaampi. Kun  $q = 10$ , eli kun latentin prosessin varianssi on suurta verrattuna poikkeavat havainnot sisältävään prosessiin, menetelmien virheneliösummat ovat hyvin lähellä toisiaan. Muutamissa tapauksissa uusi menetelmä on huomattavasti yksinkertaista tasoitusta tehokkaampi.

Kaikkia simulointeja ei esitetä graafisesti. Yksi niistä on kuvassa 6.1, josta nähdään hyvin kuinka menetelmät toimivat. Kuvan simuloinnille  $q = 1$  ja poikkeavien havaintojen lukumäärä on 5. Kuvasta nähdään, että järjestyslukuihin perustuva menetelmä palaa poikkeavien havaintojen jälkeen yksinkertaista tasoitusta nopeammin latentin aikasarjan  $\mu_t$  tasolle. Tämän simuloinnin R-koodi on liitteessä A.2.

Simulointeja toistettiin tässä esitettyjen tulosten lisäksi useita kertoja ja tulokset olivat joka kerta hyvin samanlaisia. Jos simuloinneissa asetetaan poikkeavien havaintojen arvot poikkeavammiksi kuin  $\pm 20\sigma_\epsilon$ , järjestyslukuihin perustuva menetelmä on yksinkertaiseen tasoitukseen verrattuna vieläkin tehokkaampi.

Taulukko 6.1: Uuden tasoituksen tehokkuus. Aikasarjan pituus  $N=100$ .

Poikkeavien havaintojen lukumäärä				
$q$	0	1	5	10
0.1	0.680	1.754	2.075	1.389
1.0	0.898	1.432	2.566	2.138
5.0	0.869	3.016	1.065	1.200
10	0.988	0.912	1.003	1.263



Kuva 6.1: Lokaalisen tason mallilla simuloitu aikasarja  $y_t$  ( $q = 1$ , poikkeavien havaintojen määrä on 5) ja siitä eri menetelmillä saadut ennusteet kuvissa aikasarjan  $\mu_t$  kanssa.

## 6.2 GARCH(1,1)-malli

Simuloidaan GARCH(1,1)-mallin mukaisia aikasarjoja ja ennustetaan saaduista aikasarjoista volatilitteettia. Käytetään ennusteiden tekemiseen järjestyslukuihin perustuvaa tasoitusta, yksinkertaista eksponentiaalista tasoitusta ja esimerkin vuoksi myös simuloituun sarjaan sovitettua GARCH(1,1)-mallia. Järjestyslukuihin perustuvan tasoituksen ennusteita verrataan kuitenkin tässäkin erityisesti yksinkertaiseen tasoitukseen. Volatilitteetti on nyt tunnettu, joten eri menetelmien hyvyttä voidaan verrata helposti esimerkiksi ennusteiden virheneliösumman avulla. Tämä ei ole mahdollista empiirisen aineiston kanssa, jolloin aikasarjan volatilitteettia ei havaita vaan se täytyy estimoida.

Simuloidaan muotoa (2.2) ja (2.3) olevaa GARCH(1,1)-mallia eri parametrien  $\alpha_1$  ja  $\beta_1$  arvoilla, kun  $\epsilon_t$  on normaalisti jakautunut. Parametri  $\alpha_0$  saa kaikissa simuloinneissa saman arvon  $10^{-6}$ . Simuloinnit tehdään R-ohjelmiston fGarch-paketin garchSim-funktiolla. Simuloiduissa sarjoissa on 200 havaintoa. Näistä simuloinneista saadaan sekä havaintojen aikasarja että havainnot aikaansaaneen prosessin varianssin aikasarja, jonka neliöjuureen havainnoista ennustettua volatilitteettia voidaan verrata. Nyt ajatellaan, että simuloitut havainnot ovat kappaleen (5.1) mukaisin merkinnöin shokkeja  $y_t$ . Näistä havainnoista ennustetaan volatilitteettia yksinkertaisen tasoituksen menetelmällä (5.2) ja järjestyslukuihin perustuvalla menetelmällä (5.3). Yksinkertaisen tasoituksen tasoitusparametri saadaan minimoimalla lauseke  $\sum_{t=2}^{200} [y_t^2 - \hat{\sigma}_{t-1,E}^2(1)]^2$  ja järjestyslukuihin perustuvan tasoituksen tasoitusparametri minimoimalla  $\sum_{t=2}^{200} [y_t^2 - \hat{\sigma}_{t-1,R}^2(1)]^2$ . Tasoitusparametrien estimointiin käytetään siis kaikkia havaintoja. Tässä  $y_t^2$  ajatellaan varianssin estimaattiksi. Parametrit voi estimoida myös muilla tavoilla, mutta teknisten syiden takia tässä tutkielmassa käytetään näitä lausekkeita. Simuloituihin havaintoihin sovitetaan myös GARCH(1,1)-malli, jolla ennustetaan volatilitteettia kaavan (5.1) mukaisesti.

Nyt  $\sigma_t$  on prosessin  $y_t$  tunnettu volatilitteetti, joten yksinkertaisen tasoituksen ja järjestyslukuihin perustuvan tasoituksen virheneliösummat ovat  $\sum_{t=2}^{200} [\sigma_t - \hat{\sigma}_{t-1,E}(1)]^2$  ja  $\sum_{t=2}^{200} [\sigma_t - \hat{\sigma}_{t-1,R}(1)]^2$ . Kuten lokaalisen mallin tapauksessa, suhde  $SSE_E/SSE_R$  esitetään taulukossa 6.2. Luvut voidaan jälleen tulkita uuden menetelmän tehokkuudeksi verrattuna yksinkertaiseen tasoitukseen. Taulukossa on myös yksinkertaisen tasoituksen ja järjestyslukuihin perustuvan tasoituksen tasoitusparametrien arvot.

Taulukosta nähdään, että järjestyslukuihin perustuva menetelmä on huomattavasti yksinkertaista tasoitusta tehottomampi, kun parametrin  $\alpha_1$  arvo on pieni ja parametrin  $\beta_1$  arvo on suuri, tai kun molempien parametrien



Taulukko 6.2: Uuden tasoituksen tehokkuus. Aikasarjan pituus  $N=200$ .

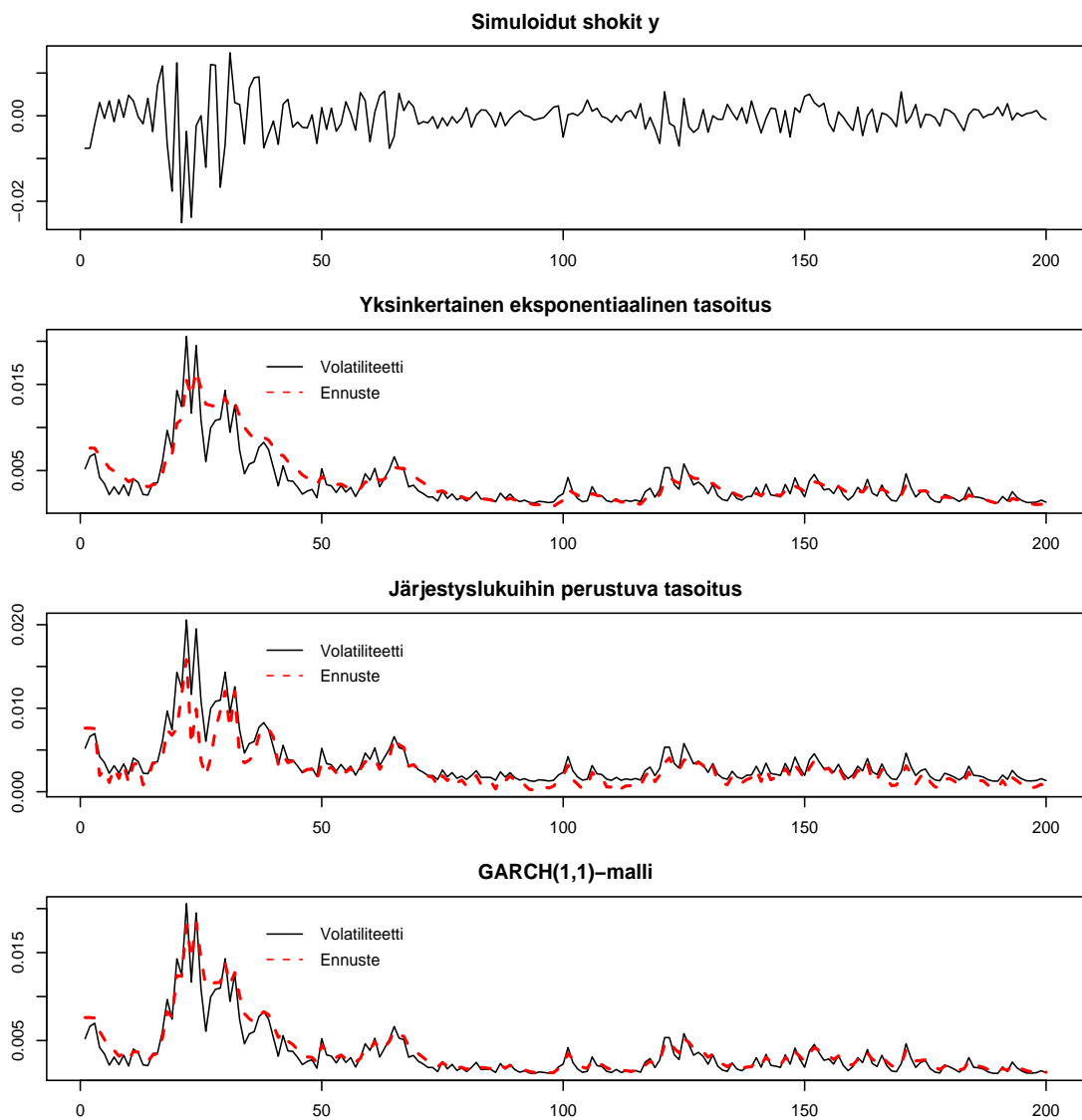
$\alpha_1$	$\beta_1$	Yks. $\alpha$	Järj. $\alpha$	SSE-suhde
0.1	0.8	0.072	0.314	0.290
0.2	0.2	0.234	0.296	0.482
0.3	0.6	0.227	0.645	0.312
0.6	0.3	0.557	0.539	0.770
0.8	0.1	0.866	0.604	1.542

arvot ovat pieniä. Kun parametrien arvot ovat  $\alpha_1 = 0.8$  ja  $\beta_1 = 0.1$ , järjestyslukuihin perustuva menetelmä on selvästi yksinkertaista tasoitusta tehokkaampi, virheneliösummien suhde on silloin 1.542. Uusi menetelmä toimii siis paremmin, kun simuloitun sarjan varianssi määrytyy suureksi osaksi aikaisemman havainnon perusteella. On kuitenkin huomattava, että sovitettaessa GARCH(1,1)-mallia empiirisiin aineistoihin, parametri  $\alpha_1$  estimoituu useimmiten pieneksi ja  $\beta_1$  suureksi. Juuri näin käy esimerkiksi kaikkien luvun (7) osakeaikasarjoihin sovitettavien GARCH(1,1)-mallien kohdalla. Tämän simulointikokeen perusteella järjestyslukuihin perustuva menetelmä ei ole tehokas tällaisten sarjojen volatiliteetin ennustamisessa.

Kuvassa 6.2 on esimerkki simuloinneista ja ennusteista, kun GARCH(1,1)-mallin parametrien arvot ovat  $\alpha_1 = 0.6$  ja  $\beta_1 = 0.3$ . Kuvasta nähdään, että järjestyslukuihin perustuva tasoitus palautuu yksinkertaista tasoitusta nopeammin matalalle volatiliteetin tasolle, kun sarjassa on poikkeavan suuria volatiliteetin arvoja. Usein järjestyslukuihin perustuvan tasoituksen ennusteet ovat kuitenkin liian pieniä. Tämän simuloinnin R-koodi on liitteessä A.3.

GARCH(1,1)-mallin antamat ennusteet ovat hyvin lähellä oikeaa volatiliteettia, mikä on ymmärrettävää, koska simuloitut havainnot ovat kyseisestä mallista. Muiden menetelmien tehokkuudet ovat yleensä alle 10% GARCH(1,1)-mallin tehokkuudesta.

Esitettyjen tulosten lisäksi simuloitiin myös GARCH(1,1)-malleja, joissa  $\epsilon_t$  oli t-jakautunut. Näistä simuloinneista saadut tulokset eivät poikkea paljon ylempänä esitetyistä. Myöskään simuloinnit eri t-jakauman vapausasteilla eivät muuta tuloksia mainittavasti.



Kuva 6.2: GARCH(1,1)-mallilla ( $\alpha_1 = 0.6$  ja  $\beta_1 = 0.3$ ) simuloitu aikasarja ja siitä eri menetelmillä saadut volatiliiteetin ennusteet.

## Luku 7

# Osakkeiden tuottojen volatiliteetin ennustaminen

Seuraavaksi järjestyslukuihin perustuvaa menetelmää käytetään volatilitietin ennustamiseen osakkeiden kurssien aikasarjoista. Tässä tutkielmassa keskitytään viikottaisen volatilitietin ennustamiseen yksi viikko eteenpäin. Saatuja ennusteita verrataan estimoituun volatilitiettiin. Menetelmän toimivuutta verrataan myös yksinkertaisen eksponentiaalisen tasoituksen ennusteisiin ja GARCH(1,1)-mallin tuottamiin ennusteisiin. Luvussa kerrotaan aluksi, kuinka osakeaikasarjoista ennustetaan viikottaista volatilitiettiä ja kuinka volatilitietin estimaatit lasketaan. Sitten osakkeista kerrotaan lyhyesti yksitellen ja volatilitietin ennusteet esitetään graafisesti. Lopuksi esitetään tarkemmin eri menetelmien ennustustulokset ja tarkastellaan jäännöksiä. Yhden osakkeen ennusteiden R-koodit ovat liitteessä A.4.

### 7.1 Viikottaisen volatilitietin ennustaminen

Aineistoiksi valitaan kolmen eri osakkeen päätöskurssien aikasarjat. Päätöskurssi tarkoittaa osakkeen kurssia pörssin sulkeutumishetkellä. Tutkittavat osakkeet ovat Patient Safety Tech, The Coca-Cola Company ja Nokia. Kaikki aineistot on saatu Yahoo Finance -sivustolta (<http://finance.yahoo.com>). Tutkielmassa ei siis tehdä laajaa vertailua ennustusmenetelmien välillä käyttämällä useita osakeaikasarjoja. Näiden esimerkkiaineistojen on tarkoitus antaa kuva siitä, minkälaisen aikasarjojen ennustamiseen järjestyslukuihin perustuvaa tasoitusta voi käyttää. Kaikki aikasarjat jaetaan alustusosaan, jota käytetään menetelmien parametrien estimointiin, ja testiosaan, jolle ennusteet tehdään. Tämä voi joissakin tilanteissa tuottaa huonoja ennusteita, jos alustus- ja testiosat eroavat paljon toisistaan. Samalla voidaan kuitenkin

tarkkailla, kuinka sovitettu malli toimii, jos testiosassa on joitakin poikkeavan suuria tai pieniä arvoja.

Osakkeiden päätöskursseista lasketaan ensimmäisen kertaluvun erotukset ja näistä erotuksista lasketaan luonnolliset logaritmit. Näin saadaan osakkeen päivittäinen logaritminen tuotto, josta käytetään nyt merkintää  $u_{tj}$ . Tässä  $t$  tarkoittaa  $t$ . viikkoa ja  $j$  tämän viikon tiettyä päivää. Näistä saadaan viikottainen logaritminen tuotto, josta käytetään merkintää  $u_t$ , laskemalla viikon päivittäiset logaritmiset tuotot yhteen

$$u_t = \sum_{j=1}^5 u_{tj}.$$

Kun viikottaisista tuotoista  $u_t, t = 1, \dots, T$ , vähennetään niiden keskiarvo, saadaan hinnan viikottainen shokki  $y_t = u_t - \frac{1}{T} \sum_{t=1}^T u_t$ . Tulokset eivät tässä tutkielmassa käsiteltyjen osakkeiden kohdalla muutu paljon, vaikka keskiarvoa ei vähennettäisikään, koska osakkeiden viikottaisten tuottojen keskiarvo on lähellä nollaa. Viikottaisten tuottojen ja viikottaisten shokkien arvot ovat siis lähes samoja. Shokkien arvoja käytetään viikottaisen volatiliiteetin ennustamiseen kappaleessa (5.1) esitetyillä menetelmillä.

Tasoisparametrin estimointi yksinkertaiselle tasoitukselle ja järjestyslukuihin perustuvalla tasoitukselle tehdään viikottaisia arvoja käyttäen. Aikasarjat jaetaan alustusosaan,  $t = 1, \dots, t_0$ , ja testiosaan,  $t = t_0 + 1, \dots, T$ . Alustusosassa on aikasarjan pituudesta riippuen eri määrä havaintoja. Yksinkertaisen tasoituksen tasoisparametri saadaan minimoimalla alustusosan virheneliösumma  $\sum_{i=2}^{t_0} [y_i^2 - \hat{\sigma}_{t-1,E}^2(1)]^2$  ja järjestyslukuihin perustuvan tasoituksen parametri minimoimalla  $\sum_{i=2}^{t_0} [y_i^2 - \hat{\sigma}_{t-1,R}^2(1)]^2$ . Tässä  $y_t^2$  ajatellaan havaitsemattoman varianssin estimaatiksi ja esimerkiksi RiskMetricsissä käytetään tätä tapaa (Taylor 2004, 277). Minimoinnit tehdään käyttämällä Rohjelman optimize-funktiota. Alustusosan viikottaisten shokkien arvoihin sovitetaan myös GARCH(1,1)-malli, jonka parametrit saadaan käyttämällä Rohjelmiston fGarch-paketin garchFit-funktiota ja nlnmb-funktiota uskottavuusfunktion maksimointiin. Tämän jälkeen menetelmiä käytetään aikasarjojen testiosissa volatiliiteetin ennustamiseen yksi viikko eteenpäin. Menetelmissä tarvittava alkuarvo asetetaan testiosan ensimmäisen arvon suuruiseksi.

Volatiliiteetin ennusteita verrataan estimoituun viikottaiseen volatiliiteettiin. Viikottaiselle volatiliiteetille voidaan laskea estimaatti  $\sigma_{t,Est}$  päivittäisistä havainnoista kaavalla

$$\sigma_{t,Est} = \sqrt{\sum_{j=1}^5 y_{tj}^2}, \quad (7.1)$$

missä  $y_{tj}$  on viikon  $t$  ja päivän  $j$  shokki (Taylor 2004, 277). Päivittäinen shokki lasketaan kaavalla  $y_{tj} = u_{tj} - \frac{1}{5T} \sum_{t=1}^T \sum_{j=1}^5 u_{tj}$ . Seuraavissa, eri osakkeita käsittelevissä kappaleissa esitettyjen ennusteiden virheneliösummien laskennassa käytetään volatilitietin estimaattia  $\sigma_{t,Est}$ . Nämä volatilitietin estimaatit ovat myös kuvissa 7.1, 7.2 ja 7.3. Virheneliösummat yksinkertaiselle tasoitukselle, järjestyslukuihin perustuvalla tasoitukselle ja GARCH(1,1)-mallille ovat muotoa  $\sum_{t=t_0+1}^T [\sigma_{t,Est} - \hat{\sigma}_{t-1,E}(1)]^2$ ,  $\sum_{t=t_0+1}^T [\sigma_{t,Est} - \hat{\sigma}_{t-1,R}(1)]^2$  ja  $\sum_{t=t_0+1}^T [\sigma_{t,Est} - \hat{\sigma}_t]^2$ . Tasoitusparametrien arvot voitaisiin estimoida myös käyttämällä volatilitietin estimaattia ja kaavoja  $\sum_{t=2}^{t_0} [\sigma_{t,Est} - \hat{\sigma}_{t-1,E}(1)]^2$  ja  $\sum_{t=2}^{t_0} [\sigma_{t,Est} - \hat{\sigma}_{t-1,R}(1)]^2$ . Parametrit eivät eroa paljon ylempänä esitetystä estimointitavasta, joten näin saatavia tuloksia ei esitetä tarkemmin.

Tuntemattoman varianssin estimaattina voidaan myös käyttää viikottaisen shokkien neliöitä (Taylor 2004, 284), jolloin tämän neliöjuuri on viikottaisen volatilitietin estimaatti. Ennusteiden virheneliösummat voidaan laskea myös käyttäen tätä estimaattia.

Molemmilla tavoilla saatavia virheneliösummia käsitellään tarkemmin kappaleessa (7.3).

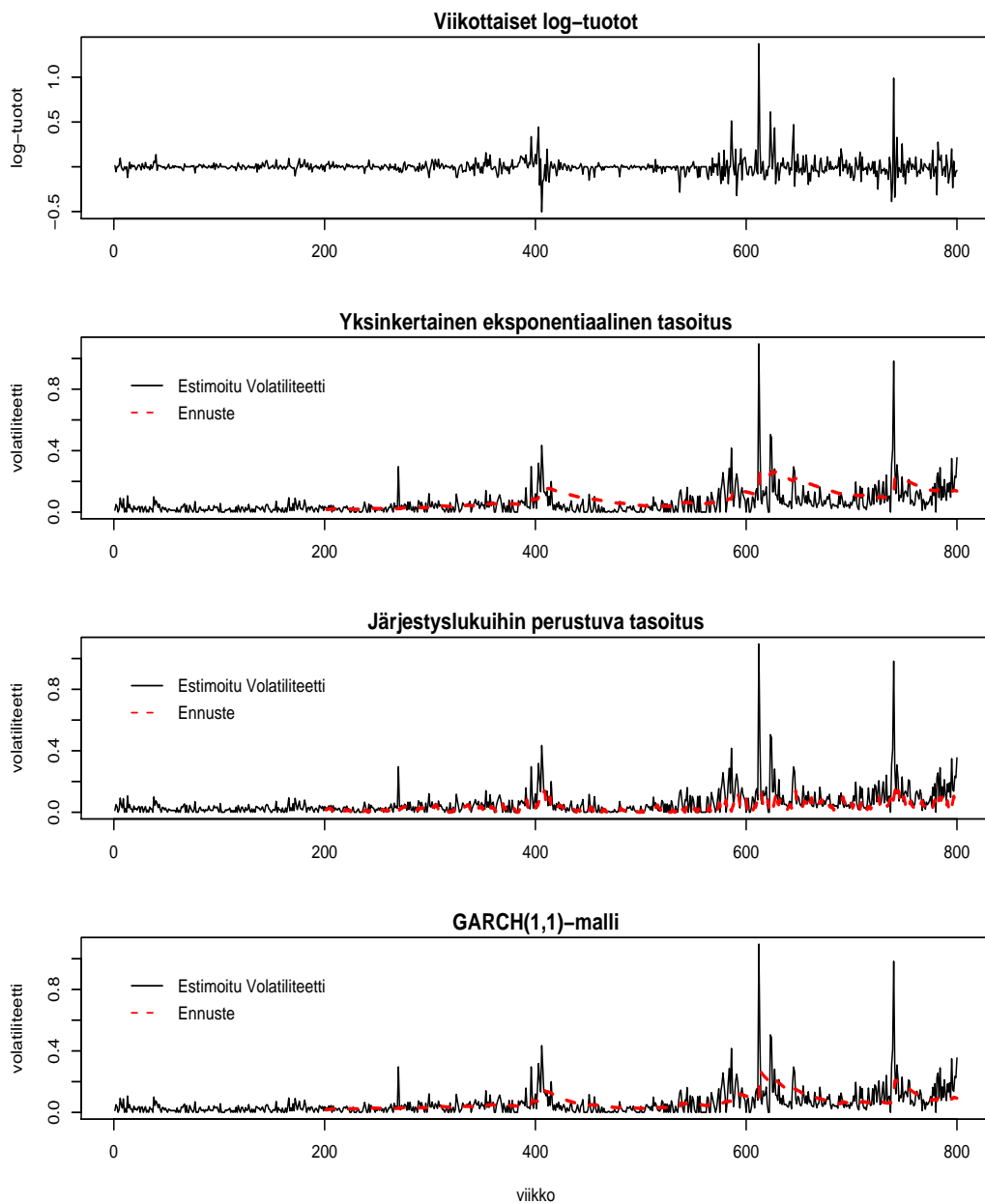
## 7.2 Ennusteet

### 7.2.1 Patient Safety Tech

Yksi valituista osakkeista on Patient Safety Tech. Osakkeen arvoja käytetään väliltä 23.3.1992–5.2.2008. Tällä aikavälillä on 4000 päivittäistä arvoa, joista saadaan 800 arvoa viikottaiselle logaritmiselle tuotolle. Alustusosaksi valitaan 200 ensimmäistä havaintoa, joten testiosan pituus on 600 havaintoa.

Yksinkertaisen tasoituksen ennusteiden virheneliösummaksi saadaan nyt 5.385 ja tasoitusparametrin arvoksi 0.0343. Järjestyslukuihin perustuvan tasoituksen virheneliösumma on 5.321 ja tasoitusparametrin arvo 0.461 ja GARCH(1,1)-mallin virheneliösumma on 4.530 ja parametrien arvot ovat  $\alpha_0 = 3.002 \times 10^{-5}$ ,  $\alpha_1 = 0.0343$  ja  $\beta_1 = 0.930$ . Uuden menetelmän virheneliösumma on siis hieman yksinkertaisen tasoituksen virheneliösummaa pienempi. GARCH(1,1)-mallin virheneliösumma on selvästi pienin.

Kuvassa 7.1 ovat eri menetelmien ennusteet. Siitä nähdään, kuinka järjestyslukuihin perustuvan menetelmän ennusteet palautuvat poikkeavan suurten volatilitietin arvojen jälkeen nopeasti matalalle volatilitietin tasolle. Jos taas volatilitietin pysyy korkealla pidempään, uuden menetelmän ennusteet saavat liian pieniä arvoja. Yksinkertaisen tasoituksen ennusteet taas pysyvät liian korkealla tasolla joidenkin suurten volatilitietin arvojen jälkeen. GARCH(1,1)-mallin ennusteet ovat tehokkaimmat.



Kuva 7.1: Patient Safety Tech. Kuvassa on viikottaiset log-tuotot ja estimoitu volatiliteetti sekä sen ennusteet.

## 7.2.2 The Coca-Cola Company

Seuraavaksi ennustetaan The Coca-Cola Company -osakkeen volatilitteettia. Tämä sarja on edeltävää pidempi, se sisältää osakkeen arvot aikaväliltä 29.12.1975–10.9.2007. Tällä välillä on 8000 päivittäistä havaintoa, joista saadaan 1600 arvoa viikottaiselle logaritmiselle tuotolle. Koska sarja on pidempi kuin ylempänä käsitelty Patient Safety Tech -aikasarja, alustusosaksi valitaan nyt 400 ensimmäistä havaintoa ja testiosan pituus on 1200 havaintoa.

Nyt yksinkertaisen tasoituksen ennusteiden virheneliösumma on 0.370 ja tasoitusparametrin arvo 0.0318. Järjestyslukuihin perustuvan tasoituksen virheneliösummaksi saadaan 0.514 ja tasoitusparametrin arvoksi 0.239, kun taas GARCH(1,1)-mallille virheneliösumma on 0.389 ja parametrien arvot ovat  $\alpha_0 = 5.136 \times 10^{-6}$ ,  $\alpha_1 = 0.0222$  ja  $\beta_1 = 0.975$ . Järjestyslukuihin perustuva tasoitus on siten selvästi tehottomin ennustusmenetelmä. Yksinkertaisen tasoituksen ja GARCH(1,1)-mallin virheneliösummat ovat lähellä toisiaan.

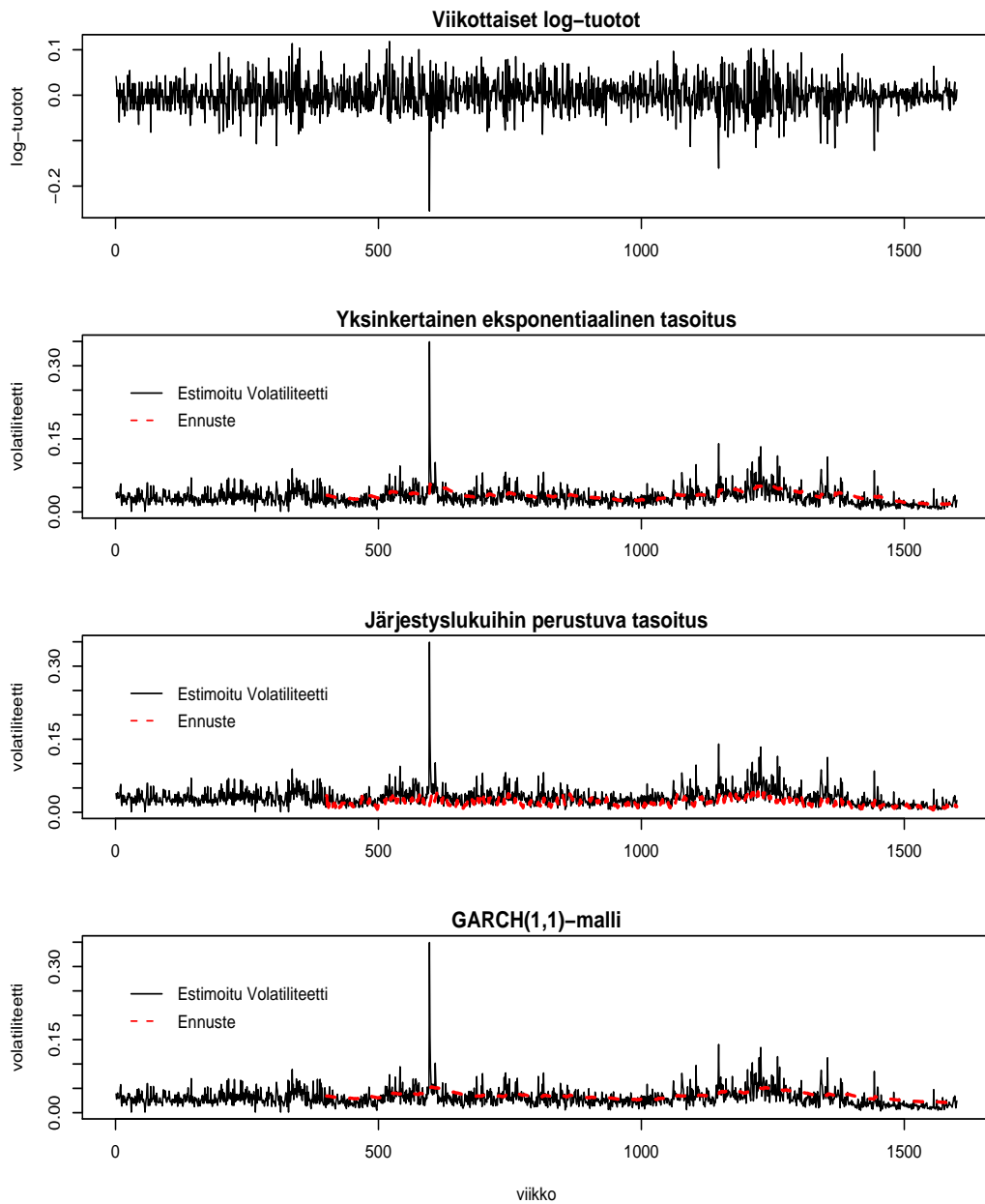
Kuvassa 7.2 esitetään viikottaiset logaritmiset tuotot ja eri menetelmien tuottamat ennusteet. Kuvasta nähdään, että järjestyslukuihin perustuvan menetelmän ennusteet ovat nyt systemaattisesti liian pieniä kaavalla (7.1) laskettuihin volatilitteetin estimaatteihin verrattuna. Yksinkertaisen tasoituksen ja GARCH(1,1)-mallin ennusteet ovat melko samanlaisia ja pysyvät lähellä estimoitua volatilitteettia.

## 7.2.3 Nokia

Lopuksi volatilitteettia ennustetaan Nokian osakeaikasarjasta. Tässä sarjassa on osakkeen arvot aikaväliltä 31.8.1999–15.8.2007. Sarjassa on 2000 päivittäistä havaintoa, joista saadaan 400 arvoa viikottaiselle logaritmiselle tuotolle. Nyt alustusosaksi valitaan ensimmäiset 200 havaintoa ja testiosan pituus on 200 havaintoa.

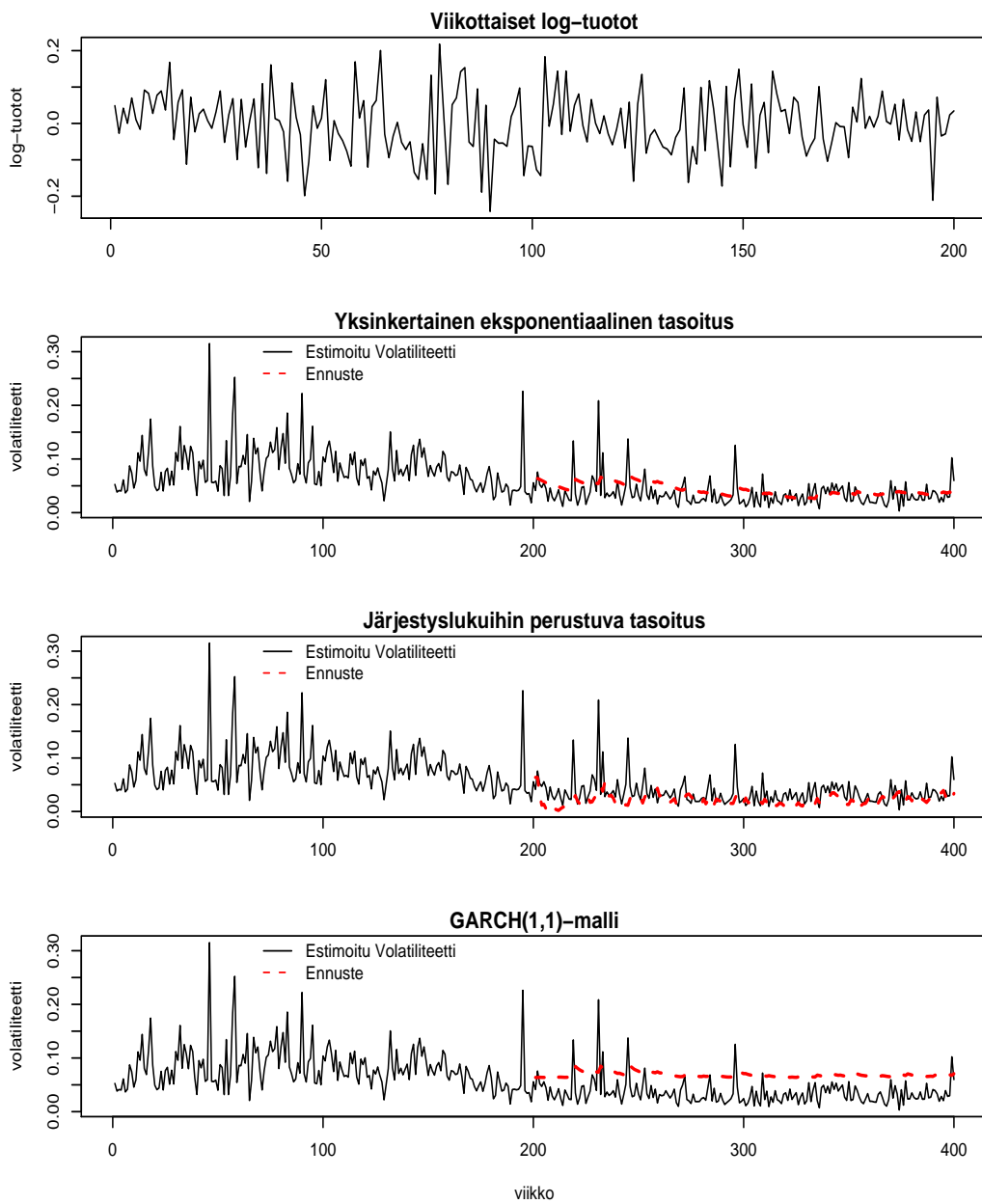
Tälle aineistolle yksinkertaisen tasoituksen virheneliösumma on 0.127 ja tasoitusparametrin arvo 0.0693. Järjestyslukuihin perustuvan tasoituksen virheneliösumma taas on 0.154 ja tasoitusparametrin arvo 0.294. Lopuksi GARCH(1,1)-mallin virheneliösummaksi saadaan 0.336 ja parametrien arvoiksi  $\alpha_0 = 0.000694$ ,  $\alpha_1 = 0.0875$  ja  $\beta_1 = 0.822$ .

Kuvassa 7.3 esitetään viikottaiset logaritmiset tuotot ja eri menetelmien tuottamat ennusteet. Järjestyslukuihin perustuvan tasoituksen ennusteet ovat jälleen hieman liian alhaisella tasolla. Yksinkertaisen tasoituksen ennusteet ovat alussa liian suuria, lopussa ennusteet ovat hyviä. GARCH(1,1)-mallin ennusteet ovat selvästi liian korkealla tasolla.



Kuva 7.2: The Coca-Cola Company. Kuvassa on viikottaiset log-tuotot ja estimoitu volatiliteetti sekä sen ennusteet.





Kuva 7.3: Nokia. Kuvassa on viikottaiset log-tuotot ja estimoitu volatiliteetti sekä sen ennusteet.

## 7.3 Tulokset

Tässä luvussa käydään tarkemmin läpi osakesarjojen volatilitiitin ennustamisessa saatuja tuloksia. Ensin vertaillaan eri menetelmien ennusteiden virheneliösummia ja sen jälkeen tarkastellaan jäännöksiä.

Taulukossa 7.1 esitetään yksinkertaisen tasoituksen ja järjestyslukuihin perustuvan tasoituksen sekä yksinkertaisen tasoituksen ja GARCH(1,1)-mallin virheneliösummien suhde samaan tapaan kuin aikaisemmin simulointikokeiden yhteydessä. Menetelmiä verrataan siis yksinkertaiseen tasoitukseen, sen virheneliösumma jaetaan toisen menetelmän virheneliösummalla. Taulukossa 7.1 virheneliösummien laskemisessa tarvittava tuntematon volatilitiitti korvataan estimoidulla viikottaisella volatilitiitillä, joka lasketaan kaavalla (7.1). Taulukossa 7.2 taas esitetään virheneliösummien suhteet, kun varianssin estimaattina käytetään viikottaisten shokkien neliötä ja volatilitiitin estimaattina sen neliöjuurta.

Taulukko 7.1: Virheneliösummien suhteet. Volatilitiitin estimaatti laskettu kaavalla (7.1).

Osake	Järjestysluk.	GARCH
Patient	1.012	1.189
Coca-Cola	0.720	0.953
Nokia	0.823	0.377

Näiden virheneliösummien perusteella järjestyslukuihin perustuva tasoitus on yksinkertaista tasoitusta tehokkaampi ennustettaessa Patient Safety Tech -osakkeen tuottojen volatilitiittia. Myös GARCH(1,1)-malli on tämän osakkeen kohdalla yksinkertaista tasoitusta tehokkaampi. Yksinkertainen eksponentiaalinen tasoitus taas on tehokkain The Coca-Cola Company -osakkeen ja Nokian volatilitiitin ennustamisessa. Järjestyslukuihin perustuva tasoitus jää näiden osakkeiden kohdalla yksinkertaisesta tasoituksesta jonkin verran. GARCH(1,1)-mallin huonot ennusteet Nokian volatilitiitille johtuvat alustusosan suuresta volatilitiitistä, jonka takia  $\alpha_0$  estimoituu liian suureksi testiosan matalamman volatilitiitin ennustamiseen.

Taulukko 7.2: Virheneliösummien suhteet. Varianssin estimaattina käytetään viikottaisten shokkien neliöitä ja volatilitietin estimaattina niiden neliöjuurta.

Osake	Järjestysluk.	GARCH
Patient	1.181	1.228
Coca-Cola	1.051	0.965
Nokia	1.124	0.463

Taulukon (7.2) tulokset eroavat huomattavasti taulukon (7.1) tuloksista. Valitulla volatilitietin estimaatilla on siis suuri merkitys ennustusmenetelmiä vertaillessa. Patient Safety Tech -osakkeelle saadaan samansuuntaiset tulokset kuin aiemmin. Järjestyslukuihin perustuva tasoitus on yksinkertaista tasoitusta tehokkaampi ja GARCH(1,1)-malli on edelleen menetelmistä tehokkain. The Coca-Cola Company -osakkeen kohdalla tulokset muuttuvat paljon ja järjestyslukuihin perustuva tasoitus onkin nyt tehokkain. Myös Nokian volatilitietin ennustettaessa uusi menetelmä toimii tällä tavalla mitattuna yksinkertaista tasoitusta paremmin. GARCH(1,1)-malli on edelleen Nokian volatilitietin ennustettaessa selvästi huonoin, koska  $\alpha_0$  estimoituu liian suureksi.

On vaikeaa sanoa, kumpi käytetyistä volatilitietin estimaateista on lähempänä oikeaa, havaitsematonta volatilitietin. Tuloksista voidaan kuitenkin sanoa, että uusi menetelmä toimii yksinkertaiseen tasoitukseen verrattuna hyvin, jos viikottaisista shokeista halutaan ennustaa tulevia viikottaisten shokkien neliöitä tai niiden neliöjuurta.

### 7.3.1 Jäännökset

Jos menetelmät sopivat volatilitietin ennustamiseen esitetyistä aineistoista, pitäisi standardoitujen jäännösten  $\tilde{y}_t = \frac{y_t}{\hat{\sigma}_{t-1}(1)}$  olla valkoista kohinaa. Jäännökset lasketaan käyttäen testiosan havaintoja ja laskettuja ennusteita. Jäännösten neliöiden tai jäännösten järjestyslukujen ei myöskään pitäisi riippua toisistaan. Jäännöksiä ja niiden riippuvuutta voidaan tutkia graafisesti sekä autokorrelaatioiden avulla.

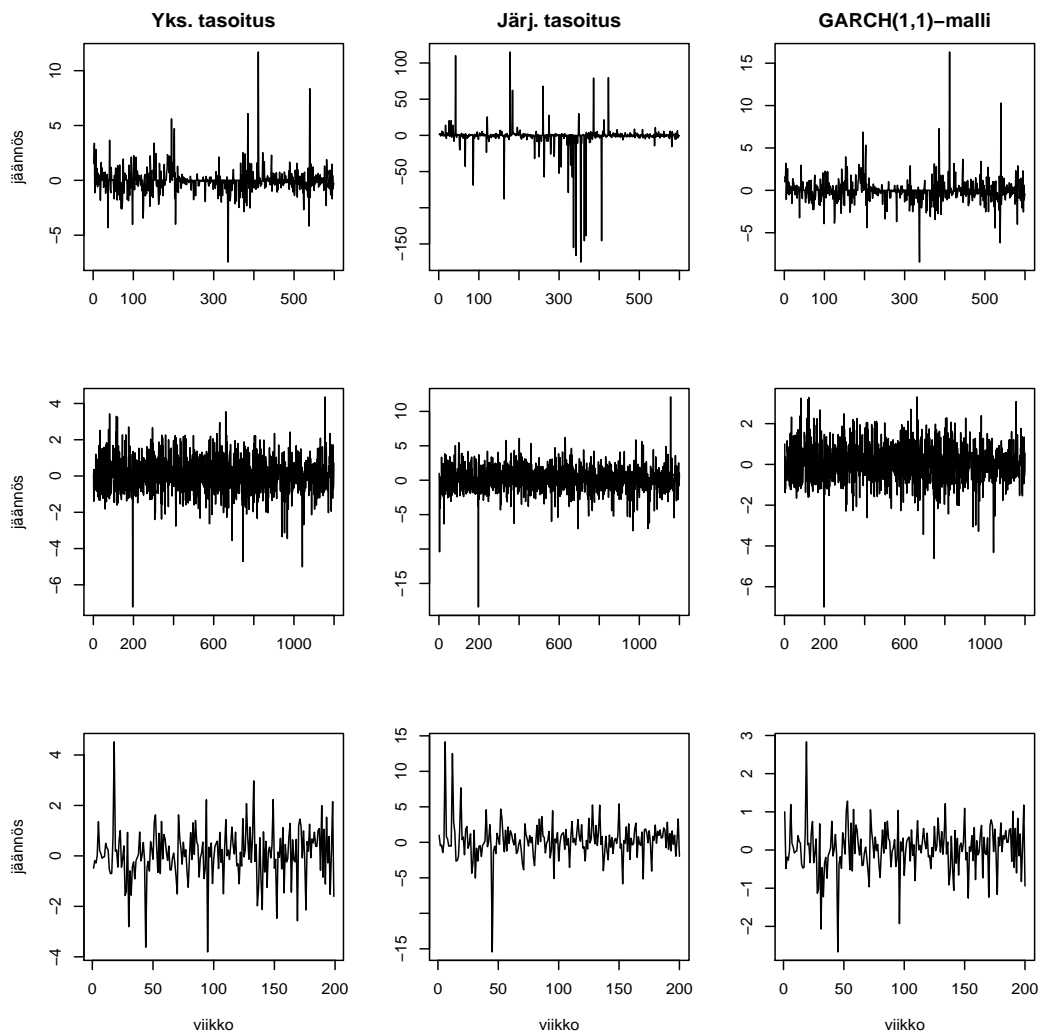
Jäännösten riippumattomuus ei kuitenkaan varsinaisesti kerro ennusteiden tarkkuudesta vaan ennemminkin siitä, voisiko mallia jotenkin parantaa. Eksponentiaalisen tasoituksen menetelmiä käytetään monenlaisiin aikasarjoihin, usein vaikka data ei sosisikaan tilastolliseen malliin, jonka ennustamiseen menetelmä on optimaalinen (Makridakis, Wheelwright & Hyndman 1998, 177).

Jäännökset esitetään kuvassa 7.4 ja jäännösten autokorrelaatiot kuvas-

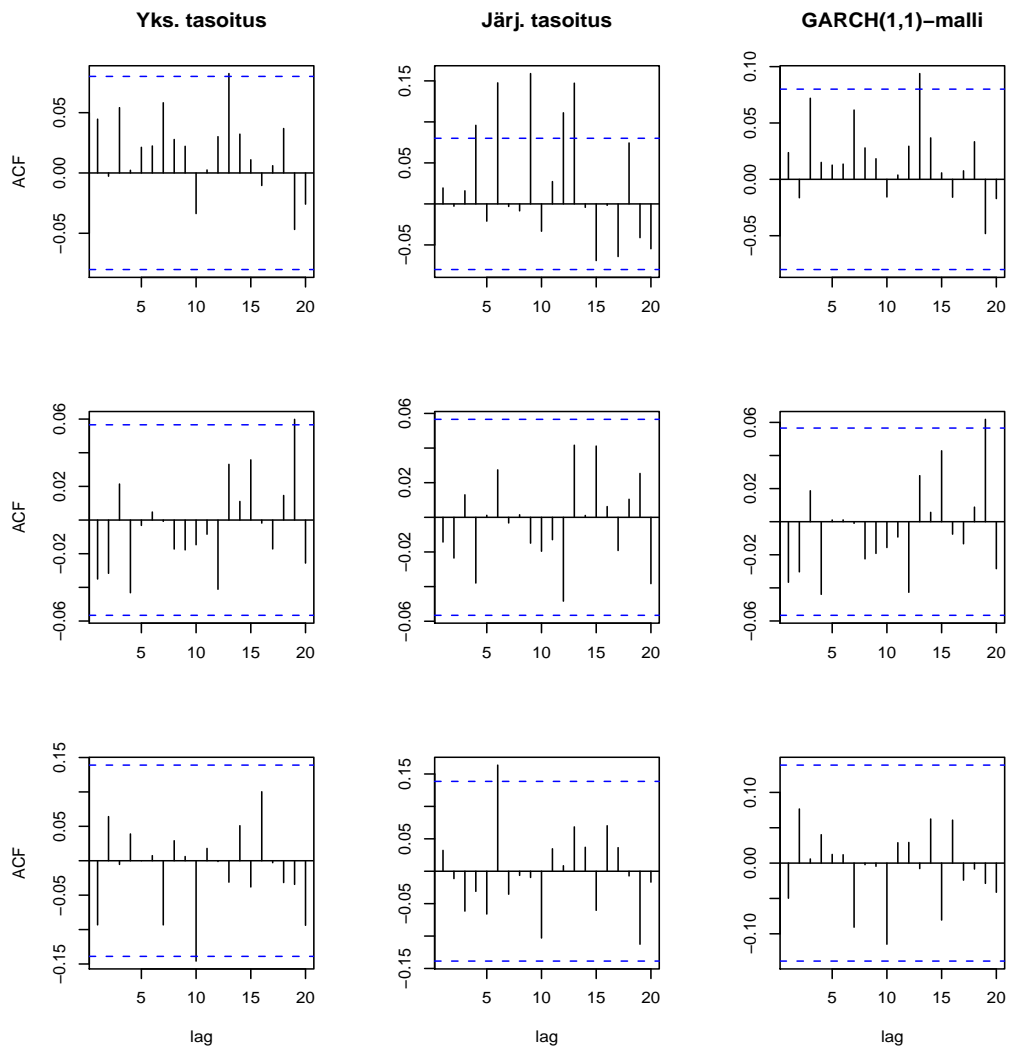
sa 7.5. Jäännösten neliöiden tai jäännösten järjestyslukujen autokorrelaatiota ei esitetä graafisesti. Kuvasta 7.4 nähdään, että varsinkin järjestyslukuihin perustuvan tasoituksen jäännösten välillä näkyy riippuvuutta. Yksinkertaisen tasoituksen ja GARCH(1,1)-mallin jäännöksissä on lievästi merkitsevää autokorrelaatiota viiveellä 13. Järjestyslukuihin perustuvan tasoituksen jäännösten välillä autokorrelaatio on muihin menetelmiin verrattuna voimakkaampaa. Myös kaikkien menetelmien jäännösten neliöiden ja jäännösten järjestyslukujen välillä on autokorrelaatiota. Jäännösten autokorrelaatio johtuu suurimmaksi osaksi siitä, että alustusosa eroaa volatilitietin vaihteluiltaan testiosasta, joten menetelmien parametrit eivät ole optimaalisia testiosan enustamista varten.

The Coca-Cola Company -osakkeelle jokaisella menetelmällä saadut jäännökset ovat lähellä valkoista kohinaa. Jäännösten tai niiden neliöiden autokorrelaatioissa ei näy merkkiä riippuvuudesta. Myöskään jäännösten järjestyslukujen välillä ei ole autokorrelaatiota.

Nokian osakkeelle järjestyslukuihin perustuvan tasoituksen jäännöksissä on pientä autokorrelaatiota viiveellä 6. Muiden jäännösten tai jäännösten järjestyslukujen välillä ei ole autokorrelaatiota. Jäännösten neliöiden välillä on jonkin verran autokorrelaatiota järjestyslukuihin perustuvalla menetelmällä viiveellä 6 ja GARCH(1,1)-mallilla viiveillä 12 ja 14.



Kuva 7.4: Eri menetelmien jäännökset. Ylimpänä Patient Safety Tech, keskellä The Coca-Cola Company ja alimpana Nokia.



Kuva 7.5: Eri menetelmien jäännösten autokorrelaatiot. Ylimpänä Patient Safety Tech, keskellä The Coca-Cola Company ja alimpana Nokia.

## Luku 8

# Yhteenvedo ja johtopäätökset

Lopuksi kerrotaan, miten järjestyslukuihin perustuva eksponentiaalinen tasoitus toimii eri tilanteissa, milloin menetelmää mahdollisesti tulisi tämän tutkielman tulosten perusteella käyttää, ja mitä tulisi tutkia vielä lisää.

Lokaalisen tason mallia simuloitaessa prosessiin asetettiin eri määriä poikkeavia havaintoja ja tästä prosessista ennustettiin latenttia prosessia, jossa poikkeavia havaintoja ei ollut. Tämä voi esimerkiksi tarkoittaa tilannetta, jossa datassa on häiriöitä mittausvirheiden takia ja ennustusmenetelmän tulisi toimia niistä huolimatta mahdollisimman hyvin. Tällaisia sovellusmahdollisuuksia on useita, ja simulointikokeen perusteella järjestyslukuihin perustuva tasoitus on näissä tilanteissa hyvä vaihtoehto yksinkertaiselle eksponentiaaliselle tasoitukselle.

GARCH(1,1)-mallin mukaisten simulointien volatiliteetin ennustamisessa järjestyslukuihin perustuva tasoitus on yksinkertaista tasoitusta tehottomampi, kun parametrin  $\alpha_1$  arvo on pieni ja parametrin  $\beta_1$  arvo suuri. Monet empiiriset osakeaikaasarjat ovat tällaisia. Ennusteet ovat usein systemaattisesti liian alhaisia. Tämä johtuu siitä, millä tavalla GARCH-mallin varianssi muodostuu aiemmista varianssin arvoista ja millä tavalla järjestyslukuihin perustuvan tasoituksen varianssin ennuste lasketaan kaavalla (5.3). Kokeen tulos ei kuitenkaan välttämättä tarkoita, että uusi menetelmä ei soveltuisi volatiliteetin ennustamiseen empiirisistä aikasarjoista, koska niissä volatiliteetti ei luonnollisesti välttämättä käyttäydy GARCH(1,1)-mallin mukaisesti.

Tässä tutkielmassa osakeaikaasarjoista ennustettiin viikottaista volatiliteettia. Volatiliteetin estimaattina käytettiin kahta eri tavoilla laskettua estimaattia, jotka esitettiin kappaleessa (7.1). Järjestyslukuihin perustuva tasoitus suoriutuu volatiliteetin ennustamisesta hyvin silloin, kun ennusteita verrataan viikottaisten shokkien neliöihin ja niiden neliöjuureen. Kolmen tässä tutkielmassa esitetyn osakkeen lisäksi menetelmiä sovellettiin useisiin muihin osakkeisiin ja saadut tulokset ovat samanlaisia. Kun volatiliteetin estimaat-

ti lasketaan kaavalla (7.1), järjestyslukuihin perustuva menetelmä on usein yksinkertaista tasoitusta ja GARCH(1,1)-mallia tehottomampi. Kun varianssin estimaattina käytetään viikottaisia shokkeja ja volatiliteetin estimaattina näiden neliöjuuria, uusi menetelmä toimii jonkin verran paremmin. Varsinkaan järjestyslukuihin perustuvan tasoituksen ennusteiden jäännökset eivät aina olleet valkoista kohinaa.

Tutkielmassa ennustettiin viikottaista volatiliteettia mutta päivittäisen volatiliteetin ennustaminen tapahtuu aivan samoin kuin viikottaisenkin, ennusteet vain tehdään viikottaisten shokkien sijaan päivittäisistä shokeista. Menetelmillä ennustettiin myös päivittäistä volatiliteettia ja tulokset olivat hyvin samansuuntaisia kuin esitetyt tulokset viikottaisen volatiliteetin ennustamisesta. Päivittäiselle volatiliteetille ei voitu laskea volatiliteetin estimaattia samaan tapaan kuin viikottaiselle, joten ennusteita verrattiin vain päivittäisten shokkien itseisarvoon.

Yhteenvetona voidaan sanoa, että järjestyslukuihin perustuva tasoitus toimii tämän pienen otoksen perusteella kohtalaisesti volatiliteetin ennustamisessa. Tulevaisuudessa vertailuja voitaisiin tehdä käyttämällä isompaa aineistoa. Jos uusi menetelmä toimii hyvin yksinkertaiseen tasoitukseen ja GARCH(1,1)-malliin verrattuna, sitä voitaisiin verrata myös kehittyneempiin volatiliteetin ennustusmenetelmiin. Menetelmä voi toimia hyvin myös muissa tilanteissa, varsinkin jos ennustusmenetelmältä vaaditaan robustisuutta.



# Lähteet

- [1] Brown, R. G. (1959). *Statistical Forecasting for Inventory Control*. New York: McGraw-Hill.
- [2] Brown, R. G. (1963). *Smoothing, Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs: Prentice-Hall.
- [3] Chatfield, C. (2004). *The Analysis of Time Series - An Introduction* (6th edition). Boca Raton: CRC Press.
- [4] Chatfield, C., Koehler, A. B., Ord, J. K. & Snyder, R. D. (2001). A new look at models for exponential smoothing. *The Statistician*, 50, 147-159.
- [5] Harvey, A. C. (1993). *Time Series Models*. London: Harvester Wheatsheaf.
- [6] Hettmansperger, T. P. & McKean, J. W. (1998). *Robust Nonparametric Statistical Methods*. New York: Wiley.
- [7] Holt, C. C. (1957). Forecasting seasonals and trends by exponentially weighted moving averages. *ONR Memorandum*, 52. Carnegie Institute of Technology, Pittsburgh.
- [8] Makridakis, S. & Hibon, M. (2000). The M3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16, 451-476.
- [9] Makridakis, S., Wheelwright, S. C. & Hyndman, R. J. (1998). *Forecasting Methods and Applications* (3rd edition). New York: Wiley.
- [10] Nyblom, J. Exponential smoothing based on ranks. ICORS2005, International Conference on Robust Statistics. Jyväskylä 12.–17. June 2005. Abstracts, 60-61.

- [11] R Development Core Team. (2009). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. <http://www.R-project.org>.
- [12] Taylor, J. W. (2004). Volatility forecasting with smooth transition exponential smoothing. *International Journal of Forecasting*, 20, 273– 286.
- [13] Tsay, R. S. (2005). *Analysis of Financial Time Series* (2nd edition). New York: Wiley.
- [14] Vuorenmaa, T. A. (2008). Elements of volatility at high frequency. Ph.D. Thesis. University of Helsinki, Department of Economics.
- [15] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6, 324-342.
- [16] Yahoo Finance. <http://finance.yahoo.com/>

# Liite A

## R-koodit

### A.1 Järjestyslukuihin perustuva eksponentiaalinen tasoitus

```
function(alpha,x){
### Järjestyslukuihin perustuva eksponentiaalinen tasoitus
### Funktion argumentit
# alpha = Tasoitusparametri
# x = Luokan ts objekti
### Funktion palauttavat arvot
# fitted = Tasoitettu aikasarja
# midrank = Järjestysluvut
# predrank = Ennustetut järjestysluvut
# SSE = Ennusteiden virheneliösumma
# x = Alkuperäinen aikasarja

# ennustetut järjestysluvut
predrank<-NULL
predrank[1]<-1
predrank[2]<-1
# tasoitettu aikasarja
fitted<-NULL
fitted[1]<-x[1]
fitted[2]<-x[1]
# järjestysluvut
midrank<-NULL
midrank[1]<-1

T<-length(x)-2

for(i in 1:T){
y<-x[1:i]
# viimeisimmän havainnon järjestysluku
```

```

midrank[i+1]<-1+length(y[y<x[i+1]])+0.5*length(y[y==x[i+1]])
# seuraavan havainnon ennustettu järjestysluku
predrank[i+2]<-alpha*midrank[i+1]+(1-alpha)*predrank[i+1]

k<-floor(predrank[i+2])
lambda<-predrank[i+2]-k

# järjestetty sarja
j<-i+1
y<-x[1:j]
y<-sort(y)
# järjestysluvun ennusteesta lasketaan itse havainnon ennuste
fitted[i+2]<-(1-lambda)*y[k]+lambda*y[k+1]
}

SSE<-sum((x-fitted)^2)
z<-list(x=x, fitted=fitted, predrank=predrank,
midrank=midrank, SSE=SSE)
# etsitään alpha joka minimoi SSE:n käyttämällä R:n
# optimize funktiota, poistetaan sitten viimeinen rivi (SSE)
# ja tasoitetaan sarja x uudelleen käyttämällä saatua alphaa,
# jolloin saadaan z.
SSE
}

```

## A.2 Lokaalisen tason mallin simulointi ja ennustaminen

```
# Simuloidaan lokaalisen tason mallia, johon asetetaan
# poikkeavia havaintoja. Ennustetaan sarjaa yksinkertaisella
# tasoituksella ja järjestyslukuihin perustuvalla tasoituksella.
# Esimerkkikoodissa q=1 ja poikkeavien havaintojen lukumäärä 5,
# kaikki muut simuloinnit on tehty vastaavalla tavalla.

mu<-NULL
mu[1]<-rnorm(1, mean=0, sd=sqrt(1))
for (i in 2:100){
mu[i]<-mu[i-1]+rnorm(1, mean=0, sd=sqrt(1))
}
mu<-ts(mu)
e<-arima.sim(n=100, list(), sd=sqrt(1))
e<-e[2:101]
e<-ts(e)
e.outlier<-NULL
for (i in 1:5){
e.outlier[i]<-sample(c(-1,1), size=1)*20*sqrt(1)
}
list<-sample(1:100, size=5)
e<-replace(e, list, e.outlier[1:5])
y<-mu+e

# Ennusteet tehdään sarjasta y, tässä RobustifunktioSSE.R on
# ylempänä esitetty järjestyslukuihin perustuvan tasoituksen
# funktio, joka palauttaa SSE:n. Robustifunktio.R taas on
# funktio, josta viimeinen rivi (SSE) on poistettu.
robustiSSE<-source("RobustifunktioSSE.R")$value
xmin <- optimize(robustiSSE, c(0, 1), tol = 0.01, x=y)
robusti<-source("Robustifunktio.R")$value
rankennusteet<-robusti(xmin$minimum,y)

# yksinkertaisen tasoituksen ennusteet
HWennusteet<-HoltWinters(y ,alpha = NULL, beta = 0, gamma = 0)

# virheneliösummat, ennusteita verrataan sarjaan mu
sum((mu-rankennusteet$fitted)^2)
sum((mu-HWennusteet$fitted[,1])^2)

# tutkielmassa esitetty kuva
par(new=T, mfcol=c(3,1), mar=c(2.5,2.5,2.5,2.5))

ts.plot(y,col=c(1,2), lty=c(1,2), xlab="", ylab="", main="Aikasarja y")

ts.plot(mu, HWennusteet$fitted[,1], col=c(1,2), lty=c(1,2), xlab="",
```

```
main="Yksinkertaisen eksponentiaalisen tasoituksen ennusteet")
#kuvateksti
legend(bty="n", x=5,y=9, lty=c(1,2), col=c(1,2),
legend=c("Aikasarja mu", "Ennuste"))

ts.plot(mu, rankennusteet$fitted, col=c(1,2), lty=c(1,2), xlab="",
main="Järjestyslukuihin perustuvan tasoituksen ennusteet")
#kuvateksti
legend(bty="n", x=5,y=9, lty=c(1,2), col=c(1,2),
legend=c("Aikasarja mu", "Ennuste"))
```

## A.3 GARCH(1,1)-mallin simulointi ja volatili- teetin ennustaminen

```
# Simuloidaan GARCH(1,1)-mallia ja ennustetaan siitä
# volatilititeettia. Esimerkissä alpha1=0.6 ja beta1=0.3.
# Tätä varten täytyy ladata fGarch-paketti.

# määritellään garch-mallin parametrit
spec <- garchSpec(model = list(alpha=c(0.6), beta=c(0.3)))
sim<-garchSim(spec, extended=TRUE, n=200)
garch<-sim[,1]
# havainnot
garch<-ts(garch[1:200])
# keskihajonta (volatilititeetti)
sigma<-sim[,2]
sigma<-ts(sigma[1:200])

yy<-garch*garch

# yksinkertaisen tasoituksen ennusteet
HW<-HoltWinters(yy ,alpha = NULL, beta = 0, gamma = 0)

#järjestyslukuihin perustuvan tasoituksen ennusteet
robusti<-source("RobustifunktioSSE.R")$value
xmin <- optimize(robusti, c(0, 1), tol = 0.01, x=yy)
robusti<-source("Robustifunktio.R")$value
tas<-robusti(xmin$minimum,yy)

# garch(1,1)-mallin sovitus
garchfit<-garchFit(data=garch, formula~garch(1,1))
# @sigma.t a numeric vector with the conditional standard deviation.
      mu          omega      alpha1      beta1
-8.330085e-05  5.344716e-07  3.775503e-01  6.047430e-01

# garch(1,1)-mallin ennusteet (laskettu omalla koodilla,
# nämä saisi fGarch-paketin funktioillakin).
sigma2<-NULL
sigma2[1]<-yy[1]
sigma2[2]<-yy[1]
for(i in 2:199){
sigma2[i+1]<-5.344716e-07+3.775503e-01*yy[i]+6.047430e-01*sigma2[i]
}
sigma2<-ts(sigma2, start=1)
sigmagarch<-sqrt(sigma2)

# kuva
op<-par(mfcol=c(4,1), mar=c(2.5,2.5,2.5,2.5))
ts.plot(garch, xlab="", main="Simuloidut shokit y", ylab="")
```

```

ts.plot(sigma, sqrt(HW$fitted[,1]), col=c(1,2), lty=c(1,2), lwd=c(1,2)
xlab="", main="Yksinkertainen eksponentiaalinen tasoitus")
# kuvateksti
legend(bty="n", cex = 1, x=35,y=0.020, lty=c(1,2), col=c(1,2),
legend=c("Volatiliteetti","Ennuste"))

ts.plot(sigma, sqrt(tas$fitted), col=c(1,2), lty=c(1,2), lwd=c(1,2),
xlab="", main="Järjestyslukuihin perustuva tasoitus")
# kuvateksti
legend(bty="n", cex = 1, x=35,y=0.020, lty=c(1,2), col=c(1,2),
legend=c("Volatiliteetti","Ennuste"))

ts.plot(sigma, sigmagarch, col=c(1,2), lty=c(1,2), lwd=c(1,2),
xlab="", main="GARCH(1,1)-malli")
# kuvateksti
legend(bty="n", cex = 1, x=35,y=0.020, lty=c(1,2), col=c(1,2),
legend=c("Volatiliteetti","Ennuste"))

# virheneliösummat
sum((sigma[2:200]-sqrt(HW$fitted[,1]))^2)
sum((sigma[2:200]-sqrt(tas$fitted[2:200]))^2)
sum((sigma[2:200]-sigmagarch[2:200])^2)

```



## A.4 Volatiliteetin ennustaminen osakeaikasarjasta

```
# Esimerkkinä The Coca-Cola Company. Muiden osakkeiden
# ennustamiseen tehdyt R-koodit ovat vastaavanlaisia.

data<-read.table("COCACOLA.txt", header=T)
#päivän päätöskurssit
data<-data$AdjClose
data<-data[3501:11501]
logdata<-log(data)
logtuotot<-diff(logdata)

# päivittäiset jäännökset/shokit (residuals of the mean equation)
pshokki<-logtuotot-mean(logtuotot)
pshokki<-ts(pshokki)
plot(pshokki)
pshokki2<-pshokki*pshokki
plot(pshokki2)

# viikottaisen volatiliteetin estimaatit, joihin
# ennusteita verrataan
estimvolatiliteetti<-NULL
T<-length(pshokki2)
loop<-c(seq(1,T,5))
j<-1
for(i in loop){
  estimvolatiliteetti[j]<-sqrt(pshokki2[i]+pshokki2[i+1]+pshokki2[i+2]
  +pshokki2[i+3]+pshokki2[i+4])
  j<-j+1
}
estimvolatiliteetti<-ts(estimvolatiliteetti[1:1600])

# viikottaiset logaritmiset tuotot
vlogtuotot<-NULL
T<-length(data)
loop<-c(seq(1,T,5))
j<-1
for(i in loop){
  vlogtuotot[j]<-logtuotot[i]+logtuotot[i+1]+logtuotot[i+2]
  +logtuotot[i+3]+logtuotot[i+4]
  j<-j+1
}
vlogtuotot<-ts(vlogtuotot[1:1600])

# jäännökset/shokit, joista ennusteet tehdään
# (residuals of the mean equation)
y<-vlogtuotot-mean(vlogtuotot)
```

```

yy<-y*y

# Järjestyslukuihin perustuvan tasoituksen ennusteet.
# SSE minimoidaan optimize funktiolla käyttämällä
# ensimmäistä 400 havaintoa (testiosa).
robustiSSE<-source("RobustifunktioSSE.R")$value
alpha <- optimize(robustiSSE, c(0, 1), tol = 0.01, x=yy[1:400])
robusti<-source("Robustifunktio.R")$value
rankennusteet<-robusti(alpha$minimum,yy[401:1600])
rankennusteet<-ts(rankennusteet$fitted, start=401)

# yksinkertaisen tasoituksen ennusteet
HW<-HoltWinters(yy[1:400] ,alpha=NULL, beta=0, gamma=0)
HWennusteet<-HoltWinters(yy[401:1600],alpha=HW$alpha,beta=0, gamma=0)
HWennusteet<-ts(HWennusteet$fitted[,1], start=402)

# GARCH(1,1)-mallin ennusteet, tähän tarvitaan fGARCG-paketti)
garchsovitus<-garchFit(data=y[1:400], formula~garch(1,1))
# ennustetaan sovitetun mallin parametreilla volatilitteettia
garchvarienssi<-NULL
garchvarienssi[1]<-yy[401]
garchvarienssi[2]<-yy[401]
for(i in 2:1199){
garchvarienssi[i+1]<-5.136247e-06+0.02222027*yy[400+i]
+0.9747736*garchvarienssi[i]
}
garchvarienssi<-ts(garchvarienssi, start=401)
garchennusteet<-sqrt(garchvarienssi)

# kuva
op<-par(mfcol=c(4,1), mar=c(2.5,2.5,2.5,2.5))
ts.plot(y[1:1600], ylab="log-tuotot", xlab="",
main="Viikottaiset log-tuotot")

ts.plot(estimvolatilitteetti, sqrt(HWennusteet), ylab"volatilitteetti",
xlab="", col=c(1,2), lty=c(1,2), lwd=c(1,2),
main="Yksinkertainen eksponentiaalinen tasoitus")
# kuvateksti
legend(bty="n", cex = 1, x=0, y=0.3, lty=c(1,2), col=c(1,2),
legend=c("Estimoitu Volatilitteetti","Ennuste"))

ts.plot(estimvolatilitteetti, sqrt(rankennusteet), ylab"volatilitteetti",
xlab="", col=c(1,2), lty=c(1,2), lwd=c(1,2),
main="Järjestyslukuihin perustuva tasoitus")
# kuvateksti
legend(bty="n", cex = 1, x=0, y=0.3, lty=c(1,2), col=c(1,2),
legend=c("Estimoitu Volatilitteetti","Ennuste"))

ts.plot(estimvolatilitteetti, garchennusteet, ylab"volatilitteetti",

```

```

xlab="viikko", col=c(1,2), lty=c(1,2), , lwd=c(1,2),
main="GARCH(1,1)-malli")
# kuvateksti
legend(bty="n", cex = 1, x=0, y=0.3, lty=c(1,2), col=c(1,2),
legend=c("Estimoitu Volatiliteetti", "Ennuste"))

# kaikkien menetelmien virheneliösummat (SSE)
sum((estimvolatiliteetti[402:1600]-sqrt(HWennusteet))^2)
sum((estimvolatiliteetti[401:1600]-sqrt(rankennusteet))^2)
sum((estimvolatiliteetti[401:1600]-garchennusteet)^2)

# jäännösten tarkastelu
errorHW<-y[402:1600]/sqrt(HWennusteet)
errorRANK<-y[401:1600]/sqrt(rankennusteet)
errorGARCH<-y[401:1600]/garchennusteet

errorHW<-ts(errorHW)
errorRANK<-ts(errorRANK)
errorGARCH<-ts(errorGARCH)

# jäännösten kuva
par(mfrow=c(3,3), mar=c(3.9,4,2.5,0.5))
plot(errorHW, ylab="jäännös", xlab="", main="Yks. tasoitus")
plot(errorTAS, ylab="", xlab="", main="Järj. tasoitus")
plot(errorGARCH, ylab="", xlab="", main="GARCH(1,1)-malli")

# jäännösten acf
par(mfrow=c(3,3), mar=c(4,4,3,2))
pacf(errorHW, lag.max=20, xlab="", ylab="ACF", main="")
pacf(errorTAS, lag.max=20, xlab="", ylab="", main="")
pacf(errorGARCH, lag.max=20, xlab="", ylab="", main="")

# jäännösten neliöiden acf
par(new=T, mfcol=c(3,1))
pacf(errorHW*errorHW)
pacf(errorRANK*errorRANK)
pacf(errorGARCH*errorGARCH)

# jäännösten järjestyslukujen acf
rankHW<-ts(rank(errorHW))
rankTAS<-ts(rank(errorTAS))
rankGARCH<-ts(rank(errorGARCH))

par(new=T, mfcol=c(3,1))
pacf(rankHW)
pacf(rankTAS)
pacf(rankGARCH)

```