Copenhagen Business School

Cand.merc.(mat)

GARCH-modeller og forecasting.

The GARCH models and forecasting.

15. januar 2018

Antal normalsider: 66 sider.

Speciale af: Jack Petersen

Vejleder: Anders Rønn-Nielsen

Abstract

This thesis examines the GARCH models. The purpose of the paper is to perform an analysis of different GARCH models, and to see how well they fit our dataset's log-returns. The log-returns come from the closing price of the stock from Danske Bank. Firstly, we want to show that GARCH models are ideal to describe financial datasets, which we did. We used the statistical programming language Rstudio to implement the different GARCH models. Subsequently, we looked at the different distributions the White noise had, and wanted to compare them across different GARCH models. We saw that the t-distribution fitted our dataset best across several models.

Next we had to look how the different models estimated their parameters and how well they fitted the dataset. It was found that low orders of GARCH models were preferred over higher orders, since the Goodness-of-fit generally decrease as the order increases. We found two models, GARCH(1,1) and EGARCH(1,1), which were the best to describe our log-returns, where the EGARCH(1,1) was marginally better than the former. That was also the case when we used the models to forecast. We saw, that EGARCH could more accurately predict the confidence interval in which our datasets log-returns occurred. Although GARCH over a longer period of time was able to get a greater amount of data within the intervals of forecasts, we concluded that EGARCH is the better model for forecasting. We also used simulation to get an idea of how the return could look like inside the intervals of the forecasted models.

Kapitel 1

1.1 Motivation

Finansielle markeder har specielle kendetegn, som man skal være klar over, før man kaster sig ud i at finde den bedst mulige model. Disse kendetegn skal vi kigge nærmere på i denne opgave, samt finde modeller som kan beskrive data fra det finansielle markedet. Yderligere skal vi på baggrund af forskellige modeller finde den bedst mulige model. Det gælder både til estimationen samt til forecasting af vores datasæt.

Det er mere reglen end undtagelsen, at finansielle data har volatilitetets clusters, altså perioder med henholdvis høj og lav volatilitet. Det gør, at det kan være svært at finde modeller til at beskrive finansielle data, og gør det svært at forecaste i forhold til data, hvor variansen er konstant. Vi kan dog bruge GARCH modeller, som tager hensyn til de perioder med henholdvis høj og lav volatilitet. ARCH gør det samme, dog for kortere perioder.

1.2 Problemformulering

Hvordan kan man beskrive finansielle data ved hjælp af GARCH-modeller, og hvordan kan denne model bruges til forecasting?

Jeg ønsker at undersøge fordele og ulemper ved brug af GARCH-modeller til beskrivelse af finansielle data. Derudover ønsker jeg at undersøge forecasting-metoder for GARCH-modellen, og se hvor gode de hver især er.

1.3 Afgrænsning

Da der findes adskillige modifikationer af GARCH modellen, har jeg valgt at holde antallet af modeller nede for ikke at gøre opgavens omfang for stort til denne afhandling. I denne opgave kigges der på ARCH, GARCH, EGARCH og ARMA+GARCH. De valgte modeller bruges på datamaterialet som bliver beskrevet senere. - Det grundlæggende er at finde den bedst mulige model til netop mit datasæt ud fra alt teorien.

1.4 Opgavens struktur

Denne afhandling er opbygget af flere kapitler, som alle sammen skal være med til at besvare ovenstående problemstilling. De forskellige kapitler er som følger.

Kapitel 1 er motivationen bag afhandlingen, men indeholder også problemformuleringen som skal besvares løbende igennem specialet. Indeholder ligeledes afgrænsningen til specialet.

Kapitel 2 indeholder fundemental teori som anvendes gennem hele specialet.

Kapitel 3 omhandler teorien til de forskellige modeller.

Kapitel 4 beskriver vores datasæt.

Kapitel 5 er den indledende dataanalyse af datasættet.

Kapitel 6 indeholder teorien til estimationen og forecasting.

Kapitel 7 er modelanalysen. Vi ser her hvordan de forskellige modeller passer til vores datasæt.

Kapitel 8 indeholder forecastingen.

Kapitel 9 opsummering af de foregående kapitler, og en følgende beskrivelse af hvad man kunne arbejde videre med.

Indholdsfortegnelse

Kapitel 2 – Indledende teori.

2.1 Indledning

I dette kapitel kigger vi på grundlæggende teori, som bliver benyttet indenfor GARCH modeller. Vi kommer her ind på grundlæggende teorier, som blandt andet skal bruges til dataanalyse senere hen.

2.2 Stationæritet

Vi definerer en stationær stokastisk proces udfra Ruppert; ”*Stationary stochastic processes are probability models for time series with time-invariant behavior”*. Her er et andet vigtigt begreb *time-invariant*, som betyder at tidsrækken, i vores tilfælde aktiepris, viser samme opførelse fra en tidsperiode til en anden. Vi starter med at kigge på, hvad det betyder, at en proces er stationær. Det er en fundamental betingelse, som gør at vi kan benytte teorien, som bliver beskrevet senere hen. Det skal dog bemærkes at finansielle data nødvendigvis ikke er stationær, men afkastet er det ofte. Det er klart at afkastet på en aktie kan variere fra en periode til en anden, men så er det vigtigt, at for eksempel middelværdien og standardafvigelsen har en varians som ikke afhænger af tidsperioden. Man kan tjekke for stationæritet ved at se om afkastet, på trods af større og mindre udsving, altid svinger udfra et fast niveau – dette kaldes mean reversion. Det kan ses i figur 3, at vores daglige log-afkast er mean reversion, at på trods af perioder med volatilitetsclusters så lader det til, at den svinger omkring et konstant niveau og altider vender tilbage dertil.

I vores tilfælde vil vi benytte en svagt stationær proces som er givet ved, at

$$E(Y_i) = \mu\,, for\ alle\ i$$

$$Var(Y_i) = \sigma^2, for\ alle\ i$$

$$Corr(Y_i, Y_j) = \rho(|i-j|), for\ alle\ i\ og\ j\ for\ en\ funktion\ \rho(h)$$

Det er værd at bemærke at både $\mu\ og\ \sigma^2$ er konstanter og ikke ændres af tiden. Korrelationen afhænger kun af perioden imellem de to observationer. For en svag stationær proces kræves der altså at middelværdi, standardafvigelsen og at korrelationen er stationære, men intet andet behøver at være det.

2.3 Autocorrelation og Ljung Box

Ovenstående funktion $\rho(h)$ er autokorrelationfunktionen, også forkortet ACF, af en proces. Den måler forholdet mellem en variables nuværende værdi og dens tidligere værdier. Såfremt at vi antager, at Y_t er en stationær proces, kan vi bruge nedenstående formel til at finde sample autokovarians funktionen;

$$\hat{\gamma}(h) = n^{-1} \sum_{j=1}^{n-h} (Y_{j+h} - \bar{Y})(Y_j - \bar{Y})$$

Herefter kan funktionen $\rho(.)$ estimeres ved, at vi benytter sample AFC defineret som

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$$

ACF bruges i vores tilfælde til at tjekke om det daglige log-afkast er korreleret eller ej. Det ses senere i dataanalysen, at Rstudio plotter sample ACF med grænser på. Det bruges til at teste om nulhypotesen om, at autokorelationen er 0 kan forkastes.

En anden måde at teste korrelationen er ved hjælp af Ljung Box. Her tester man hypotesen om nedenstående gør sig gældende. Hvis vi kan forkaste, altså at p-værdien er under 5%, så kan vi konkludere, at mindst en autokorelation er forskellig fra 0.

$$\rho(1) = \rho(2) = \cdots = \rho(K) = 0, \text{ for et valgt K.}$$

2.4 Hvid støj

Hvid støj er meget vigtigt, når vi har med GARCH modeller at gøre, og derfor er det vigtigt at få defineret den hvide støj. Hvid støj er et eksempel på en stationær proces. For at støjen skal være hvid støj, skal variablene være ukorrelerede. Derudover skal variablene som afled for svagt stationære proceser have forventet værdi nul og den samme begrænsede varians. Matematisk ser betingelserne således ud;

$$E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma^2 < \infty, Corr(\varepsilon_1, \varepsilon_2) = 0$$

En måde at opnå hvid støj på er at antage, at variablene er uafhængige og ens fordelte. Når vi senere i opgaven kommer til dataanalysen, så vil vi tillade støjen at have forskellige fordelinger. De tre fordelinger, som vil være relevante, gennemgås hurtigt herunder.

Gennem det meste af teorien antager vi, at den hvide støj er normalfordelt, og hvis det er tilfældet så har den nedenstående tæthedsfunktion.

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y)^2}{2\sigma^2}}$$

Bemærk, at vi i vores analyse også vil tillade støjen at være t-fordelt som har følgende tæthedsfunktion;

$$f(y) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v\pi}\Gamma\left(\frac{v}{2}\right)}\left(1+\frac{y^2}{v}\right)^{-\frac{v+1}{2}}$$

Hvor v er antallet af frihedsgrader, og Γ er gamma funktionen $\Gamma(x) = \int_0^\infty y^{x-1}e^{-y}dy$

Den sidste mulighed er, at vi lader støjen være GED fordelt. GED er en forkortelse af *Generalized error distributions,* og det er en fordeling som har eksponentielle haler. Den standardiseret GED har en form parameter *v* og har tætheden:

$$f_{ged}^{std}(y) = k(v)e^{-1/2\left|\frac{y}{\lambda_v}\right|^v}, -\infty < y < \infty$$

Hvor

$$\lambda_v = \left(\frac{2^{-\frac{2}{v}}\Gamma(v^{-1})}{\Gamma\left(\frac{3}{v}\right)}\right)^{1/2}, k(v) = \frac{v}{\lambda_v 2^{1+1/v}\Gamma(v^{-1})}$$

Hvor $v > 0$ bestemmer halernes vægt, og des mindre v er jo tungere er halerne.

2.5 Kurtosis og skævhed

Både kurtosis og skævheden hjælper med at få et overblik over en sandsynlighedsfordeling. Kurtosis giver et indtryk i, hvor meget af datasættet der er nær midten af data, og samtidigt også hvor tunge halerne er i datasætet. En rimelig antagelse omkring midten af data er området fra $\mu - \sigma$ til $\mu + \sigma$. Venstre hale er går fra $-\infty$ til $\mu - 2\sigma$. Det betyder, at højre hale går fra $\mu + 2\sigma$ til ∞. Vi ved, at finansielle data som regel har tunge haler, og Kurtosis er et nyttigt redskab til at bedømme, hvor tunge halerne er. Kurtosis for Y findes ved hjælp af nedenstående formel. Det skal nævnes, at en normal fordeling har Kurtosis lig 3.

$$Kur = \frac{E\{Y - E(Y)\}^4}{\sigma^4}$$

Så en Kurtosis større end 3 indikerer haler, som er tungere end normalfordelingen, hvor en kurtosis mindre end 3 indikerer, at halerne er lettere. Da vi også skal kigge på t-fordelingen kan det være værd at nævne, hvad Kurtosis er for denne fordeling. Vi ved, at kurtosis er givet ved nedenstående formel såfremt, at v er større end 4. Derudover er Kurtosis uendelig såfrem $2 < v \leq 4$, og ellers er den udefineret.

$$\frac{6}{v - 4}$$

Skævheden siger noget om, hvor symmetrisk datasættet er. Hvis der ikke er nogen skævhed, altså at $Sk = 0$, så er der symmetri. Positiv skævhed betyder, at der er mere til højre, altså at højre hale er tungere end venstre. Negativ skævhed betyder naturligvis det modsatte – at venstre hale er tungere end den højre. Nedenstående formel giver definition på skævheden for Y.

$$Sk = \frac{E\{Y - E(Y)\}^3}{\sigma^3}$$

2.6 Implementation

En afsluttende bemærkning til dette kapitel må være, hvordan al teorien i denne afhandling implementeres. Vi benytter udelukkende Rstudio til at analysere datasættet udfra teorien og modeller som løbende bliver beskrevet. I takt med at datasættet bliver analyseret, og vi skal forsøge at fitte den bedst muligt model, vil jeg forklare alt output, som kan forekomme svært at tolke. Hele koden som er blevet benyttet til dette speciale er at finde under bilaget i Bilag 4 fra side 81.

Kapitel 3 – Modeller.

3.1 Indledning

Dette kapitel indeholder den grundlæggende teori bag GARCH modeller. Heriblandt autoregressive processer, moving average, og naturligvis ARCH og GARCH. Formålet ved dette kapitel er at beskrive, hvorfor det er at GARCH er fordelagtig at benytte til at beskrive finansielle data. Til dette kapitel er der primært blevet anvedt Ruppert og Mateson bog.

3.2 AR

Før vi kan kigge på GARCH modeller, skal vi først kigge på ARCH, men allerførst skal vi kigge på AR processer, som er autoregressive processer. Autoregressive processer er en stokastisk proces, hvor vi antager, at den nutidige værdi afhænger af tidligere værdier. Vi starter med den mest simple autoregressive proces, som er AR(1), hvilket betyder at processen er af første orden. Det betyder at AR(1) udelukkende afhænger af værdien lige før. Som nedestående også viser, ses det tydeligt at den nutidige værdi afhænger af den forgående værdi og så afhænger den af den hvide støj. Det andet som er værd at bemærke er, at φ afgør feedbacken. Jo større dens værdi er, desto større påvirker den den nutidige værdi, hvorimod hvis den slet ikke påvirker, altså er 0, afhænger værdien kun af middelværdien og den hvide støj.

$$Y_t = (1-\varphi)\mu + \varphi Y_{t-1} + \varepsilon_t$$

Vi kan derefter udvide modellen, og AR(2) afhænger altså af de to værdier før, og sådan forsætter det. AR(p) afhænger direkte af de p forudgående værdier og indirekte af alle de forudgående variable. Autoregressive processer må man altså forvente kan bruges til at beskrive finansielle data, da vi må forvente, at den nutidige værdi afhænger af værdierne forud. Problemet med AR processer iforhold til at beskrive finansielle data, er at AR afhænger lineært af sine tidligere værdier. Herunder følger formlen for AR(p);

$$Y_t = \beta_0 + \varphi_1 Y_{t-1} + \cdots + \varphi_p Y_{t-p} + \varepsilon_t$$

Hvor $\beta_0 = (1 - (\varphi_1 + \cdots + \varphi_p))$ og er interceptet. $\beta_0 > 0$ hvis den skal være stationær. Det medfører, at middelværdien $E[Y_i] = \mu = 0$ hvis β_0 også er det.

3.3 MA

Som vist ovenover, er det grundlæggende princip bag AR processer, at den nutidige værdi skal afhænge af de forudgående. Der skal med andre ord altså være en korrealation mellem de tidligere værdier og de nutidige. Nogle gange er der dog kun korrealation i korte perioder, og derfor får AR processer problemer. Her fungerer Moving Average (MA) processer helt anderledes. Definitionen af en MA er process lyder som følgende (Fra Ruppert side 223):

En proces Y_t er en moving average proces, hvis Y_t kan blive udtrykt som en vægtet gennemsnitlig proces af de tidligere værdier af den hvide støj proces ε_t.

Herunder følger formlen på MA(q) processen:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

Det ses på nedenstående Figur 1, at der tilføjes et moving average gennem observationerne. I dette tilfælde er $q = 3$, og det betyder for eksempel at $y_5$afhænger af Y_4 og Y_3, men ikke af den anden observation. Hvis man skal definere det generelt matematisk, kan man sige, at Y_t afhænger af $Y_{t-q}, men\ ikke\ af\ Y_{t-q-1}$. Det ses, at q er antallet af observationer som en observationer afhænger af. Det gør, at vi nu kan "dele" datasættet op ved hjælp af et moving average og dermed beskrive det bedre.

Figur 1. Illustration af et moving average.

3.4 ARMA

Den skarpe læser har formodentlig allerede regnet ud, hvad en ARMA process er, da det er lige nøjagtig, hvad navnet angiver – en process hvor både AR og MA indgår. Vi starter med at skrive formlen for en ARMA(p,q) model ned:

$$(Y_t - \mu) = \varphi_1(Y_{t-1} - \mu) + \cdots + \varphi_p\big(Y_{t-p} - \mu\big) + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q}$$

Ved hjælp af en Backwards operator, en simple notation, angivet som $B^k Y_t = Y_{t-1}$, hvor k er antallet af "skridt" du går baglæns. For eksempel, hvis k=3 så skal vi gå 3 observationer tilbage.

$$\big(1 - \varphi_1 B - \cdots - \varphi_p B^p\big)(Y_t - \mu) = 1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$$

Det ses, at hvid støj er en ARMA (0,0) process. Det ses hurtigt at hvis såvel p som q er lig 0, så fås $(Y_t - \mu) = \varepsilon_t$. Lad os kigge på et næsten lige så simpelt eksempel af en ARMA process, for at skabe et bedre overblik.

$$Y_t = \varphi Y_{t-1} + \theta\varepsilon_{t-1} + \varepsilon_t$$

Ovenstående formel er en ARMA(1,1) process, hvor vi har antaget, at $\mu = 0$. Det ses her matematisk at en ARMA proces afhænger af 3 ting; 1. den tidligere værdi, 2. Støjen før, og til sidst 3. støjen nu. Det samme vil gøre sig gældende for en ARMA(p,q) hvor p og q afgør henholdvis hvor mange tidligere værdier, vi skal have med, samt hvor "mange" perioder som data skal inddeles i. Det er klart, at en model som beskriver data så præcis muligt er at foretrække, men samtidig ønsker vi også en så simpel model som muligt. Med simpel mener jeg at estimationen, den kommer vi til senere, skal være så nøjagtig som mulig – *Det må forventes, at man estimerer mere præcist, hvis der er færre parametre at estimere.* Så senere bliver det et spørgsmål om at finde en balance mellem at beskrive data så godt som muligt, men samtidig også holde modellen så simpel som muligt. Det kan gøres ved hjælp af flere redskaber, men dem kommer jeg til senere.

3.5 ARCH

Vi ved, at finansielle data har perioder med høj volatilitet, ligesom det har perioder med lav volatilitet. Dette kan ARCH og GARCH processer modellere i modsætning til ARMA-modeller. Vi starter med at kigge på ARCH. ARCH er en forkortelse af autoregressive conditional heteroscedasticity. Som forkortelsen viser, så indebærer det, at der er varrierende varians i datasættet, hvilket passer godt med det typiske finansielle data.

Den grundlæggende forskel på AR, MA og ARMA modeller i forhold til ARCH og GARCH er, at vi ved de sidstnævnte kan lade variansen være tilfældig. Ved de først nævnte processer er den betingede varians, afhængig af fortiden, konstant. For ARCH eller GARCH processer er variansen netop ikke konstant, men derimod tilfældig. Støjen har i disse processer en betinget middelværdi lig 0, og en betinget varians lig 1. Det er den grundlæggende forskel på de to typer proceser. Det at modellere modeller hvor den betingede varians ikke er en konstant, kaldes *varians function models*, og her er ARCH og GARCH vigtige.

Vi starter med at kigge på ARCH, som er en forkortelse af *Autoregressive Conditional Heteroscedasticity*. Vi starter med at kigge på en ARCH(1) proces, som er den mest simple version. Vi forsætter med at lade ε_t være normalfordelt hvid støj, men som vi så i den indledende teori så ville den også kunne være t-fordelt. Vi antager, som tidligere nævnt, at støjen har middelværdi 0, og varians 1, og det ser således ud matematisk.

$$E(\epsilon_t|\epsilon_{t-1}, \dots) = 0$$

$$Var(\epsilon_t|\epsilon_{t-1}, ..) = 1$$

Det er meget vigtigt at understrege, da det er den grundlæggende forskel på AR, MA eller ARMA og ARCH eller GARCH – at variansen nu er tilfældig. Det nye i forhold til de tidligere beskrevet proceser, er at den betingede varians på Y-variablene nu får lov at varriere.

Nedenstående viser en ARCH(1) proces, hvor vi krævet at $\omega \; og \; \alpha_1$ er større end 0.

$$a_t^2 = (\omega + \alpha_1 a_{t-1}^2)\epsilon_t^2$$

Udvider modellen ved at tillade variansen at være en tilfældig proces. Som det kan ses minder den meget om AR, dog med a^2 og med multiplikativ støj med middelværdi 1, fremfor additiv støj med middelværdi 0. Nedenstående formel er fundamental for at forstå ARCH og GARCH processer. Bemærk, at vi kan skrive formlen som;

$$a_t^2 = \sigma^2 \varepsilon_t^2 \text{ , hvor}$$

$$\sigma_t^2 = \omega + \alpha_1 a_{t-1}^2$$

Det ses at hvis a_{t-1}, altså værdien før den nuværende, har en usædvanlig stor værdi så er standardafvigelsen også større en normalt og dermed også a_t. Det er hele grundlaget for at GARCH processer er gode til at beskrive finansielle data. For hvis den tidligere værdi er usædvanlig stor, eller lille, så er den næste også, og sådan udbreder det sig så det forsætter dog ikke for evigt. Da a_1 er mindre end 1 vender den betingede varians tilbage til den ubetingede varians. Forskellen på AR og ARCH er, at AR har en konstant betinget varians, og en ikke konstant betinget middelværdi hvor ARCH er det modsatte – altså en ikke konstant betinget varians, og en konstant betinget middelværdi.

Nu går vi videre og kigger på ARCH(p) modeller. Vi forsætter med at antage, at ε_t er normalfordelt hvid støj. Vi bruger den samme måde at definere procesen på som tidligere. Forskellen er, at vi nu lader den betingede varians, σ_t^2, afhænge af flere af de tidligere a-værdier.

$$a_t = \sigma_t \varepsilon_t$$

$$\sigma_t = \sqrt{\omega + \sum_{i=1}^{p} \alpha_i a_{t-1}^2}$$

Her gør det sig, ligesom ved ARCH(1), gældende at standardafvigelsen afhænger af tidligere observationer af processen, og den betinget varians er ikke en konstant.

3.6 GARCH

Alt med ARCH lyder jo lovende, men har også en ulempe, som gør, at GARCH kan være foretrukket over ARCH. Det er, at ARCH kun kan klare at volatiliteten kommer i små bølger, hvor de normalt foregår over længere perioder. GARCH muliggør, at volatiliteten er mere vedvarende end ARCH, og det gør den ved, at standardafvigelsen afhænger af den forrige afvigelse. Det ses af nedenstående formel

$$a_t = \sigma_t \varepsilon_t$$

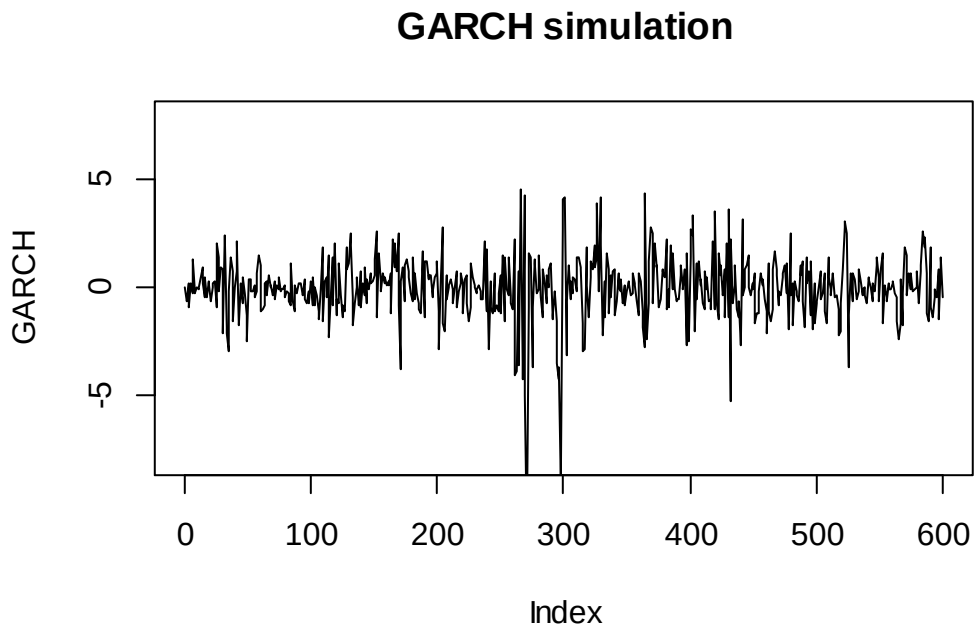
$$\sigma_t = \sqrt{\omega + \sum_{i=1}^{p} \alpha_i a_{t-1}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-1}^2}$$

Vi skal senere hen se nærmere på hvilken betydning det har på et datasæt om man bruger de forskellige modeller beskrevet i dette kapitel. Vi ved at AR og ARMA modeller afhænger lineært af deres tidligere værdier, hvilket vi må formode får en betydning. Derfor antager vi, før vi har kigget på datasættet i Rstudio, at ARCH og GARCH bedre kan beskrive vores finansielle datasæt. Det som både ARCH og GARCH modeller gør er at tillade periode med høj eller lav volatilitet – kendt som volatilitets clusters. ARCH gør det dog for en kortere periode, hvorimod GARCH tillader perioderne at være længerevarende.

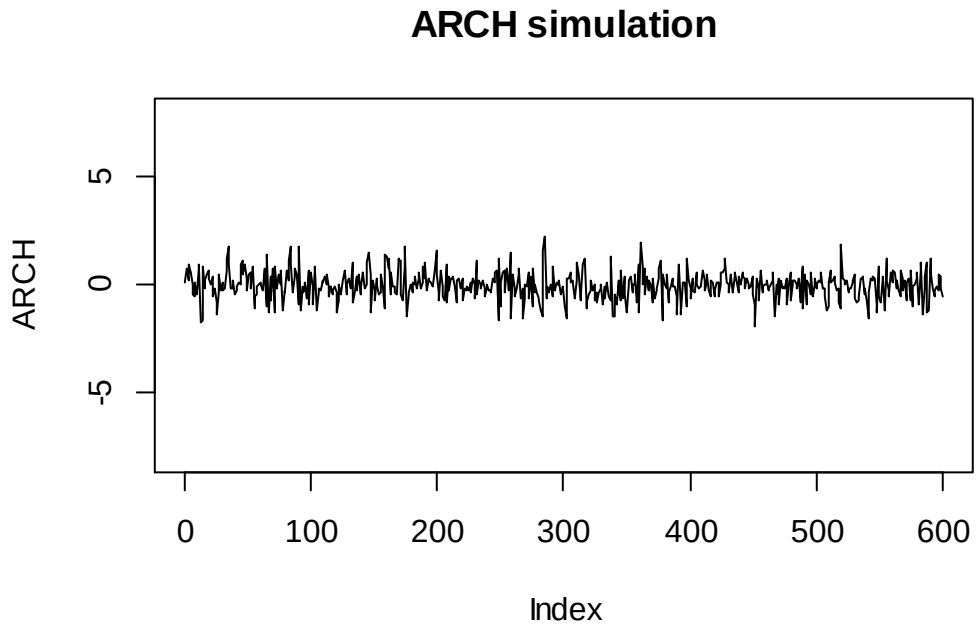
For at vise forskellen er der blevet lavet en simulation af henholdvis en GARCH og ARCH proces. For begge processer har vi $n = 600,\ \omega = 0{,}2\ og\ \alpha = 0{,}7$. For ARCH er β naturligvis 0, imens vi for GARCH procesen har valgt at $\beta = 0{,}3$. Forskellen kan ses på Figur 2 og 3, som findes på næste side. Koden som er blevet benyttet i Rstudio er at finde i Bilag 1 på side 68.

Det ses, at ARCH simulationen laver langt mindre udsving end GARCH simulationen. Det som er værd at bemærke er, at en GARCH model ikke blot lavere større udsving fra deres middelværdi, men også at den bliver derude længere tid. Dette stemmer fint overens med at standardafvigelsen afhænger af den forrige afvigelse. Som nævnt så kan en ARCH proces godt klare hvis volatiliteten kommer i små bølger, hvor GARCH muliggør at den er mere vedvarende. Disse to simulationer giver et godt billede af forskellen på ARCH og GARCH, og de kan måske gøre, at vi kan konkludere at en GARCH proces er mere passende før vi kigger på de forskellige modeller. Med denne

information kan vi muligvis, blot ved at kigge på vores datasæt, afgøre om en ARCH eller GARCH proces kan beskrive det bedst.



Figur 2. GARCH simulation.



Figur 3. ARCH simulation.

3.7 ARMA/GARCH

En anden model som vi kan benytte, er en ARMA-GARCH model. Vi kigger her udelukkende på den mest simple model, nemlig ARMA(1,1)-GARCH(1,1) model. Her bruges ARMA til at modellere middelværdien, og hvor GARCH bruges til at modellere variansen. I dette tilfælde er støjvariablen, ε_t, ikke uafhængig hvid støj, men i stedet er givet ved en GARCH proces. Det skal dog bemærkes, at der forsat er tale om svag hvid støj. Matematisk ser det således ud;

$$Y_t = \mu + \varphi Y_{t-1} + \theta \varepsilon_{t-1} + \varepsilon_t$$

$$a_t = \sigma_t \varepsilon_t$$

$$\sigma_t = \sqrt{\omega + \alpha_1 a_{t-1}^2 + \beta_1 \sigma_{t-1}^2}$$

3.8 EGARCH

Den sidste model som vi vælger at bruge kaldes EGARCH. Vi har denne model med, da flere har observeret, at negative og positive prisændringer påvirker volatiliteten forskelligt. Vi vil, i dette afsnit, udelukkende se på EGARCH(1,1), som er givet ved nedenstående formel. Senere skal vi dog også se på EGARCH processer af større orden, men princippet er det samme og burde kunne modellere assymetri bedre end en standard GARCH proces.

$$\log(\sigma^2) = \alpha_0 + \frac{\alpha_1 a_{t-1} + \gamma_1 |a_{t-1}|}{\sigma_{t-1}} + \beta_1 \log(\sigma_{t-1}^2)$$

Den helt store fordel ved EGARCH sammenlignet med de øvrige er, at den tillader at positive og negative ændringer påvirker forskelligt. Det vil senere hen vise sig om denne model er den bedste.

Kapitel 4 – Data.

4.1 Indledning

Målet med dette kapitel er at beskrive datasættet som bliver benyttet til denne afhandling. Derudover skal det indeholde hvilke overvejelser og planer jeg har med hensyn til hvordan datasættet skal bruges til at svare på problemformuleringen.

4.2 Data

Datasættet som skal bruges i den opgave er Danske Bank aktien. Aktien er noteret på NASDAQ København og er en del af både OMX C20CAP indekset og det nye C25 indeks.[1]Man kan finde aktien på Danske Banks egen hjemmeside og se hvordan den har udviklet sig historisk. Der kan man så vælge en tidsperiode og derefter hente periodens data ned i et Excel dokument. Det er hvad jeg har gjort, men først skulle jeg vælge et tidsinterval.

Tidsintervallet som jeg har valgt at benytte til denne opgave går fra den 26. Marts 2007 til den 23. Oktober i 2017. Det er en periode som indeholder 2642 handelsdage, og vi har altså observationer fra alle disse dage. Det er altså en periode på mere end 10 år, og burde være et tilpas langt interval hvorfra vi kan bruge GARCH modeller til at beskrive datasættet. Derudover er intervallet langt nok til at vi kan forecaste. Senere skal vi vise at forecasting udfra hele perioden ikke giver den store mening, men mere om det senere.

Hver enkelt observation, en handelsdag, indeholder 5 variable. Vi har Lukkekurs, som er kursen som aktien lukkede med den pågældende dato. Volumen som er mængden af handler af aktien den pågældende dag. Derudover indeholder datasættet en kollone som kaldes Åben, som er hvilken kurs aktien åbnede datoen med. Til sidst indeholder det ligeledes en Høj og Lav, som fortæller henholdvis den højeste og laveste kurs på datoen. Det ses at datasættet er med de nyeste datoer først, men det kan nemt ændres i R. For at kunne benytte teorien skal datasættet nemlig være i kronologisk rækkefølge.

[1] Det skriver Danske Bank selv på deres hjemmeside. https://danskebank.com/da/investor-relations/aktien

Da afhandlingen skal beskrive finansielle data, har vi et godt datasæt da det jo er finansielle data vi har. Da vi samtidig nemt kan sætte det i kronologisk rækkefælge, har vi alle forudsætninger for at beskrive finansielle data ved hjælp af de forskellige modeller. Vi kan sammenligne de forskelige, og se hvilken passer vores datasæt bedst. Derudover kan vi ved hjælp af modellerne forecaste indenfor, og udenfor, tidsintervaller som datasættet indeholder. Datasættet må altså formodes at være mere end tilfredsstillende til at hjælpe med at besvare problemformuleringen.

Kapitel 5 – Indledende dataanalyse.

5.1 Indledning

Yogi Berra sagde engang ”You can see a lot by just looking”[2] og det er netop, som vi gør i dette kapitel.

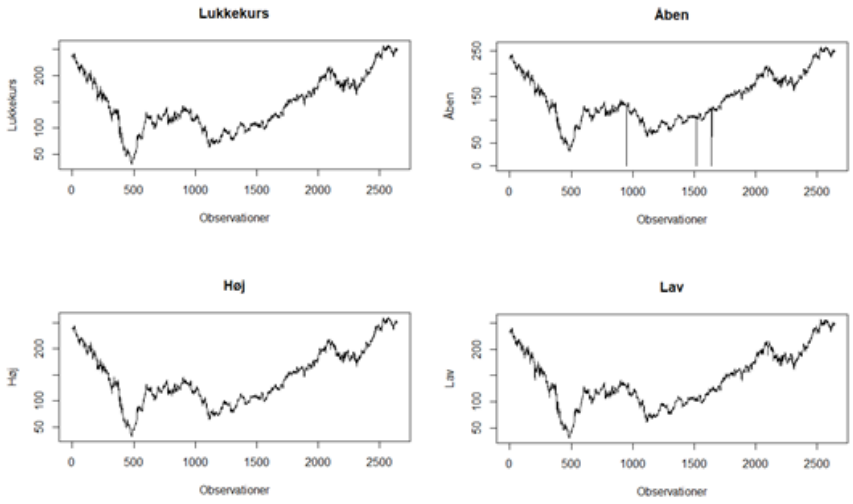
Som tidligere nævnt har finansielle data nogle særlige kendetegn, og dem skal vi nu se nærmere på. Vi skal altså se på hvordan datasættet ser ud, og forklare hvorfor vi skal anvende de forskellige modeller som er nævnt tidligere. Derudover skal vi gøre nogle antagelser i forhold til datasættet, som skal gøre den videre dataanalyse, som kommer i kapitel 7, nemmere og mere optimal.

5.2 Indledende dataanalyse

Det første der skal gøres er at sætte datasættet i kronologisk orden, det vil sige at vi starter i 2007 og ender i 2017. Dette gøres nemt og hurtigt i Rstudio.

Til at starte med skal vi sikre os at datasættet ikke har manglende observationer. Det ses, at der lader til at være noget manglende ved ”Åben”, men de tre øvrige mangler ingen observationer, og det lader ikke til at der er den store forskel på dem. Jeg vælger at bruge ”Lukkekurs” delen fra datasættet til den resterende del af opgaven. Det er muligt, at man kan argumentere for de to andre, men lukkekursen har den fordel, at det er samme tidspunkt fra handelsdag til handelsdag. Både ”Høj” og ”Lav” varierer i tidspunkt fra dag til dag – eller det må man forvente, da det vil være særdeles usandsynligt at det er samme tidspunkt at kursen er høj eller lav adskillige dage i træk. Det at have et stabilt tidspunkt, samt at det er begrænset hvor meget den varierer fra dag til dag, gør at lukkekursen er at foretrække til denne opgave. Det skal især ses med henblik på forecastingen, som kommer senere i opgaven. Herunder er vedhæftet de 4 plots for de 4 variable vi har i datasættet; Åben, Lukkekurs, Høj og Lav. Det ses, som også netop er blevet beskrevet, at der ikke er den helt store forskel på dem, hvor Åben dog har et par enkelte nulværdier. Derudover er det værd at bemærke at vi kommer ind midt under finanskrisen i 2007.

[2] Ruppert side 2.



Figur 4: Her ses de fire variable som indgår i vores data, og det ses at der ikke er nogen markant forskel udover nulværdierne for variablen åben.

Da vi kommer ind midt under krisen har jeg valgt at benytte de sidste 2100 observationer, det vil altså sige at de første 542 er blevet taget ud af datasættet. Det betyder at det "nye" datasæt begynder i Juni 2009 efter finanskrisen, og slutter i Oktober 2017. Det gør, at vi kan forecaste fra efter krisen, hvor markedet er opadgående og mere stabilt, og se om der er forskel på forecasting der og senere hen i tidsintervallet. Vi skal huske på, at C20-indekset repræsenterer de mest handlede aktier, og derfor må Danske Bank aktien i høj grad afspejle det generelle marked. Vi ved at i finansielle data, så findes der "regimeskift" – hvor udviklingen i kurserne er forholdvis persistent og hele markedet er påvirket af disse faktorer. Vi får derfor mulighed for at forecaste og estimere når markedet er vendt efter krisen, og ser hvordan vores forskellige forecasting modeller fungere.

Fra nu af i denne opgave, hvis ikke andet angivet, vil vi bruge "Lukkekurs" fra det nye datasæt hvor vi kommer ind i Juni i 2009. Vi kigger nu på det daglige afkast, som findes ved nedenstående formel. Vi ser at r_t afhænger af aktien til tid P_t og kursen fra dagen før P_{t-1}.

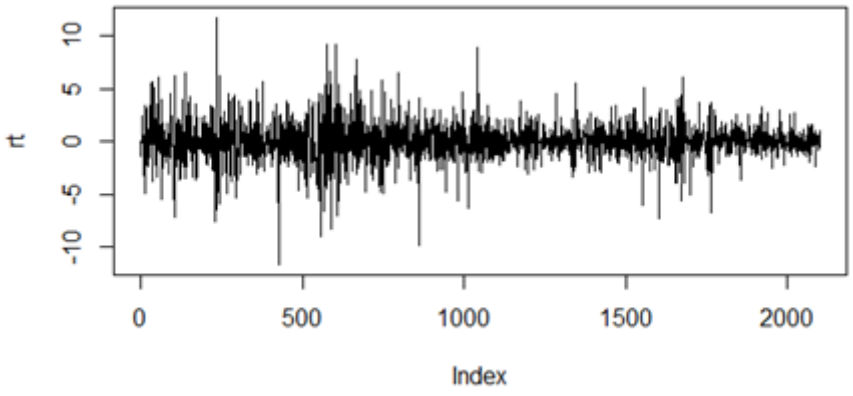
Det er alment kendt at log-afkast, også kaldt *continuously compounding returns*, er mest nyttigt til tidsrækkeanalyser, og derfor hvad vi benytter i denne opgave. Der er flere årsager til, at det er det mest ubredte, hvilket blandt andet skyldes tidsadditivitet samt, at de er normalfordelte. Hvis afkastet er lille (under 10 procent) så er logafkastet næsten det samme som afkastet selv. Matematisk er det givet ved formlen $\log(1+r) \approx r$. For eksempel kan vi se at hvis vi har et

afkast på $\pm$syv procent, så er logafkastet henholdvis 6.77% eller -7.26%. Det er klart at des mindre afkastet er, des tættere er log-afkastet på den sande værdi af afkastet. Ruppert har beskrevet det meget kort, og ellers kan der refereres til quantivity linket som beskriver fordelen ved logafkast særdeles godt.

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

Som tidligere nævnt benyttes Rstudio til at analysere data, og her er det formlen $\log(P_t) - \log(P_{t-1})$ som er blevet implementeret.

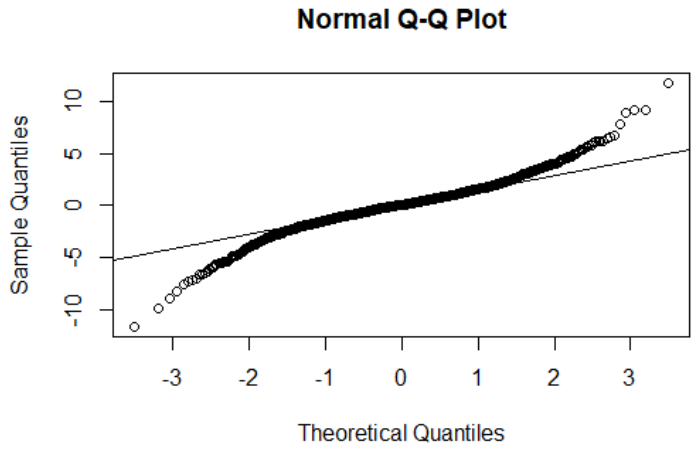
Vi plotter herefter afkastet, og ser hvordan det ser ud. Det ses i Figur 5 herunder. Det ses at det indeholder klassiske kendetegn for finansielle data. Det ses at der er store udsving i afkastet, hvilket er naturligt for finansielle data. På trods af de store udsving lader det dog til, at det er stationært, siden at variationen er konstant over hele tidsintervallet, og at det svinger udfra et fast niveau. Derudover ser vi også volatilitetsklyngning, altså perioder med høj, eller lav, variation. Det kan muligvis ses som et tegn på afhængighed indenfor den betinget varians i datasættet. Dette gør, at at GARCH højst sandsynlig er at foretrække over ARMA – vi husker at ARMA ønsker konstant varians, og det lader bestemt ikke til at være tilfældet i hele perioden.



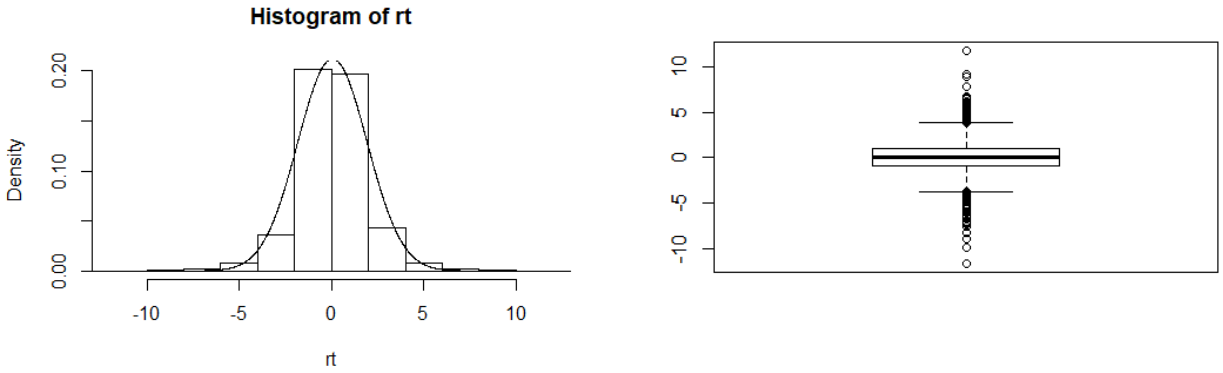
Figur 5. Det daglige log-afkast af vores datasæt efter vi har fjernet observationer fra finanskrisen.

Herefter kigger vi på fordelingen af vores afkast. Det er normalt at der er tunge haler i finansielle data, hvilket også gør sig gældende i vores tilfælde. Figur 6, som er QQ plottet, viser tunge haler i forhold til en normalfordeling.

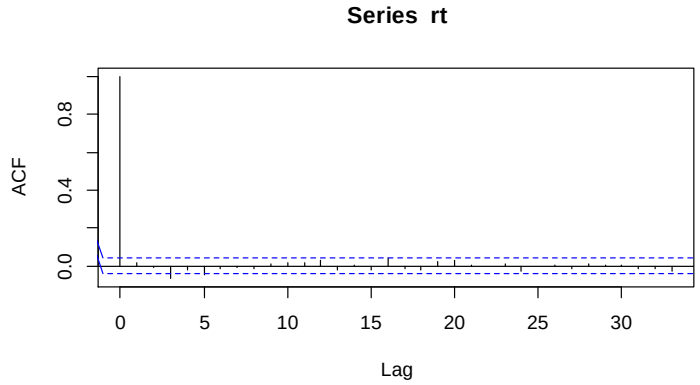
Histogramet, Figur 7 sammen med Boxplottet, ser rimelig normalfordelt ud, men har ligeledes tunge haler. Boxplottet viser også klare tegn på tunge haler. Det ses, at der er adskillige observationer som ligger langt fra boxen, som indeholder data fra første til tredje kvartil – Boxen indeholder altså den midterste halvdel af datasættet. Det er ikke overraskende at datasættet har tunge haler, da finansielle data generelt er meget udsatte overfor outliers (ekstreme værdier). Det lader dog til at der i vores datasæt er adskillige "ekstreme" værdier, hvilket kan tyde på at der ganske enkelt bare er perioder hvor variationen er større end generelt. Det er endnu en ting som kan tale for at GARCH modeller kan være at foretrække da den jo netop kan klarer at variansen "kommer i bølger" og er mere vedvarende.



Figur 6. QQ-plot for vores log-afkast.



Figur 7. Til venstre ses histogrammet for vores log-afkast, og boxplottet til højre.



Figur 8. ACF for vores afkast.

Det ses udfra Figur 8, at der ikke er nogen klar korrelation for logafkastet. Ljung Box testen for K=5 (outputtet fra Rstudio ses herunder) indikerer, at man kan forvente, at mindst en af de første fem autokorelationer er forskellig fra 0. Det er dog ikke tilfældet, ved et signifikant niveau på 5%, hvis K er 10, 15 eller 20, hvor p-værdien er så stor at vi ikke kan afvise at nulhypotesen om at autokorelationen er 0.

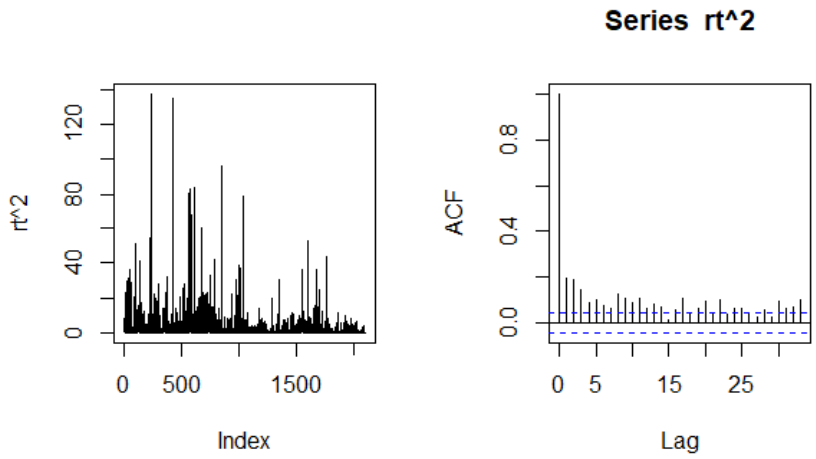
```
        Box-Pierce test

data:  rt
X-squared = 14.999, df = 5, p-value = 0.01037
```

R output 1. Resultatet af Ljung-Box testen for K=5.

Ved hjælp af pakken *skewness* i Rstudio, fås en skævhed på cirka -0.070, hvilket indikere at fordelingen er rimelig symmetrisk på trods af tunge haler dog med marginal tungere venstre hale end den højre. De tunger haler kan ses QQ-plottet, men ved hjælp af Rstudios pakke *kurtosis* fås en værdi på lige næsten 7, hvilket er mere end det dobbelte af hvad der forventes ved en normalfordeling.

Nu går vi videre til at kigge på det kvadreret log-afkast. Vi kigger på dem da det er hvad GARCH modellerne benytter, og vi kan få en indikation på om variansen er konstant over tid.

Nedenstående figur viser til venstre plottet for det kvadreret log-afkast hvor den til højre er ACF. Det ses på plottet til venstre at volatilitetscluster er blevet endnu tydeligere nu end de var før vi kvadrededede log-afkastet. Derudover viser sample ACF plottet at der nu er klar korrelation, det bekræftes også af en Ljung Box test hvor vi ser at korrelationen er forskellig fra 0 hvor K er 5, 10, 15 eller 20. Herefter køres Kurtosis og skævheden i Rstudio og det ses her at Kurtosis viser et klart tegn på de karakteriske tunge haler da den er tæt på 81. En skævhed på over 7 indikere at der er tale om en tung højre hale.



Figur 9. Kvadreret log-afkast og sample ACF for det kvadreret log-afkast.

Det lader altså til, at vi kommer til at få stor gavn af at benytte de forskellige GARCH modeller da vi har perioder med høj og lav volatilitet. Derudover kunne det tyde på, at den hvide støj har bedre af hvis fordelingen er andet end normalfordelt. Selve fordelingen af vores data minder mest om en GARCH proces hvis vi husker tilbage på ARCH og GARCH simulationen. Udfra den indledende dataanalyse, kan vi forestille os at GARCH passer bedre til datasættet, ligesom at støjvariable i hvert fald ikke lader til at være normalfordelt. Det kigger vi dog nærmere på i den store dataanalyse. Dette kapitel er til for at give os et overblik over datasættet, hvilket vi har fået nu. Derudover har vi fået kigget på nogle af de særlige kendetegn som finansielle data indeholder.

Kapitel 6 – Estimation og forecasting teori.

6.1 Indledning

Dette kapitel omhandler estimation og forecasting for forskellige GARCH modeller. Vi starter med at kigge på estimationen, hvorefter vi går videre til forecasting. Til beskrivelsen af estimationsmetoden er Würtz, Chalabi og Luksan blandt andet blevet anvendt. Forecasting afsnittet tager blandt andet udgangspunkt i Reiders, 2009.

6.2 Estimation

I dette afsnit skal vi beskrive, hvordan parameterne bliver estimeret i vores modeller. Man kan godt benytte lineeær regression til at estimere, men vi skal i dette afsnit beksrive maximum log-likelihood estimation (forkortet MLE). Grunden til at vi benytter denne estimationsmetode er at det er den som Rstudio benytter sig af. Vi skal i det næste kapitel se hvor nemt og smertefrit vi kan estimere de ønskede parametre, men førts skal vi kende teorien bag.

For at estimere parameterne med MLE skal vi naturligvis skabe en likelihood funktion, som er en fælles sandsynlighed tætheds funktion. I stedet for at tænke på likelihooden som en funktion af data givet vores parametre, skal vi nu tænke på den som en funktion af parameterne givet vores data. Matematisk ser det således ud; $L(\theta|y_1, y_2, \dots, y_n)$, hvor θ er alle parametre vi ønsker at estimere. I en GARCH(1,1) model er det nedenstående parametre vi ønsker at estimere;

$$\theta = \{\mu, \omega, \alpha_1, \beta_1\}$$

Vi ved, at i en GARCH model så er afkast ikke uafhængige af hinanden, hvilket gør at vi skal skrive den fælles sandsynligheds tæthed funktion som et produkt af den betinget tæthedsfunktion som er vist herunder.

$$f(y_1, y_2, \dots, y_n) = f(y_n|y_1, y_2, \dots, y_{n-1}) f(y_{n-1}|y_1, y_2, \dots, y_{n-2}) \dots f(y_1)$$

Vi kan nu opskrive likelihood funktionen for en GARCH (1,1) model, som i dette tilfælde er normalfordelt.

$$L(\mu,\omega,\alpha_1,\beta_1|y_1,y_2,\dots,y_n)=\frac{1}{2\sqrt{\pi\sigma_n^2}}e^{-\frac{(y_n-\mu)^2}{2\sigma_n^2}}\frac{1}{2\sqrt{\pi\sigma_{n-1}^2}}e^{-\frac{(y_{n-1}-\mu)^2}{2\sigma_{n-1}^2}}\dots\frac{1}{2\sqrt{\pi\sigma_1^2}}e^{-\frac{(y_1-\mu)^2}{2\sigma_1^2}}$$

Vi tager logaritmen for at nå frem til loglikelihoodfunktionen. Efter lidt reduktion nås frem til nedenstående for en normalfordeling;

$$Log\big(L(\mu,\omega,\alpha_1,\beta_1|y_1,y_2,\dots,y_n)\big)=-\frac{n}{2}Log(2\pi)-\frac{1}{2}\sum_{i=1}^{n}Log(\sigma_i^2)-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{(y_i-\mu)^2}{\sigma_i^2}\right)$$

For GARCH modeller kan vi erstatte $\sigma_i^2=\omega+\alpha_1 a_{i-1}^2+\beta_1\sigma_{i-1}^2$, og vi har dermed en funktion som kun afhænger af afkastet og de ønskede parameter. Det skal dog nævnes at vi også skal estimere volatiliteten til at starte med, dvs σ_1. I vores tilfælde burde det dog ikke få den store betydning da vores tidsrække er lang hvilket vil medfører at den ikke bliver nogen væsentlig faktor. Herunder følger log likelihood funktionerne for henholdvis t-fordelt og GED fordelt. Samme fremgangsmåde hvor vi kan erstatte volatiliteten til sidst, og derudover er det værd at bemærke at vi skal estimere en ekstra parameter, v.

$$\begin{aligned}&Log\big(L(\mu,\omega,\alpha_1,\beta_1,v|y_1,y_2,\dots,y_n)\big)\\&\quad=n*Log\left[\frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{\pi(v-2)}\Gamma\left(\frac{v}{2}\right)}\right]-\frac{1}{2}\sum_{i=1}^{n}Log(\sigma_i^2)-\frac{v+1}{2}\\&\quad-\frac{1}{2}\sum_{i=1}^{n}Log\left(1+\frac{(y_i-\mu)^2}{\sigma_i^2(v-2)}\right)\end{aligned}$$

$$\begin{aligned}&Log\big(L(\mu,\omega,\alpha_1,\beta_1,v|y_1,y_2,\dots,y_n)\big)\\&\quad=n\left(Log(v)-\mathrm{Log}(\lambda)-\left(1+\frac{1}{v}\right)\mathrm{Log}(2)-\mathrm{Log}\left(\Gamma\left(\frac{1}{v}\right)\right)\right)-\frac{1}{2}\sum_{i=1}^{n}Log(\sigma_i^2)\\&\quad-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{(y_i-\mu)^2}{\lambda^2\sigma_i^2}\right)^{\frac{v}{2}}\end{aligned}$$

6.3 AIC og BIC

Når vi er færdige med at fitte vores forskellige modeller og ser på estimationen, skal vi naturligvis sammenligne dem. Vi skal se hvor godt de forskellige modeller beskriver data, og se på hvor præcist vores variable er estimeret. Den mest normale måde at sammenligne modellers brugbarhed på samme datasæt, uanset hvor kompliceret modellerne hver især måtte være, er ved at anvende et informationskriterium. Et første bud på et informationskriterium kunne være være at benytte minus Loglikeliihoodfunktionen, da denne meget passende måler hvor godt modellen passer til data. Dog vil dette kriterie foretrække mere komplekse modeller fremfor en simplere udgave af den samme model. Derfor benyttes et lidt mere sofistikeret kriterium, hvor graden af modellernes kompleksitet tæller den modsatte vej.

Grundtanken bag informationskriterietet er, at hver model straffes for dennes kompleksitet, som i vores tilfælde er antallet af parametre. Det er klart at en simpel model er at foretrække, men den skal ikke være for simpel, da den kan beksrive datasættet dårligere. Derfor kigger informationskriterie test på brugbarheden og straffer modellen des flere parametre, som anvendes. Vi anvender i denne opgave primært Akaike informationskriteriumet (AIC), men vi nævner også kort Bayesian informationskritereiumet (BIC) da dette også er udbredt i modelsammenligninger. AIC og BIC er begge defineret herunder. For begge kriterier er en lav værdi at foretrække fremfor en højere. Det ses, at begge kriterier benytter den maksimeret logfunktion for modellen med k parametre. Derudover ses det at BIC også anvender antallet af N, som er observationer benyttet.

$$AIC = 2k - 2Log\left(\hat{L}\right)$$

$$BIC = kLn(N) - 2Log\left(\hat{L}\right)$$

6.4 Forecasting

De tre største årsager til at forecaste volatiliteten er *i)* Risikostyring, *ii)* Fordelingen af aktiver og *iii)* for at gætte på den fremtidige volatilitet. En stor del af at risikostyre sine aktiver kræver at man estimerer fremtidigens volatilitet og korrelationer. Den mest kendte måde at fordele sine aktiver på er ved at minimere risikoen for et givet niveau af forventet afkast. Den simpleste måde at estimere volatiliteten er, at benytte sig af historiske data og dennes standardafvigelse.

Vi kan omskrive en GARCH(1,1) så den ser ud som herunder. Det ses, at variansen til tidspunkt *t*, er en vægtet sum af tidligere afkast, og af tidligere $\beta' er$. Det ses, at det er de nyeste værdier som vægter mest i en forecasting. Dette giver god mening i en GARCH model, da vi tidligere har nævnt at der findes volatilitetesclusters. Det vil sige, at hvis et givent tidspunkt har høj volatilitet, så er der større sandsynlighed for at næste tidspunkt også har høj volatilitet, end hvis forrige tidspunkt har lav volatilitet. Derfor er det en god idé, at den seneste værdi har den største betydning på den nutidige fremfor at de alle havde lige så betydning. Dette gør, at vi forventer fornuftige resultater af vores forecasting såfremt vores model passer datasættet nogenlunde. Man kan dog også opleve, at forecastingen er baseret på noget data, hvortil fremtiden ikke gør. Hvis vi havde valgt at beholde hele datasættet og estimerede vores parametre på baggrund af forløbet før og under krisen, ville vi formentlig havde svært ved at forecaste fornuftige for efter krisen da datasættet derefter opfører sig anderledes. Ved at fjerne første del af vores data har vi gjort vores datasæt mere stabilt, og det burde resultere i en bedre forecasting end ellers.

$$\sigma_t^2 = \frac{\omega}{1-\beta_1} + \alpha_1 \sum_{i=0}^{\infty} a_{t-1-i}^2 \beta_1^i$$

Vi starter nu med at kigge på forecasting for næste tidpunkts varians $\hat{\sigma}_{t+1}^2$. Her har vi erstattet den ubetingede varians; $\sigma^2 = \frac{\omega}{1-\alpha_1-\beta_1}$.

$$\sigma_t^2 = \omega + \alpha_1 a_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

$$\hat{\sigma}_{t+1}^2 = \omega + \alpha_1 E[a_t^2 | I_{t-1}] + \beta_1 \sigma_t^2$$

$$\hat{\sigma}_{t+1}^2 = \sigma^2 + (\alpha_1 + \beta_1)(\sigma_t^2 - \sigma^2)$$

Samme fremgangsmåde kan benyttes til at forecaste for den næste periode, og til sidst kan vi forecaste for l skridt frem som vist herunder. Det er værd at bemærke, at hvis l går mod uendelig så vil vores forecast gå mod den ubetinget varians $\frac{\omega}{1-\alpha_1-\beta_1}$. Derudover skal det nævnes at det er α_1 og β_1 som afgører hvor hurtigt at vores forecast går mod den førnævnte ubetinget varians.

$$\hat{\sigma}_{t+2}^2 = \omega + \alpha_1 E[a_t^2|I_{t-1}] + \beta_1 E[\sigma_{t+1}^2|I_{t-1}]$$

$$\hat{\sigma}_{t+2}^2 = \sigma^2 + (\alpha_1 + \beta_1)^2(\sigma_t^2 - \sigma^2)$$

$$\hat{\sigma}_{t+l}^2 = \sigma^2 + (\alpha_1 + \beta_1)^l(\sigma_t^2 - \sigma^2)$$

Det må forventes, at en forecasting har fejl og ikke er helt præcis. Derfor er *forecast error* et nyttigt begreb. Den har nedenstående formel, og det kan ses, at det er forskellen på vores kvadreret afkast minus den betinget forventede værdi af afkastet - I_{t-1} er den informationen vi har til rådighed på det givne tidspunkt.

$$e_t = a_t^2 - E[a_t^2|I_{t-1}] = a_t^2 - \sigma_t^2$$

I næste kapitel gennemgår jeg et eksempel med MLE estimationen i Rstudio, hvor ovenstående teori er blevet anvendt.

Kapitel 7 – Estimation.

7.1 Indledning

Dette kapitel indeholder teorien fra de tidligere kapitler og denne teori anvendes på vores datasæt. Igennem dataanalysen er det ambitionen at nå frem til den model, som beskriver data bedst muligt - George Box sagde ”All models are false but some models are useful”, og vi ønsker at finde i hvert fald en brugbar model.

7.2 Modelanalyse

For at finde ud af hvilken model som er den bedste skal vi naturligvis gennemgå adskillige modeller. Vi starter med modeller med lav orden, estimerer disse modeller først og ser hvor godt de fitter datasættet. Herefter gør vi det samme for modeller med større orden og flere variable. Således forsætter vi imens vi sammenligner vores modeller, og til sidst har vi fundet den mest optimale model til vores datasæt. Som tidligere nævnt, må det forventes, at vi estimerer mere præcist, hvis vi har færre parameter at estimere, og derfor er en model med så få parameter som muligt ønsket. Desuden er en simplere model lettere at fortolke og nemmere at arbejde med. Det er dog klart, at modellen ikke skal være for simpel, men alt dette kigger vi på nu.

Vi starter med at kigge på en GARCH(1,1) model hvor vi blandt andet får nedenstående output fra Rstudio. Outputtet kommer fra ugarchfit funktionen, som giver os den nødvendige information til, at vurdere hvor godt vores valgte model fitter datasættet. Outputtet bliver gennemgået grundigt for GARCH(1,1) modeller med forskellig fordeling, hvorefter vi har grundlaget til at sammenligne med øvrige modeller uden at skulle i dybden med samtilige modeller. Det er samme fremgangsmåde, som bliver benyttet til at vurdere hvor godt modellerne fitter datasættet.

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.096119    0.032678   2.9414 0.003268
omega   0.074449    0.025092   2.9670 0.003007
alpha1  0.103404    0.017946   5.7621 0.000000
beta1   0.880586    0.020987  41.9583 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.096119    0.029150   3.2975 0.000976
omega   0.074449    0.050828   1.4647 0.142990
alpha1  0.103404    0.039389   2.6252 0.008660
beta1   0.880586    0.047922  18.3756 0.000000

LogLikelihood : -4098.527

Information Criteria
------------------------------------

Akaike       3.9090
Bayes        3.9198
Shibata      3.9090
Hannan-Quinn 3.9130
```

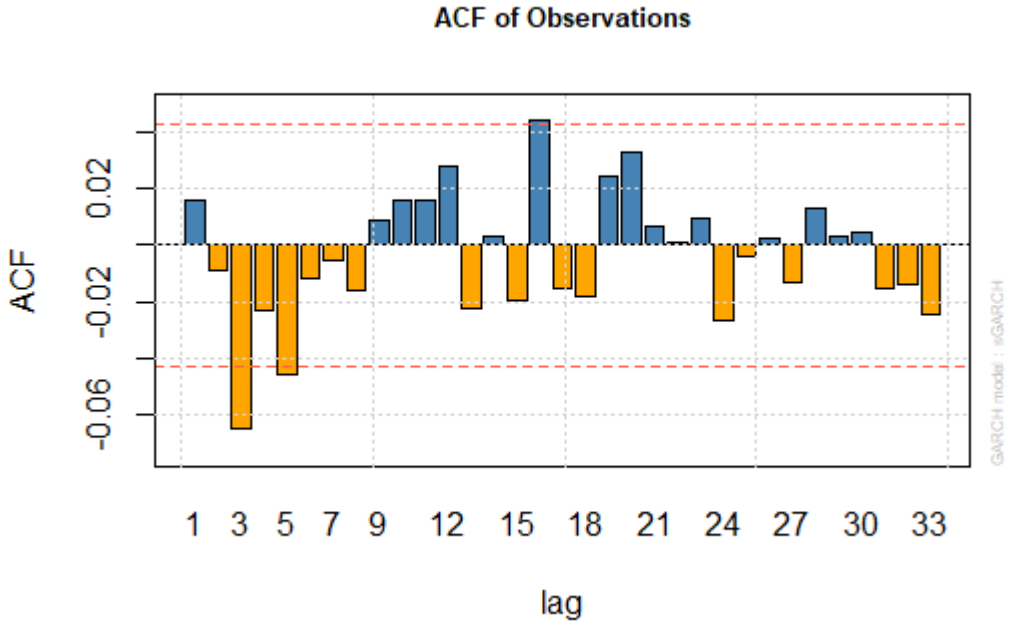
R output 2. Parameter estimation, Loglikelihooden samt informationskriterie for en GARCH(1,1) med normalfordelt støj.

Det ses her, at vores GARCH model er en standard GARCH(1,1), og at det i dette tilfælde er støjen en normalfordeling som vi har fittet for. Vi sammenligner senere normalfordelingen med de øvrige, som er t-fordelt og GED-fordelt for at komme til en konklussion om hvilken fordeling er passende. Derefter skal vi kigge på de øvrige GARCH-modeller som blev beskrevet i modelafsnittet. Det ses, at outputtet giver os 2 forskellige estimater for hver parameter. Den som vi skal benytte os af er den øverste, da disse er fremkommet ved hjælpen af estimationen som tidligere er blevet beskrevet teoretisk. Fremover vil der kun blive vist output med, hvad vi benytter os af, men I dette tilfælde vil vi blot vise, at der også er en anden estimationsform i vores Rstudio-pakke output.

Vores estimerede parameter er mu som er $\hat{\mu}$, omega som er $\widehat{\omega}$, alpha1 som er $\widehat{\alpha_1}$, og til sidst har vi beta1 som er $\widehat{\beta_1}$. Hvis vi kigger på den estimation vi skal bruge, de som står før Robust Standard errors, så ser vi, at vi har 4 forskellige output for hver parameter; *Estimate, Std. Error, t value* og til sidst $Pr(> |t|)$. Vi gennemgår hurtigt deres betydning nu.

Estimate; Estimatet af vores parameter. Det er blandt andet meget bemærkelsesværdigt at $\widehat{\beta_1} \approx 0.88$ da det indikerer volatilittets clustering. Hvis det ikke var tilfældet ville parameteren været meget nærmere 0. *Std. Error*; Standard afvigelsen af vores estimatet. Herefter kommer *t value*; som fortæller om hvor mange standardafvigelser vores estimate er væk fra 0. Det ses, at både mu og omega er mindre end 3 standard afvigelser fra 0. T værdien bruges blandt andet også til at udregne det sidste som er $Pr(>|t|)$; en lille værdi her fortæller os, at vores estimat har et forhold til vores data. Normalt bruges en p-værdi på 5% som en meget god grænse, og vi kan se at alle vores estimater i dette tilfælde er signifikante.

Det sidste som vi kan se fra ovenstående del af outputtet er LogLikelihood, men det vender vi tilbage til lidt senere. Dette tal fortæller os ikke rigtigt noget, men det kan bruges når vi lidt senere skal sammenligne forskellige modeller. Det samme gør sig gældende med *Information Criteria* som findes herunder. Det kan hurtigt nævnes, at Akaike og Bayes er henholdvis AIC og BIC som tidligere er blevet beskrevet. Derfor vil vi fremover heller ikke have output med Shibaya og Hannan-Quinn med. Før vi går videre, skal det nævnes at AIC og BIC er normaliseret (delt med 2099), og det betyder, at forskellen reelt er større end, hvad den virker til. Dette kigger vi ligeledes på om et øjeblik, efter vi har færdiggjort at kigge på det sidste output.

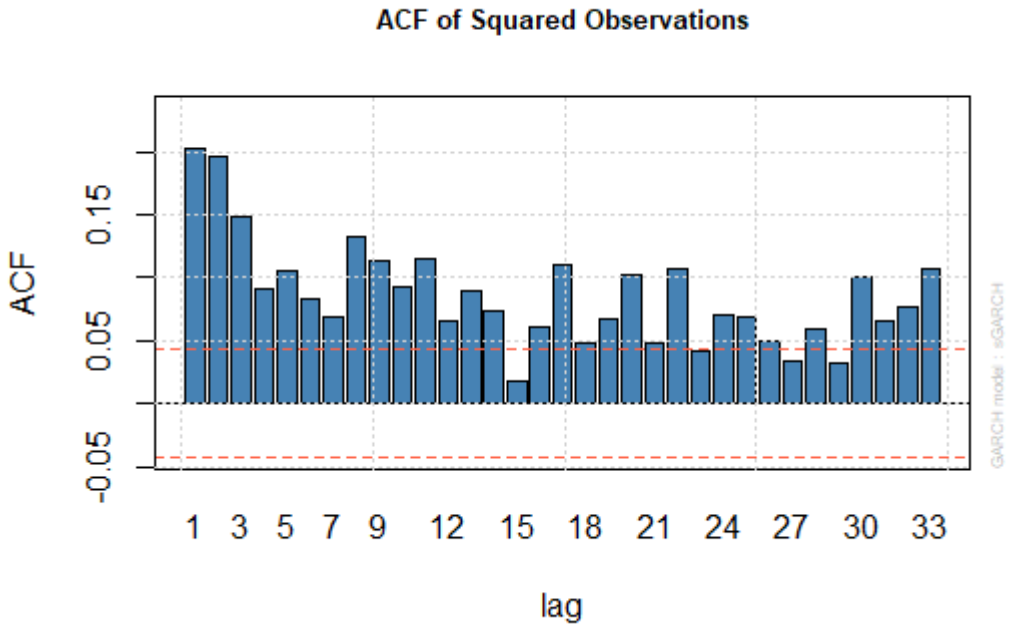


Figur 10. ACF for GARCH(1,1) med normalfordelt støj.

```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.1432  0.7051
Lag[2*(p+q)+(p+q)-1][2]     0.1580  0.8807
Lag[4*(p+q)+(p+q)-1][5]     1.9088  0.6401
d.o.f=0
H0 : No serial correlation
```

R output 3. Ljung-Box test på de standardiserede residualer.

Hvis vi starter med at kigge på de standardiserede residualer, hvor Figur 10 viser ACF plottet, og Ljung-Box testet. Det ses, at modellen lader til at fjerne serie korrealationen, og at vores model derfor muligvis er brugbar. Det samme gør sig gældende hvis vi kigger på de kvadreret residualer, hvor vi igen har plottet ACF og kigget på Ljung-Box testen.



Figur 11. ACF for de kvadreret afkast af GARCH(1,1) med normalfordelt støj..

```
Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.5309  0.4662
Lag[2*(p+q)+(p+q)-1][5]     1.1877  0.8159
Lag[4*(p+q)+(p+q)-1][9]     2.6190  0.8198
d.o.f=2
```

R output 4. Ljung-Box test på de standardiserede kvadreret residualer.

Herefter kigger vi på *Sign Bias test.* Testen fortæller, om vores valgte model i tilstrækkelig grad fanger det som kaldes en leverage-effekt i data. Som tidligere nævnt så kan finansielle data blive påvirket mere af negative ændringer end de positive, og derfor er dette test godt at have med til at vurdere modellen. *Sign Bias testet* tester hvordan 3 faktorer påvirker volatiliteten, og de 3 faktorer er; Fortegnet, virkningen af størrelsen på negative stød, og til sidst virkningen af størrelsen på positive stød. For at modellen skal fitte godt til data, så skal vi have at alle værdierne (eller som minimum den fælles effekt) i Sign Bias testen er over 5%. Det ses i denne model, at det ikke er tilfældet, og derfor skal vi muligvis bruge en asymmetrisk model til at fange forskellen virkningen af på de positive og negative stød. Vi vil senere fitte en EGARCH-model der er et eksempel på en asymmetrisk model. Man kunne også bruge en asymmetrisk normalfordeling, men dette er afprøvet og gavner ikke modellen. Det gør i stedet for at der kommer en ekstra parameter med som skal estimeres, og derfor benytter vi i dette speciale EGARCH, da det burde kunne måle leverage effekten.

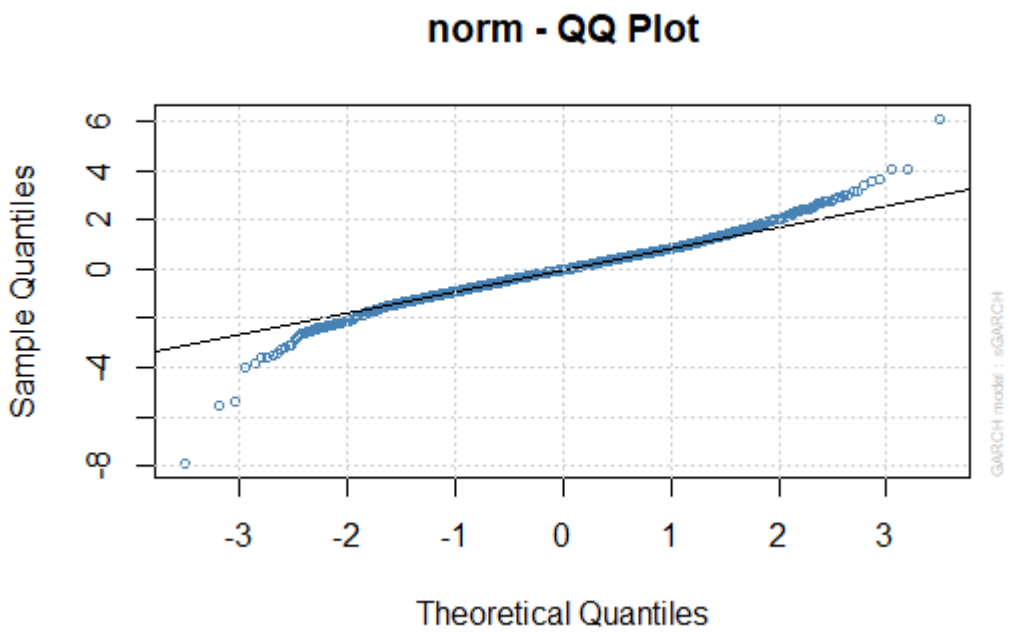
```
Sign Bias Test
-------------------------------------
                   t-value    prob sig
Sign Bias           0.8162 0.41447
Negative Sign Bias  2.2162 0.02679  **
Positive Sign Bias  0.8049 0.42098
Joint Effect        6.4754 0.09064   *
```

R output 5. Sign Bias testet for GARCH(1,1), norm.

Det sidste som ugarchfit funktionen spytter ud er *The Goodness-of-Fit*. Den sammenligner den empiriske fordeling af de standardiserede residualer med den teoretiske fordeling, som i dette tilfælde er en normalfordeling. De små p-værdier indikerer, sammen med QQ-plottet, at der ikke er tale om en normalfordeling. Det ses især tydeligt på QQ-plottet, som er Figur 12 på side 37, at der er tunge haler, og at det langt fra ligner en normalfordeling. Dette er dog ingen overraskelse, og vi må forvente, at en model hvor støjen er andet end normalfordelt er at foretrække. Derfor er vores næste skridt at kigge på, hvad der sker hvis støjen er t-fordelt.

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     65.85    4.440e-07
2    30     84.85    2.207e-07
3    40     93.07    2.581e-06
4    50    113.60    4.812e-07
```

R output 6. Goodness-of-fit test for GARCH(1,1) med normalfordelt støj.



Figur 12. QQ plottet for GARCH(1,1) med normalfordelt støj.

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(0,0,0)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.074059    0.029676   2.4956 0.012576
omega   0.101579    0.042627   2.3830 0.017173
alpha1  0.132238    0.031002   4.2655 0.000020
beta1   0.846631    0.036367  23.2803 0.000000
shape   5.475648    0.622071   8.8023 0.000000

LogLikelihood : -4011.443

Information Criteria
------------------------------------

Akaike       3.8270
Bayes        3.8405
Shibata      3.8270
Hannan-Quinn 3.8319

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1194  0.7297
Lag[2*(p+q)+(p+q)-1][2]    0.1450  0.8892
Lag[4*(p+q)+(p+q)-1][5]    1.7785  0.6716
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                   0.007522  0.9309
Lag[2*(p+q)+(p+q)-1][5]  0.988900  0.8622
Lag[4*(p+q)+(p+q)-1][9]  2.816506  0.7888
d.o.f=2

Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           0.5183 0.6043
Negative Sign Bias  1.4689 0.1420
Positive Sign Bias  1.0780 0.2811
Joint Effect        4.4589 0.2160


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     24.92      0.16332
2    30     39.20      0.09782
3    40     39.33      0.45499
4    50     51.81      0.36479
```

R output 7. Relevant output for GARCH(1,1) med t-fordelt støj.

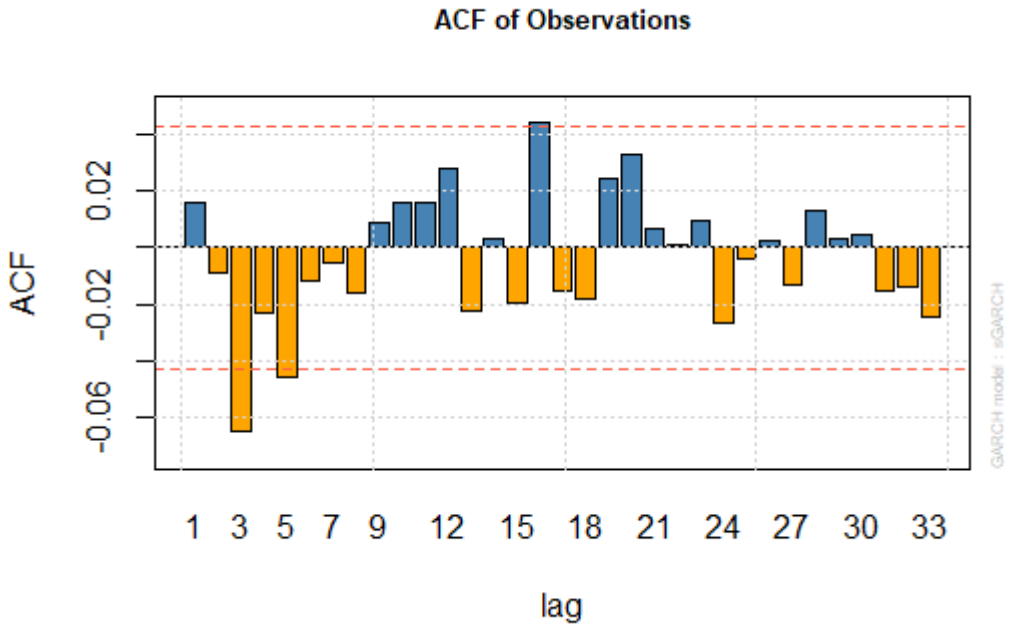
t-fordelt støj

Det væsentlige output fra Rstudio for en GARCH(1,1) med t-fordelt støj er at finde på side 38. Der kan udledes flere ting alene udfra at kigge på outputtet. Først og fremmest ses det, at der er kommet en ekstra parameter nemlig *shape* estimatet, som påvirker den generelle form på vores t-fordelingen.

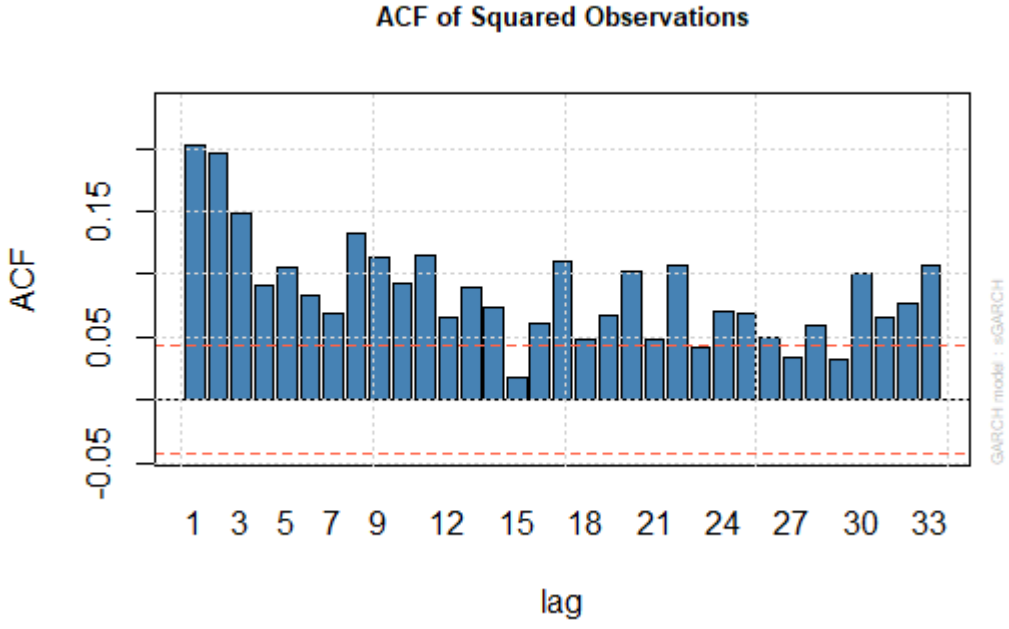
Lad os hoppe direkte til konklussionen da den ses hurtigt i vores output. Den nye model fitter vores data bedre! Det ses både på at Log-likelihooden er større, og at AIC er mindre. I denne situation er det hurtigt at komme frem til konklussionen, da det er samme GARCH model som er blevet benyttet. Hvis den nye model havde haft flere parametre med så havde vi også skulle tage højde for det.

AIC er nu 3.8270 fremfor 3.9090 hvor den var normalfordelt. Denne grund alene lader til, at vi skal vælge modellen hvor støjen er t-fordelt fremfor normalfordelt. Vi kan dog også finde andre årsager som taler for at den forrige model, med normaltfordelt støj, var decideret forkert.

Vi har som sådan ikke ændret andet i vores model end at støjen nu er t-fordelt. Dette kan ses på Ljung-Box testen og ACF plots, da de minder om hinanden og at vi ikke kan vælge den ene model fremfor en anden på den baggrund. Vi ser dog tydeligt, at modellen med t-fordelt støj lader til at fitte datasættet betydeligt bedre. Det ses i i Sign Bias testet, og Goodness of fit testen. Vi starter med at kigge på Sign Bias testen: Det ses, at vi nu har fået det ønskede resultat og at modellen nu i tilstrækkelig grad fanger leverage-effekten i vores data. Alle 3 faktorer, samt den fælles effekt er alle over de 5%, hvilket gør at vores model tilsynladende godt kan fange, hvordan volatiliteten påvirkes af ændringer såvel positive som negative. Dette er lidt overraskende, da vi intet har gjort for at modellen skal opfange leverage-effekten. Det kan tyde på at modellen før var så dårlig, at det også gav udslag i Sign Bias testet.

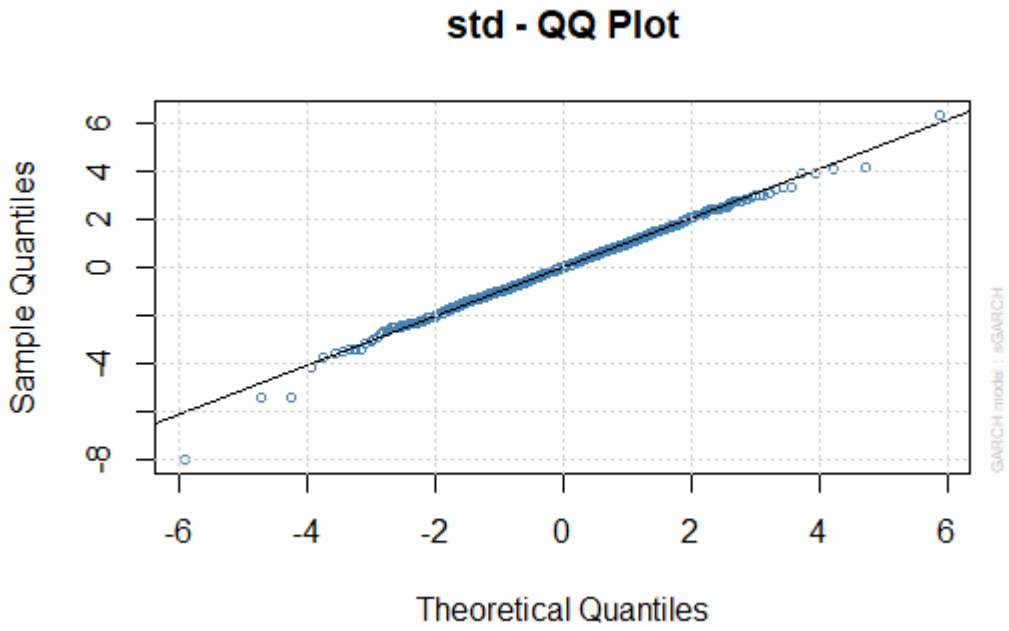


Figur 13. ACF for GARCH(1,1) med t-fordelt støj.



Figur 14. ACF for de kvadreret observationer i GARCH(1,1) med t-fordelt støj

Det sidste og muligvis vigtigste er, hvis vi ser på Goodness of fit testen. Her kan vi se, at der nu er væsentlig højere p-værdier, og at det dermed virker mere rimeligt at benytte den t-fordelte støj. Hypotesen om at den empiriske fordeling er lig med den teoretiske kan vi altså i dette tilfælde ikke afvise. Dette, og det øvrige som er omtalt i dette afsnit, gør at denne model er at foretrække over den første model, hvor støjen var normalfordelt. Det ses endnu tydeligere hvis vi kigger på nedenstående QQ-plot. Det ses, at vi har langt fladere haler end før, og at QQ-plottet generelt ser meget finere ud. Indtil videre må vi kunne konkludere at t-fordelt støj er den bedste løsning til en GARCH(1,1), vi mangler dog at se på GED-fordelt. Det væsentlige output fra Rstudio for en GARCH(1,1) med GED-fordelt støj er at finde på side 42**.**



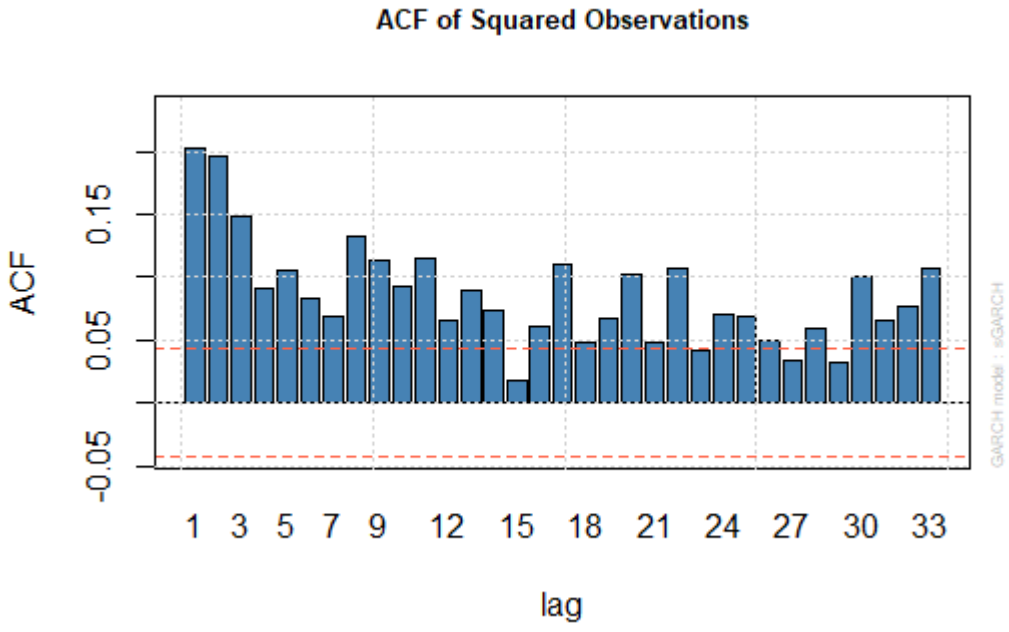
Figur 15. QQ plottet for GARCH(1,1) med t-fordelt støj.

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(0,0,0)
Distribution     : ged

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.068387    0.029698   2.3028 0.021292
omega   0.086360    0.036698   2.3533 0.018608
alpha1  0.116511    0.026719   4.3606 0.000013
beta1   0.863551    0.031926  27.0489 0.000000
shape   1.275390    0.050151  25.4309 0.000000
LogLikelihood : -4024.573

Information Criteria
------------------------------------

Akaike       3.8395
Bayes        3.8530
Shibata      3.8395
Hannan-Quinn 3.8444

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1295  0.7189
Lag[2*(p+q)+(p+q)-1][2]    0.1515  0.8849
Lag[4*(p+q)+(p+q)-1][5]    1.8311  0.6588
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1392  0.7091
Lag[2*(p+q)+(p+q)-1][5]    0.9458  0.8718
Lag[4*(p+q)+(p+q)-1][9]    2.6146  0.8205
d.o.f=2

Sign Bias Test
------------------------------------
                   t-value    prob sig
Sign Bias           0.5863 0.55774
Negative Sign Bias  1.7999 0.07202   *
Positive Sign Bias  0.9217 0.35679
Joint Effect        5.2637 0.15348


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     33.51      0.02097
2    30     43.41      0.04174
3    40     51.87      0.08136
4    50     59.67      0.14128
```

R output 8. Output for GARCH(1,1) med t-fordelt støj.

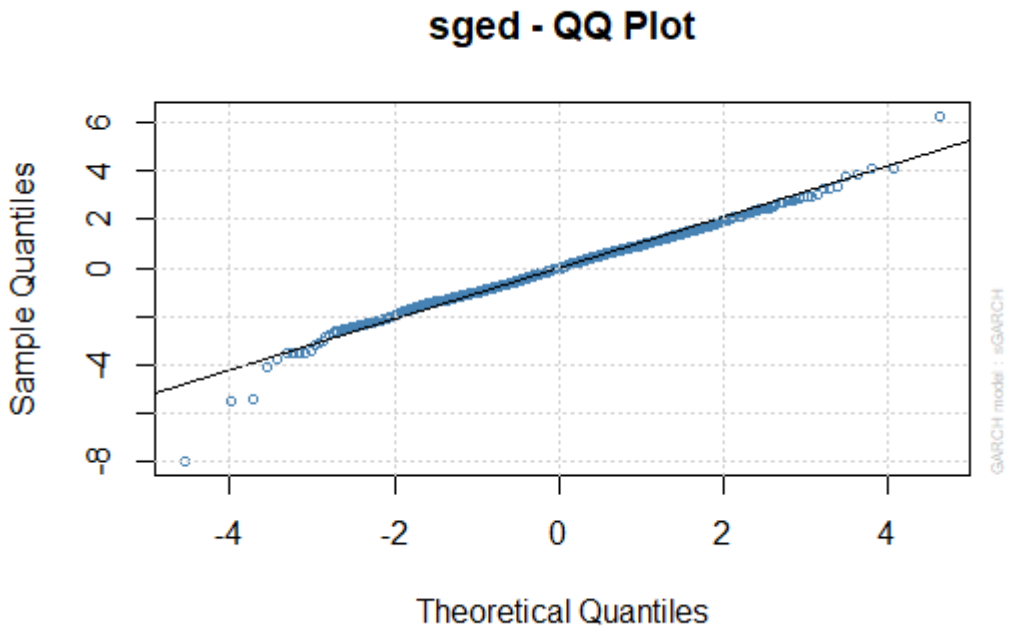
Nu skal vi sammenligne GARCH(1,1) med GED-fordelt støj, med vores hidtil bedste model, som er en GARCH(1,1) med t-fordelt støj. Hvis vi udelukkende, kigger på outputtet som vi får ud af Rstudio, så ses det, at der umiddelbart ikke er den store forskel på de to modeller hvis man kigger på første del af outputtet. De estimerede parametre er alle signifikante, dog er AIC for t-fordelingen bedre, 3.8270 mod 3.8395. Det samme gør sig gældende for Log-likelihooden hvor den nye model også er lidt dårligere end den forrige.

Resultatet af Ljung-Box testen for de 3 modeller indtil videre er forsat ret ens, og er allerede blevet forklaret og vil i dette tilfælde ikke bliver gennemgået nærmere. Sign Bias testet er, for den nye model, over 5% for alle 3 faktorer, ligesom den fælles effekt også er over 5-procent grænsen. Det ses dog, at virkningen af negative stød blot er på syv procent, og derfor forklarer modellen med t-fordelt støj leverage-effekt bedre end hvis støjen er GED-fordelt. Den næstsidste ting som taler for, at den nye model ikke er bedre end den med t-fordelt støj, er hvis vi kigger på Goodness-of-fit testet, hvor vi ser at p-værdierne er betydelig mindre end før.



Figur 16. GARCH(1,1) med GED-fordelt støj.

Alle foreløbige tests peger på at GARCH(1,1) med t-fordelt støj passer vores data bedst, og nedenstående QQ-plot kan ikke overbevise os om andet end at t-fordelingen stadigvæk er vores bedste løsning til at beskrive datasættet. Dette må være det sidste som vi behøver for at konkludere, at vores bedste model af de tre gennemgået uden tvivl ville være GARCH(1,1) med t-fordelt støj. Det er noget bedre end hvis støjen er GED-fordelt, og begge modeller er modellen med normalfordelt støj overlegn. Modellen med normalfordelt støj er en helt horribel model, og kan slet ikke beskrive vores datasæt i sådan en grad, som vi ønsker. Modellen er ubrugelig, og det må formodes, at det skyldes at vores støj bestemt ikke er normalfordelt. Vi vil nu se på at AIC langt fra er det eneste som afgører hvilken model er den mest brugbare.



Figur 17. QQ plottet for en GARCH(1,1) med GED-fordelt støj.

7.3 Yderligere modelsammenligning.

Model	**AIC**
GARCH(1,1), norm	3.9090
GARCH(2,1), norm	3.9100
GARCH(1,2), norm	3.9085
GARCH(2,2), norm	3.9095
GARCH(1,3), norm	3.9063
GARCH(2,3), norm	3.9066
GARCH(3,3), norm	3.9082
GARCH(3,1), norm	3.9109
GARCH(3,2), norm	3.9104
GARCH(1,4), norm	3.9040
GARCH(2,4), norm	3.9008
GARCH(3,4), norm	3.9017
GARCH(4,4), norm	3.9025
GARCH(4,1), norm	3.9119
GARCH(4,2), norm	3.9114
GARCH(4,3), norm	3.9086
GARCH(5,5), norm	3.9043

Tabel 1. AIC for forskellige GARCH modeller med normalfordelt støj.

Ovenstående tabel viser AIC for nogle forskellige GARCH modeller med normalfordelt støj. Det ses at der generelt ikke er den store forskel på dem. Det ses, at GARCH(2,4) har den laveste AIC af dem alle, men hvis vi kigger på de estimerede parameter er de alle ikke signifikante, og derfor ikke estimeret lige så præcis som ved GARCH(1,1) modellen. Det var dog også forventet da vi tidligere har nævnt, at med færre parameter at estimere, må vi også forvente en mere præcis estimation. Grovt sagt er både $\widehat{\beta_1}$ og $\widehat{\beta_3}$ lig 0, og har ingen indflydelse på processen. Dette betyder dog ikke nødvendigvis at modellen er ubrugelig, da denne stadig er forskellig fra for eksempel GARCH (1,3). I vores tilfælde er det dog ikke brugbart, da parameterne som er estimeret ikke er signifikante, og vi får derfor ingen gavn af at tage de ekstra parameter med. Dette kan man også se på GARCH(1,2), hvor AIC ligeledes er lavere end for GARCH(1,1), men her er der også en parameter som ikke er signifikant. I dette tilfælde er den dog meget tæt på grænsen på 5% (er på knap 8 procent), og måske kunne denne model være et bedre fit til vores data. Det er dog ikke værd at undersøge endnu, da vi stadig mangler mange modeller at gennemgå.

Tabel 1 er en indikation på, at AIC langt fra fortæller hele billedet. Det er naturligvis at foretrække at ens parametre bliver estimeret præcist, som også er gældende for størstedelen af modellerne i Tabel 2, men det betyder ikke at en model med signifikante estimater nødvendigvis er en bedre model end en model hvor en, eller to parametre ikke er signifikante. En model hvor ikke alle parametre er signifikante kan sagtens beskrive datasættet glimrende. Det handler om at kunne finde en model som beskriver datasættet så godt som muligt, og ikke hvor signifikante parametrene er da disse to nødvendigvis ikke hænger sammen. Tabel 2 indeholder modeller som vi nu skal kigge nærmere på. Det er en af disse modeller som vi formoder kan beskrive datasættet bedst muligt. Vi kunne godt have haft mere komplekse modeller med, men der er ingen grund til en mere kompliceret og kompleks model, hvis det ikke gavner i forhold til at kunne beskrive datasættet. I Tabel 2 så har de fleste modeller parametre som er estimeret signifikant, dog er de modeller markeret med * ikke.

Model	**AIC**
GARCH(1,1), std	3.8270
ARMA(1,1)+GARCH(1,1), std	3.8239
ARMA(1,1)+GARCH(1,1), ged	3.8367
ARCH(1), std	3.8897
ARCH(2), std	3.8604
ARCH(3), std	3.8534
ARCH(4), std	3.8512
*EGARCH(1,1), std	3.8108
*EGARCH(1,3), std	3.8044
*EGARCH(2,1), std	3.8052
EGARCH(1,3), ged	3.8153

Tabel 2. Relevante modeller.

Tabel to indeholder som nævnt de modeller som passede datasættet bedst. Alt relevant output fra Rstudio for disse modeller kan findes i bilager under Bilag 2 fra side 69**.** De første to sider, og modeller er ARMA-GARCH modellerne. Derefter følger fire sider med ARCH, og slutter af med de 4 valgte EGARCH modeller.

Adskillig litteratur stiller spørgsmålstegn ved, om noget er bedre end en GARCH(1,1)[3], og vi gør nu det samme. Det er op til ovenstående modeller at forbedre vores, på nuværende tidspunkt bedste model som er GARCH(1,1) med t-fordelt støj. De modeller som er markeret med * har, som tidligere nævnt, ikke parametre som er estimeret signifikant, men jeg synes alligevel, at de er relevante at have med.

Hvis vi starter med at kigge på AIC alene og husker at vores hidtil bedste model har en AIC på 3.8270, så det ikke overraskende, at ARCH modellerne ikke kan forbedre AIC'en. Som tidligere nævnt, så ville vi forvente at GARCH ville kunne beskrive datasættet bedre, og det lader til at være tilfældet her. Vi sammenligner dog alligevel hurtigt ARCH(4,0), som er vores bedste ARCH model af de fire skrevet ind i Tabel 2, med GARCH(1,1), for at vise hvor meget vi mener, at der skal til før at den nye model slår den hidtil bedste.

Vi starter med at kigge på estimationen af parametrene. Her er alle signifikante, dog er $\widehat{\alpha_4}$ for ARCH(4) under 0,5% fra at ryge over vores grænse på fem procent. Dette er dog ikke den store overaskelse da vi nu har flere parameter at estimere og vi derfor forventede en dårligere estimation. På det grundlag ville vi ikke kunne foretrække den ene model fremfor en anden, men det kan vi tilgengæld hvis vi kigger på AIC og Log-likelihooden. Her taler begge to til GARCH(1,1) fordel. Der er ikke nogen nævneværdig forskel på Ljung-Box testet eller på de plottede residualer, både standard og kvadrerede.

Så har vi tilbage at sammenligne Sign Bias testet og Goodness-of-fit testet og vi starter med først nævnte. Det ses her, at ARCH(4) har alle 4 signifikante, dog er den ene på knap 8% hvilket gør at GARCH at foretrække. Det der dog taler for ARCH er, at både virkningen af fortegnet samt den negative virkning har meget signifikante (begge over 95 procent), og dermed ville det være svært, på baggrund af denne test, at foretrække den ene fremfor den anden. Vi kigger derfor på den sidste test, Goodness-of-fit, og her ser vi at ARCH(4) har væsentlige højere p-værdier.

[3] Blandt andet Hansen & Lunde i 2001. https://www.bauer.uh.edu/rsusmel/phd/HansenLunde_Garch.pdf

Man kan argumentere for begge modeller, men jeg villr stadig vælge GARCH(1,1) modellen over ARCH(4) modellen i hvert fald til vores datasæt. For at vi skal foretrække en model over GARCH(1,1) mener jeg, at der skal være mere overbevisende beviser. AIC og Log-likelihooden taler stadig for GARCH, ligesom alt teorien også gør. Derfor er GARCH(1,1) stadigvæk vores bedste model, men lad os se om ARMA(1,1)-GARCH(1,1) med t-fordelt støj kan lave om på det.

Først skal det nævnes, at ARMA(1,1)-GARCH(1,1) med t-fordelt støj er bedre end hvis støjen er GED-fordelt. Det gælder både AIC og Loglikelihooden, men også de øvrige test taler klart for t-fordelt støj. Det ses ***i*** Bilag 2, at såvel Sign Bias testet og goodness-of-fit taler klart for støjen er t-fordelt. Derfor skal vi nu se på om vores bedste ARMA-GARCH model er bedre end GARCH(1,1). Estimationen for de to modeller er meget ens, udover at ARMA-GARCH naturligvis har to ekstra parametre. GARCH modellen har marginal dårligere AIC og Loglikelihood, og dette er ikke noget man nødvendigvis skal lægge for meget i. Vi har tidligere set, at flere parametre kan give en bedre AIC selvom modellen fitter datasættet dårligere. ARMA-GARCH har generelt højere p-værdier for goodness-of-fit, hvilket taler for modellen, og fra Sign Bias testet ser det ud som om at denne model bedre fanger virkningen på negative og positive stød. Til gengæld klarer den sig væsentlig dårligere end GARCH på fortegnets virkning og den fælles virkning. Det er ikke overraskende for os, at der er ligheder mellem de to modeller. Som nævnt i teorien, side 18, så modellerer denne model både middelværdien og variansen. Her er støjvariablen ikke uafhængig, men derimod givet GARCH-processen. Derfor var det forventet, at de to modeller ville have ligheder. Jeg mener ikke, at der er beviser nok til at fravælge GARCH(1,1) som vores bedste model. Det man gavner ved at få de ekstra parametre er ganske enkelt for lidt.

Som I kan se i bilaget fra side 75, er den eneste EGARCH model som har signifikante estimater er EGARCH(1,3) med GED-fordelt støj, hvor de andre EGARCH modeller ikke kan estimere $\hat{\mu}$ signifikant. De tre EGARCH modeller med t-fordelt støj har hver deres fordele, men er generelt ret ens. Deres AIC og Loglikelihood er tæt på hinandens ligesom Ljung-Box testet er. Derfor må det være op til Goodness-of-fit samt Sign Bias testet til at afgøre hvilken af de tre modeller som er at foretrække. For alle 3 modeller er Sign Bias testet, som forventet,

Alle tre modeller klarer, som forventet, Sign Bias testet uden problemer og de varrierer indbyrdes imellem om hvilken af faktorerne som er mest signifikant. Hvis vi, for eksempel, sammenligner EGARCH(1,1) og EGARCH(1,3) ses det, at førstnævnte klarer virkningen af fortegnet samt de positive stød bedre, hvor sidstnævnte klarer den fælles virkning samt virkningen af negative stød bedst. Det betyder, at det gør det meget vanskeligt at vælge den ene model over den anden alene udfra virkningen af de fire faktorer. Samme argument kan bruges for Goodness-of-fit testet hvor disse heller ikke giver et klart billede af at den ene model er de andre overlegne. Det betyder, alt taget i betragtning, at vi vil vælge EGARCH(1,1) over de andre da vi ikke får nogen nævneværdig gavn af at få ekstra parametre med.

Vi skal nu afgøre om EGARCH(1,3) med GED-fordelt støj, kan fitte vores data bedre end vores EGARCH(1,3) med t-fordelt støj. Der er valgt samme model med forskellige støj for at vurdere modeller med samme antal parametre. Som allerede nævnt så taler det for at det er tilfældet, at samtlige parameter er signifikante, hvorimod det taler imod, at vi tidligere så at t-fordelt støj lod til at passe vores datasæt bedst. Det ses også hurtigt, at de to modeller minder meget om hinanden i hvert fald indtil vi kommer ned til den sidste test som er Godness-of-fit. Indtil da er modeller meget ens, men her er der en meget stor forskel. Hvis støjen er GED-fordelt er samtlige p-værdier under 5% hvilket indikerer, at modellen slet ikke fitter datasættet. Hvis støjen derimod er t-fordelt så ses det at vores model fitter data hvor den mindste p-værdi er på 11 procent. Dette betyder at vi til enhver tid ville vælge modellen hvor støjen er t-fordelt ligesom også var tilfældet for en GARCH(1,1).

Valget står nu imellem GARCH(1,1) eller EGARCH(1,1), begge med t-fordelt støj og her er der argumenter som taler for begge modeller. Det som taler for GARCH(1,1) er, at samtlige estimater er signifikante, og at der er en færre. Derudover ikke er nogen betydelig forskel på de to modellers AIC og Loglikelihood og deres Ljung-box tester gør heller ikke, at vi ville foretække EGARCH over vores standard GARCH. Det som taler for EGARCH er, ikke overraskende, at Sign Bias testen hvor den er betydelig bedre end GARCH, men dette var i den grad forvente da den skulle fange asymmetrien. Goodness-of-fit taler også for EGARCH da dennes p-værdier er højere.

Der synes ikke, at være nok til at vælge den ene fremfor den anden. Hver model har sine fordele, og ulemper, og dette gør at man ikke med sikkerhed kan sige at den ene model er den anden overlegen. I min problemformulering lød det, blandt andet, at jeg ønskede, at undersøge fordele og ulemper ved brug af GARCH modeller til beskrivelse af finansielle data. Det mener jeg har fundet sted nu, og det lader til, ihvert fald i vores datasæt, at GARCH modeller er bedre end de øvrige. Det er dog svært, at sige om det er GARCH(1,1) eller EGARCH med samme orden som er den bedste. Det dog klart, at støjvariable skal være t-fordelt og Goodness-of-Fit for EGARCH i forhold til GARCH kan tyde på at en asymmetrisk model er det bedste fit.

Kapitel 8 – Forecasting.

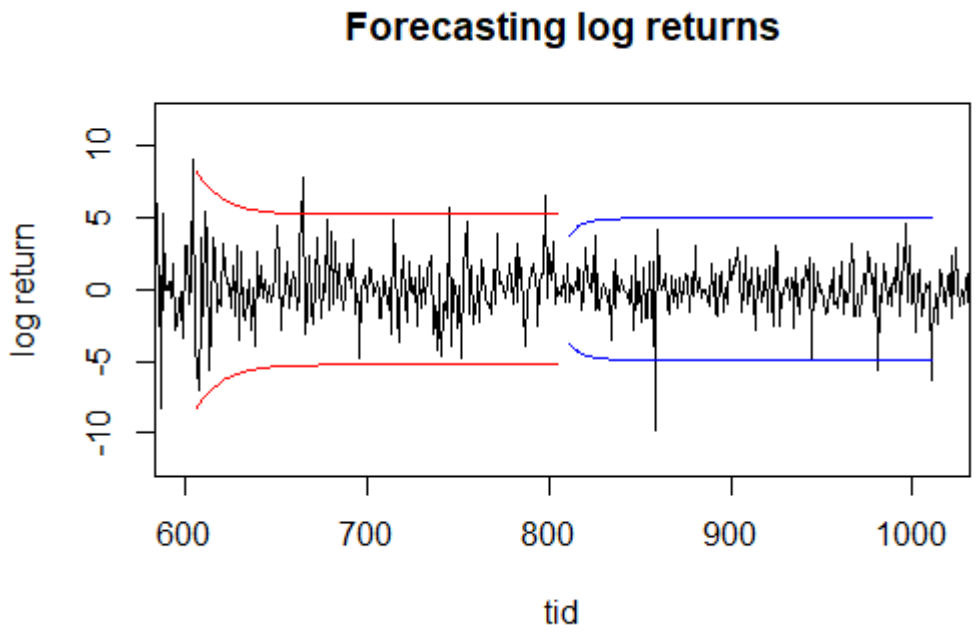
8.1 Indledning

I dette kapitel skal vi se nærmere på forecastingen af forskellige modeller, og se hvad modellerne betyder for restultatet af forecastingen. Vi skal derudover, se hvordan man kan anvende simulation som en forecasting metode.

8.2 Forecasting

Vi starter med at kigge på forecastingen for en GARCH(1,1) med t-fordelt støj, som vi i forrige kapitel så var en af vores bedste modeller. Det er en forecasting hvor vi forecaster 200 observationer fremad, og hvor vi forecaster udfra to forskellige mængder observationer. Det ses tydeligt på nedenstående plot, at det har en betydning for vores forecast, hvor meget af datasættet vi har med som grundlag for forecastingen. Det ses, på den røde linje, at hvis vi starter vores forecasting fra observation 606, altså at vi har brugt de første 605 observationer til at forecaste de næste 200, hvilket er illustreret med det røde interval på ***Figur X***. Denne figur indeholder også en anden forecasting, hvor vi har benyttet de første 810 observationer som grundlaget for at forecaste de næste 200, som er det blå interval. Intervalerne er begge to 95 procent intervaller, og det ses at de begge to har størstedelen af fremtidige værdier indenfor deres intervaller, men ikke alle værdier. Det betyder dog langt fra at forecastingen er mislykkedes, da vi kan se at det lader til, at det er rimeligt sandsynligt at det som ligger udenfor vores konfidensgrænser er de resterende 5 procent.

Vi ved, at den forventede middelværdi altid er 0 hvilket resulterer i at intervallerne naturligt nok ligger omkring 0. Det som man gør ved en forecasting, ved hjælp af en GARCH-model, er, at man forecaster volatiliteten som er bredden på intervallet.



Figur 18. Forecastingen af 200 observationer for en GARCH(1,1) med t-fordelt støj.

Hvis vi kigger nærmere på de to forskellige forecastninger beskrevet for oven, ses det at de begge to konvergerer mod en værdi, nemlig usikkerheden hvor man benytter den ubetingede varians. De to grænser er ikke helt identiske. Det skyldes, at forecastingerne er baseret på to forskellige datasæt, og at der for hver forecasting er fittet en model til det data, som forecastingen er baseret på. Det betyder altså, at hver forecasting er forskellig da hver model har forskellige parametre og dermed bliver den ubetingede varians i hver af de to fittede modeller naturligvis også forskellig. Vi husker tilbage på teori afsnittet for forecasting, hvor vi nævnte at hvis forecastingen går mod uendelig, så vil vores forecast gå mod den ubetingede varians $\frac{\omega}{1-\alpha_1-\beta_1}$. Det ses tydeligt, at den ubetingede varians afhænger af modelparametrene. Det resulterer i to forskellige værdier for den ubetingede varians som forecastingen konvergerer imod. Den første forecasting, lavet på de første 610 observationer i datasættet, konvergerer mod en ubetinget varians, som er 2,620, hvorimod den anden forecasting, lavet på de første 810 observationer, konvergerer mod en ubetinget varians på 2,468. Det er også værd, at bemærke at den første forecasting er længere tid om at ramme den betingede varians, da den skal bruge 108 forecastede observationer for at nå den værdi, til sammenligning skal den anden forecasting blot bruge 50 forecastede observationer for at ramme den betingede varians. Det er også værd at bemærke, hvordan de to forskellige forecastinger starter. Den første starter højt og går langsomt ned mod sin betingede varians, og

den anden starter lavt og konvergerer hurtigere mod værdien. Som skrevet tidligere, så er det modelparametrene, som afgører hvor hurtigt, eller hvor langsomt, processen går mod den ubetingede varians.

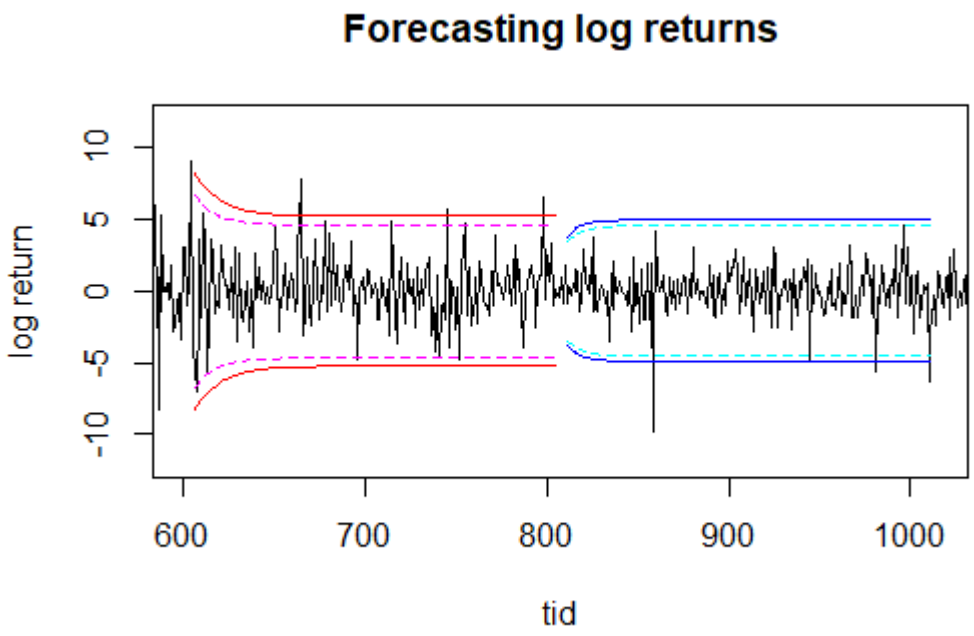
Lad os nu prøve at sammenligne GARCH(1,1) forecastingen med en EGARCH(1,1) hvor begge har t-fordelt støj. Det ses på Figur 19, at der er en forskel på de to modellers forecasting på trods af, at de forecaster udfra samme datamængde. Ligesom før så er henholdvis den røde og den blå linje forecastingen for GARCH(1,1), men her har vi også intervallerne for EGARCH modellen i plottet, som er de halvstiplede linjer.

Det ses her, at GARCH modellens forecasting intervaller er større end den anden models. Derudover går begge dens konfidensgrænser langsommer mod den ubetingede varians for første forecasting, og hurtigere for den anden forecasting. Ved den første forecasting begyndelse er volatiliteten stor, og det ses, at GARCH på grund af den varians har behov for størreintervaller for at få sit konfidensinterval. Ved begyndelsen af den anden forecasting er der betydelig mindre volatilitet, og det udnytter EGARCH til at kunne nøjes med et smalt interval til at forudsige, hvor afkastet ender.

Det kan indikere, at EGARCH modellerer volatiliteten bedst af de to modeller. Forskellen imellem de to modeller skyldes, at der naturligvis er forskel på de to modeller. Der er blevet estimeret forskellige parametre, udfra deres model, som anvendes til at finde konfidensgrænserne. Vi ser, at begge modellers forecasting har nogenlunde den samme mængde data indenfor deres intervaller, hvilket indikerer at EGARCH modellen forecasting er bedre. Det skyldes formentligt, at EGARCH bedre fanger asymmetrien i vores datasæt, og derfor ikke har behov for et lige så bredt interval som GARCH har det.

Da begge modeller cirka har den samme datamængde indenfor deres intervaller, må det siges at være en fordel at benytte EGARCH eftersom intervalet er mindre. Det medfører, at vi med samme sikkerhed, nemlig 95%, kan sige hvor hvor log-afkastet ender henne, men at vi med EGARCH får et smallere interval. Det ses dog også, at det smallere interval ved EGARCH gør at den efter en længere periode undgår at få observationer med som GARCH gør.

Konklussionen for de to modeller er efter alt dette rimelig klar. EGARCH har eftersom den bedre fanger asymmetrien ikke samme behov som GARCH for et lige så bredt konfidensinterval. Dette kommer klart EGARCH til gode i første del af forecastingen hvor den med samme sikkerhed kan placere vores afkast indenfor et smallere interval. EGARCH må siges, at være den bedste model til at forecaste hvis det handler om en kortere periode. Det ses dog også, at GARCH over en længere tidshorisont får flere observationer med end EGARCH. Derfor kan man argumentere for GARCH over EGARCH såfremt tidshorisonten er længere, men der man skal naturligvis tage højde for at den i starten af forecastingen har behov for et bredere interval for at opnå samme sikkerhed som en EGARCH. Alt dette taget i betragtning gør, at man må antage at EGARCH er den bedste model og at den klarer sig dårligere på en længere forecasting horisont kan skyldes, at modellen har problemer med at estimere middelværdien signifikant.



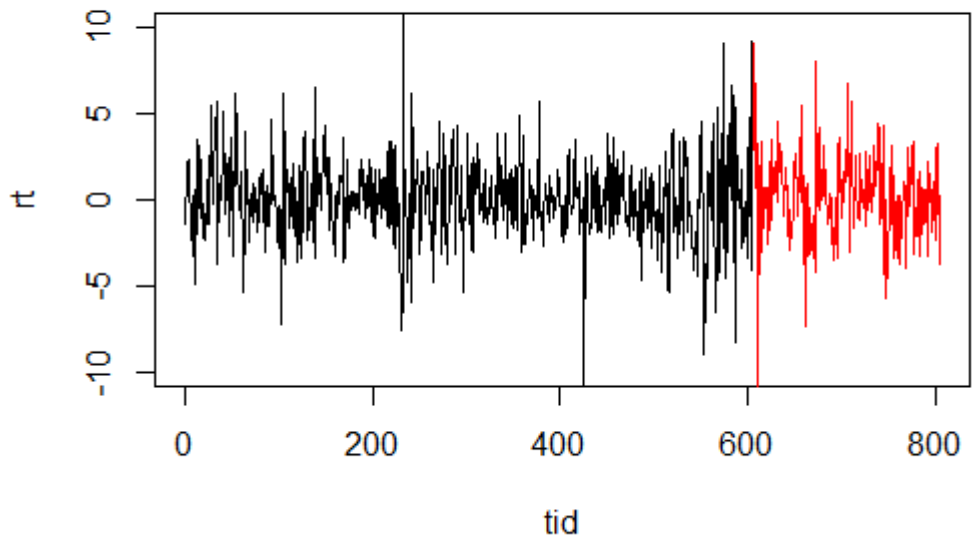
Figur 19. Forecasting af 200 observationer hvor både GARCH(1,1) og EGARCH(1,1) intervaller er at finde.

8.3 Simulation

Som nævnt i indledningen skal vi også kigge på, hvordan simulation kan benyttes som en form for forecasting. Figur 19 viser en simulation af en GARCH(1,1) model med t-fordelt støj. Det sorte er logafkastet fra vores datasæt, hvor det røde er simulationen af 200 observationer. Simulationen starter fra samme sted som vores første forecasting gjorde. Simulationen vil naturligvis varierer fra simulation til simulation, men hver enkelte simulation giver et bud på, hvordan hvordan det kunne se ud. Til gengæld har man et godt bud på usikkerheden ude i fremtiden såfremt man simulerer mange processer, da man på denne måde kan vurdere hvor simulationerne oftest befinder sig.

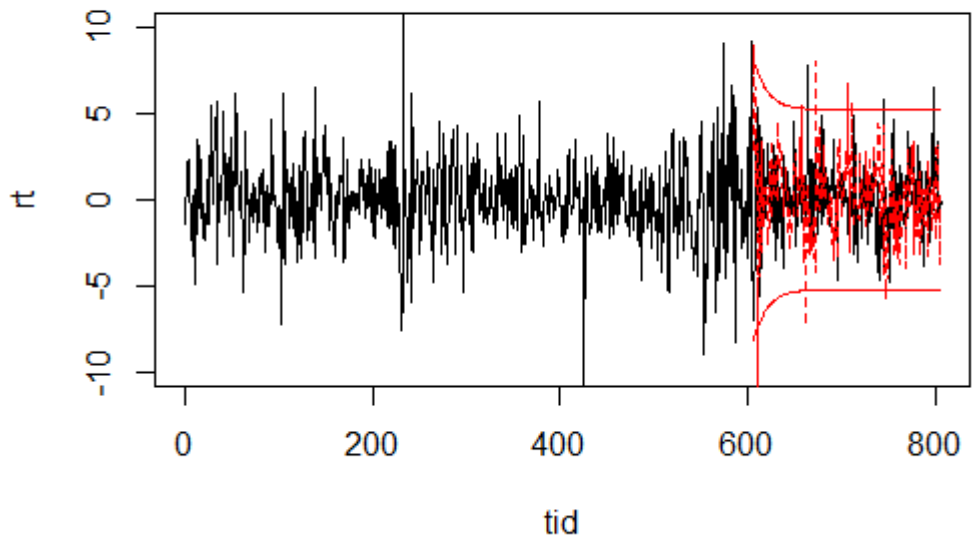
Simulationen fungerer udfra den model som er valgt, i dette tilfælde en GARCH(1,1) med t-fordelt støj, og denne models parametre som er fittet ved hjælp af datasættet, som i dette tilfælde er de første 605 observationer. Simulationen regner videre udfra de observerede rt-værdier og de estimerede sigma værdier der er set frem til lige inden simulationen starter. Vi ved, for en GARCH(1,1) proces, at man skal benytte parametrene i modellen, ligesom man også skal kende den forrige sigma og rt værdi. De indgår sammen med et nye epsilon i definitionen af, hvordan den næste rt-værdi og næste sigma-værdi skal regnes. Fordelen ved forecastingen er, at den giver faste grænser for hvor vi forventer at finde kurven i fremtiden. Fordelen ved simulation er, at man får et indblik i hvordan kurven kunne forløbe. Ved hjælp af simulationen kan vi altså se, hvor meget den går op og ned, eller hvor sandsynligt det er, at den er flad i er periode. Med simulationen man får hele fordelingen af intervallet med, hvor man ved en forecasting får konfidensgrænserne. Med andre ord kan man altså sige, at forecastingen viser hvor det simulerede er, hvor simulationen viser fordelingen.

I dette tilfælde bruger, såvel forecastingen som simulationen, samme datasæt og GARCH(1,1) model til at estimere parametrene. Disse parametre ændres naturligvis ikke under simulationen, da det er dem som påvirker de simulerede rt-værdier og sigma-værdier. Derudover går vores usikkerhed for simulationen imod den ubetingede usikkerhed med samme hastighed som ved vores forecasting. Det er dog væsentlig sværere at se i simulationen sammenlignet med forecasting. Vi vil formentlig ikke kunne se noget ekstra om forskellen på GARCH og EGARCH ud fra simulationen, som vi ikke allerede så i forecastingen.



Figur 20. Simulation af 200 observationer for en GARCH(1,1) proces.

Det kan være svært at fortolke Figur 19, da den blot viser én af mange mulige simulationer. Umiddelbart er det ikke usandsynligt, at afkastet kunne se ud som simulationen viser, men meget mere kan vi ikke sige udfra denne figur. Hvis vi dog ændrer lidt, så vil vi kunne sige meget mere om simulationen og den valgte model anvendt til at simulere. Vi plotter nu afkastet fra samme periode som før, men tilføjer også de 200 observationer hvor vi har simuleret. Derudover tilføjer vi et 95 procent konfidensinterval fra vores forecasting af samme GARCH(1,1) model. Vi ved, at både forecastingen som simulationen benytter den rigtige betingede fordeling, og det kan vi udnytte.

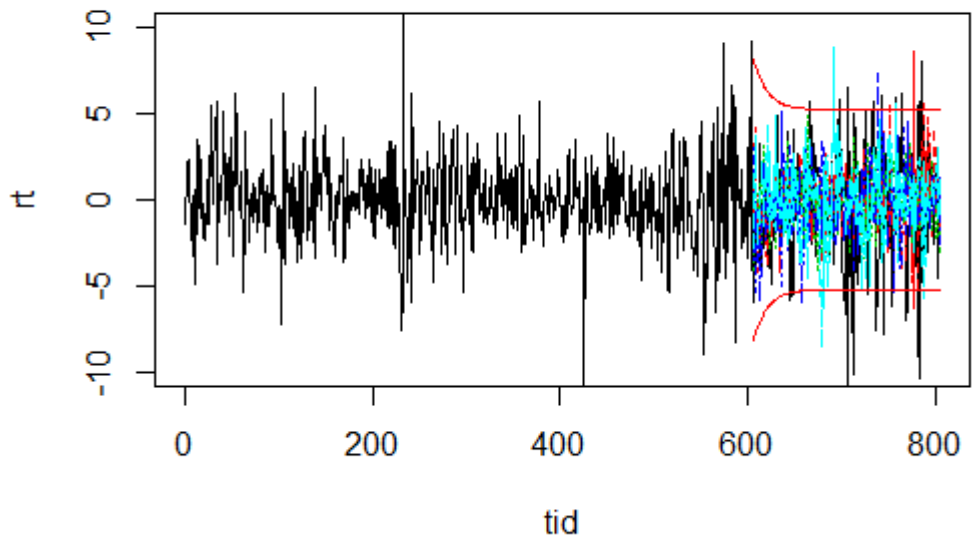


Figur 21. Indeholder det faktiske logafkast sammen med simulationen og konfidensgrænser for GARCH(1,1).

Ovenstående figur kan vi udlede langt mere fra, end vi kunne fra Figur 19. Ved sidstnævne kunne vi ikke udlede så meget, men set med det blotte øje så var det ikke usandsynligt at afkastet kunne se sådan ud. At den kunne se sådan ud bliver bakket op af Figur 20. Det ses, at den simulerede kurve har nogenlunde samme form som det faktiske afkast for den simulerede periode. Det betyder, at vores GARCH(1,1) med t-fordelt støj, tilsynladende er brugbar. Med brugbar menes der, at den i hvert fald ikke er helt forkert. Det ses, at den model lavet udfra de første 605 observationer også kan anvendes på den simulerede del.

Yderligere ses det også, at både den simulerede og den faktiske kurve med rimelighed holder sig inden for forecastingens grænser. Det er langt fra overraskende, at det er tilfældet for den simulerede eftersom dens fordeling jo er den samme som forecastingens. Som sagt så benytter både forecastingen og simulationen den rigtige betingede fordeling, som i dette tilfælde er givet de første 605 observationer. Derfor var det forventet, at simulationen ville holde sig inden for grænsen med rimelig sandsynlighed, og det faktum at det faktiske afkast også gør det taler for, at modellen er brugbar. Det lader til, at modellen passer eftersom at data efter de første 605 observationer opfører sig efter samme model og de samme parametre som det gjorde frem til observation 605.

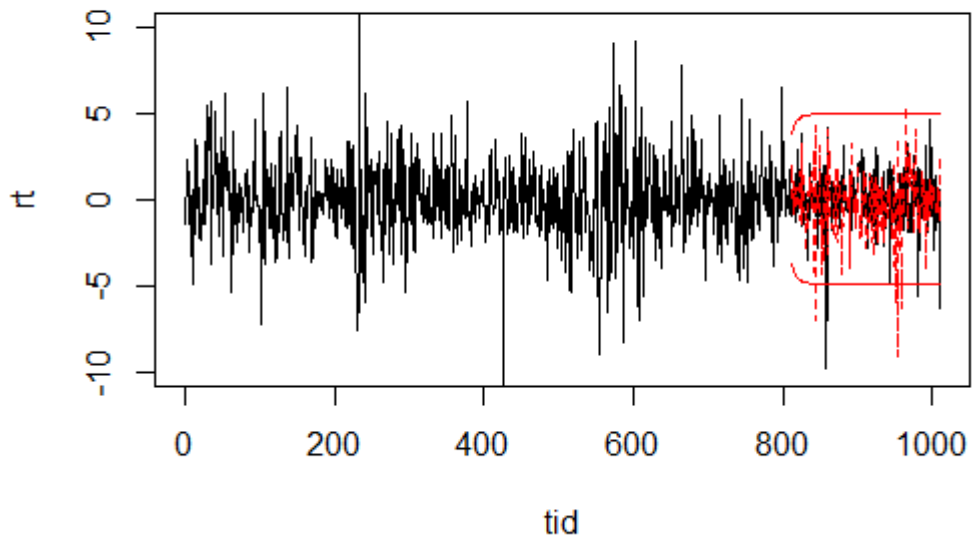
Det er klart, at eftersom simulationen ikke er ens hver gang, at man kan være heldig og ramme en simulation som tilfældigvis passer fint med det faktiske afkast. En måde at sikre sig, at modellen passer nogenlunde er ved at lave flere simulationer samtidig. Resultatet derfra kan ses i Figur 21. En enkelt simulation siger som sagt ikke meget om sandsynligheden for hvor kurven kan findes i fremtiden. Ved at simulere flere kurver på samme tid, kan vi opnå en anden form for forecasting. I Figur 22 er der blevet simuleret 5 forskellige kurver udfra vores GARCH(1,1) model, og vi har endnu engang sat konfidensgrænserne fra vores forecasting på. Lige netop ved denne simulation, har vi valgt at simulere fem forskellige kurver som alle er lige sandsynlige at finde sted. Vi kunne sagtens haft valgt flere simulationer, men det kunne gøre det sværere at se hvad der faktisk sker. Det er måske heller ikke helt tydeligt, hvad der sker med hver enkelt kurve i Figur 21, men vi kan trods alt se noget. Vi kan blandt andet se, at største delen af de 5 kurver befinder inden for forecastingens grænser. Dette er naturligvis ikke overraskende, men ved hjælp af flere kurver kan vi endnu bedre få en fornemmelse for hvor det er sandsynligt at den faktiske kurve befinder sig. Grunden til, at man kan anvende simulation, som en form for forecasting er følgende. Såfremt man simulerer rigtig mange processer, så vil man have et godt bud på usikkerheden ude i fremtiden. Man kan se på hvor stor en andel af kurverne som når langt ud. Des større andelen er des mere usikkerhed må man forvente der er. Vi ser, at de forskellige simulationer i dette tilfælde passer meget godt med det faktiske afkast, hvilket igen indikerer at vores model er brugbar.



Figur 22. Flere simulerede kurver for en GARCH(1,1).

Vi kan gøre det samme, for den anden forecasting periode hvor vores simulation bygger på de første 810 observationer. Resultatet bliver mere eller mindre det samme og fremgangsmåden er ens. Det er dog værd at bemærke, at når vi rykker tidspunktet for simulationen, så vil det kun være modelparameterene som ændres, ligesom det var tilfældet i forecastingen. Eftersom vi nu har mere data, må vores parametre blive estimeret mere sikkert. Eftershånden som man kommer længere fra sit starttidspunkt, så vil simulationen nærme sig den ubetinget usikkerhed, ligesom forecastingen nærmer sig sin ubetinget varians. Det er dog væsentlig nemmere, at se hastigheden hvorpå den nærmer sig under forecastingen fremfor når der bliver simuleret.

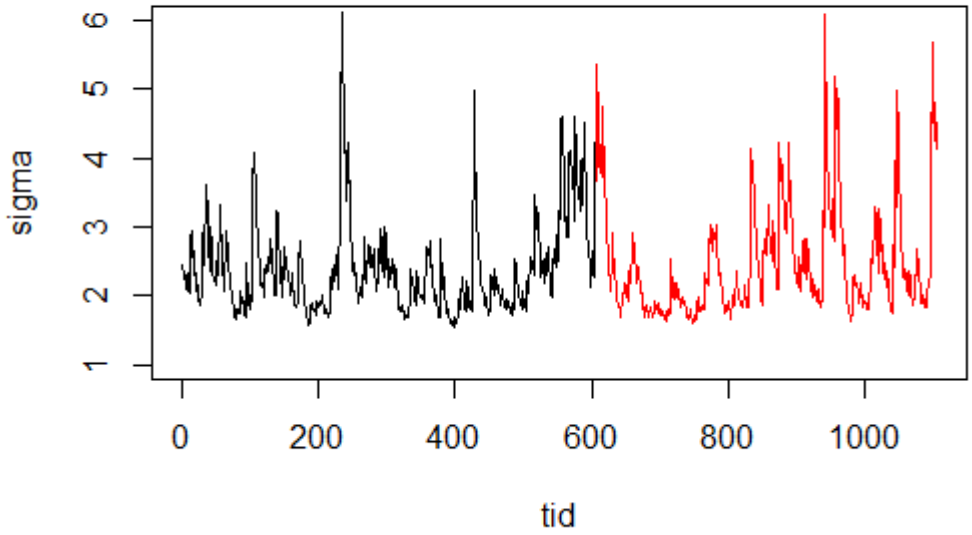
Som man kan se, så er der ikke nogen betydelig forskel på Figur 21 og Figur 23. Derfor er konklussion for den nye simulation den samme som for den første. Det lader til, at modellen passer eftersom at data efter de første 810 observationer opfører sig efter samme model og de samme parametre som det gjorde frem til observation 810.



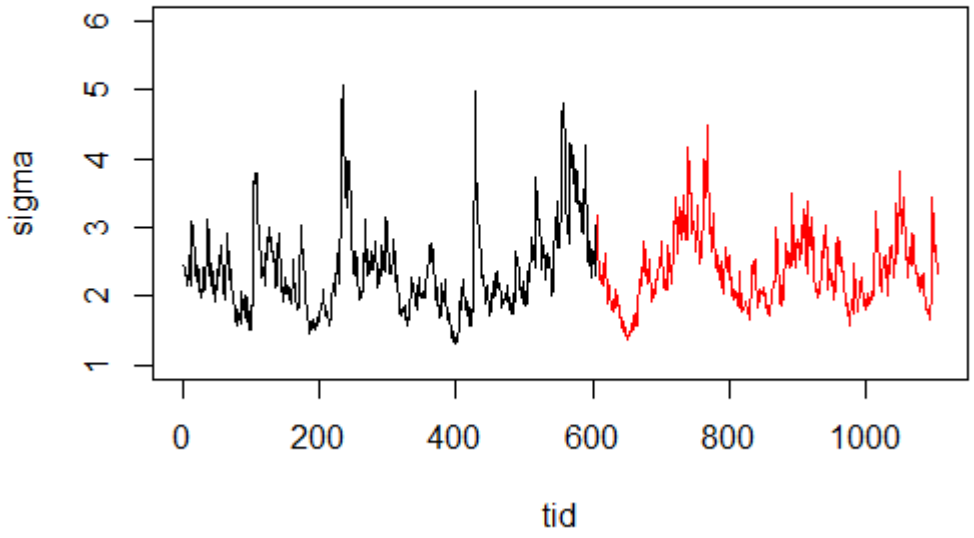
Figur 23. 200 simulationer fra tid 811 for en GARCH(1,1).

Vi vil nu, ligesom i forecasting afsnittet, også kigge på en EGARCH(1,1) med t-fordelt støj. Det viser sig dog hurtigt, at der umiddelbart ikke er den store forskel udover tilfældigheder. Det kan ses i Bilag 3, side 79. Ved forecastingen var det nemt, at se at deres ubetinget varians var forskellig, men når der bliver simuleret er det væsentlig sværere at se usikkerheden. Vi ved dog, at den ubetingede varians, og dermed vores usikkerhed, for EGARCH, uanset hvor simulationen starter fra, er lavere end for GARCH. Det lader dog ikke til, at det har nogen nævneværdig betydning når der bliver simuleret, og det kan derfor være svært at nå frem til en konklussion alene udfra simulationen. Vi så dog, at EGARCH tilsynladende var et bedre fit til datasættet i estimationskapitlet og forecastingen. Da forecastingen og simulationen bygger på samme model ville vi derfor også foretrække EGARCH over en standard GARCH når der simuleres.

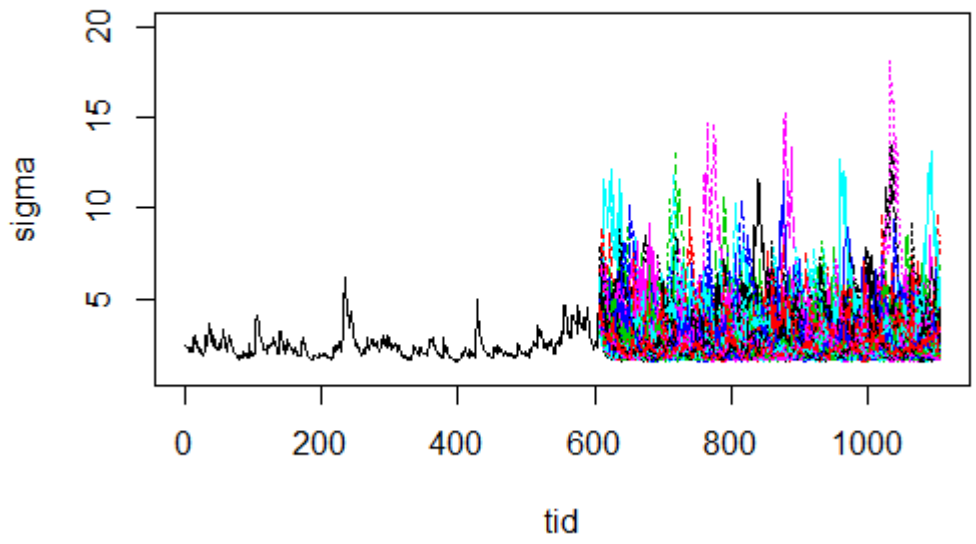
Vi kan naturligvis se, at der er en forskel hvis vi kigger på de estimeret varianser samt simulationen af disse. *Figur 24 og Figur 25* viser de simulerede varians-processer sammen med de estimerede processer af variansen baseret på data. De simulerede er de røde, imens de estimerede processer er de sorte. Det ses, at der umiddelbart lader til at være lavere varians for EGARCH modellen sammenlignet med GARCH. Dette gøres endnu tydligere hvis vi laver mange simulationer i samme plot. Dette ses på *Figur 26 og Figur 27*. Den lavere varians for EGARCH modellen, kan meget vel skyldes at den er bedre til at modellere asymmetrien i datasættet.



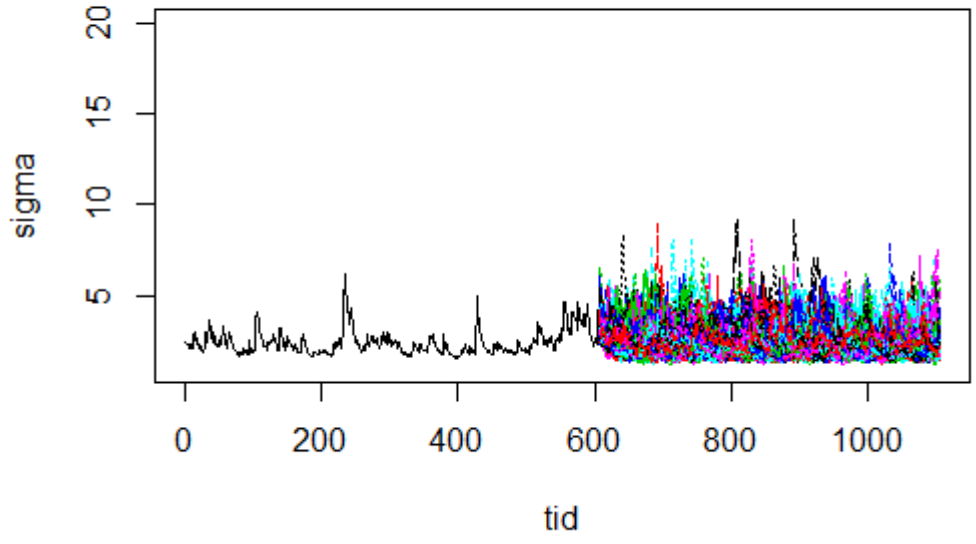
Figur 24. Sigma for GARCH(1,1)



Figur 25. Sigma for EGARCH(1,1)



Figur 26. 200 simulationer af GARCH.



Figur 27. 200 simulationer af EGARCH.

Kapitel 9

9.1 Indledning

I dette kapitel opsummerer vi, hvad vi har fundet ud af i de foregående kapitler. Derudover kommer vi frem til en endelig konklussion af specialet, og giver forslag til hvad man kunne arbejde videre med.

9.2 Konklussion

Man kan sige, at meningen med denne afhandling primært er at vise hvorfor GARCH modeller er at foretrække når man skal beskrive finansielle data. Vi så allerede i den indledende dataanalyse, at GARCH modeller lod til at passe datasættet bedst. Vi indså også hurtigt, at GARCH modellen formentlig ville kunne beskrive datasættet bedre end ARCH blandt andet udfra simulationen. Det var tydeligt, at vores datasæt opførte sig mest som en GARCH proces. Datasættet indeholdt mange klassiske kendetegn som finansielle data har. Der var klare tegn på volatilitetsclustering, og at datasættet havde tunge haler. Vi så derudover også tegn på, at støjvariablen skulle være andet end normalfordelt.

Vi fik hurtigt bekræftet, at støjvariablen ikke skulle være normalfordelt, og skulle derfor afgøre hvilken fordeling den skulle have. Her stod valget mellem t-fordelt og GED-fordelt. Igennem estimationen indså vi hurtigt at den bedste fordeling er t-fordelt for en GARCH(1,1). Det var også gældende for de øvrige modeller. Estimationen indikerede klare tegn på leverage effekt i datasættet eftersom EGARCH lod til at klare sig bedst i Sign Bias testet. EGARCH lod til at have nemmest ved at modellere virkningen af de 4 effekter: Fortegnet, negative stød, positive stød og til sidst den fælles virkning.

Vi indså dog også, ligesom flere andre, at GARCH(1,1) modellen var svær at forbedre. Det var begrænset hvor meget man fik ud af hvis man gjorde orden af GARCH modellen større. Man kunne beskrive en ting bedre, men så kunne GARCH(1,1) beskrive en, eller flere, ting bedre end den nye model. Vi indså at det ikke kunne betale sig at lave GARCH modeller med høj orden, og derfor endte vi med to modeller som var bedre end de andre, nemlig GARCH(1,1) og EGARCH(1,1). Som nævnt tidligere i konklussionen så er deres støj variable naturligvis t-fordelte, da tidligere resultater talte for det. Vi kunne konkludere, at EGARCH modellen var marginalt bedre end GARCH modellen, men at de begge havde deres fordele og ulemper.

På trods af at estimationen talte for, at EGARCH var den bedste model til vores datasæt, så blev det besluttet at tage begge modeller med videre til forecastingen for at sammenligne dem. Her så vi, at EGARCH behøvere et smallere konfidensinterval sammenlignet med en standard GARCH. Vi kunne derfor endnu en gang konstatere at EGARCH var en tand bedre end GARCH, i hvert fald hvis man så på en kortere forecasting tidhorisont. Den kunne ganske enkelt modellere volatilitetsclusteringen bedre end hvad GARCH kunne, og kunne derfor mere præcist placere konfidensgrænserne.

Simulationen udfra vores modeller så meget ens ud, og vi viste ved hjælp af disse, hvordan fremtidige værdier kunne se ud. Det var særdeles brugbart, da vi dermed fik en idé om hvad der skete inden for forecastingen intervaller, fremfor blot at vide at de fremtidige værdier kunne findes inden for deres intervaller. Her kunne vi ikke se nogen nævne værdi forskel, udover at EGARCH forudså en lavere varians end den anden model. Vi kunne dog benytte teorien og konstatere, at EGARCH nok engang var den bedste model.

9.3 Perspektivering

Som vi så i *R output 2* på side 33 så kom der to estimationsformer ud. Vi benyttede os af den velkendte MLE, men det kunne være interessant at benytte den anden estimation på vores datasæt. Standardafvigelserne for den nye estimationsform er generelt set højere, og mange fortalere for Robust standard errors mener, at den er god til at opdage outliers, eller ekstreme værdier. Når man i praktisk skal opfange ekstreme værdier, benytter man normalt klassiske fitting metoder, man kunne for eksempel prøve at fitte en ret linje udfra adskillige observationer. Den rette linje kan dog blive påvirke så meget af én enkelt ekstrem værdi, at den efterfølgende ikke kan opdage den ekstreme værdi. Dette fænomen kaldes også *masking effect*. Et klassisk fit kan også resultere i at en "god" observation fremstår som en ekstrem værdi, og det kaldes *swamping*. Et robust fit findes ved, at man skal finde et fit som er tæt på det fit man ville opnå uden de ekstreme værdier. Derefter kan vi identificere de ekstreme værdier udfra deres store afvigelse til det robuste fit. Det kunne være spændende, at se om en robust estimation ville medføre andre resultater.

Som nævnt allerede i afgrænsningen på side 2 så har vi valgt, at antallet af modeller er holdt nede i denne afhandling. Derfor kunne det være oplagt, at se om vi ville komme frem til samme konklussion hvis vi indragede flere modeller i vores analyse. Da der findes mange forskellige GARCH modeller, kunne det være interessant og vurdere de forskellige typer i forhold til hinanden. Især modeller *Nonlinear Asymmetric GARCH*, forkortet til NAGARCH kunne være relevant at benytte. Det skyldes, at vi har set klare indikationer på at vores data er asymmetrisk, og at det ikke er lineært. Der findes dog adskillige andre GARCH modeller, som alle kunne være interessante at teste på vores data.

Det have ligeledes været interessant, at se på hvor gode de to forecastingmetoder, henholdvis EGARCH og GARCH, hver især er til at forudsige den faktiske proces. Her kunne vi kigge på den estimerede sigma proces, hvor den naturligvis er estimeret på baggrund af den rigtige fremtidige proces. Vi kunne derefter se hvor tæt de forecastede sigma-værdier er på de estimerede værdier. Det er forventet, at der ville være et informationstab ved en forecasting, og man kunne så bruge velkendte metoder til at måle forecastingens præcision. Vi kunne, for eksempel, bruge *Mean Error*

eller *Mean Square Error*. I vores tilfælde lader det til at EGARCH er den bedste model, og det kunne værre interessant, at se om præcisionen også er mere præcis end ved en standard GARCH.

Til sidst kunne det også være interessant at kigge på et andet datasæt. Man kunne, for eksempel, udvide tidshorisonton således, at man have en god mængde data med før finanskrisen. På den måde kunne vi muligvis være i stand til, at sige noget om hvordan de forskellige modeller ville kunne forecaste før, under og efter en krise. Derudover kunne det ligeledes være spændende, at vælge et andet datasæt, selvom Danske Bank aktien er en del af C20-indekset så ville en anden aktie måske medføre at man nåede frem til en anden konklussion.

Litteraturliste

David Ruppert, David S. Matteson (2015). *Statistics and Data Analysis for Financial Engineering.*

John Knight, Stephen Satchell (2007). *Forecasting Volatility in the Financial Markets.*

Greg N. Gregoriou & Razvan Pascalau (2011). *Nonlinear Financial Econometrics: Forecasting Models, Computational and Bayesian Models.*

Rob Reider (2009). *Volatility Forecasting I: GARCH Models.*

Diethelm Würtz, Yohan Chalabi, Ladislav Luksan. *Parameter Estimation of ARMA Models with GARCH/APARCH Errors. An R and SPlus Software Implementation.*

Peter J. Rousseeuw, Mia Hubert (2011). *Robust statistics for outlier detection.*

Jesper H. Pedersen (2013). *ARMA(1,1)-GARCH(1,1). Estimation and forecast using rugarch 1.2-2.*

Data fundet på: https://danskebank.com/da/investor-relations/aktien

https://quantivity.wordpress.com/2011/02/21/why-log-returns/

https://people.orie.cornell.edu/davidr/SDAFE2/Rscripts/garch.R

http://www.portfolioprobe.com/2012/07/06/a-practical-introduction-to-garch-modeling/

Bilag

Bilag 1. ARCH og GARCH simulation.

```
## Indlæser pakkerne som er krævet.

library(fGarch)

library(timeSeries)

library(rugarch)

## GARCH simulation.

spec1 <- garchSim(garchSpec(model = list(omega = 0.2, alpha = .7, beta = 0.3)), n=600, n.start=600, extended=TRUE)

attach(spec1)

summary(spec1)

plot(garch,type='l',ylim=c(-8,8),ylab="GARCH",main="GARCH simulation")

plot(sigma,type='l',ylim=c(-0,8),ylab="Sigma",main="GARCH simulation")

## ARCH simulation.

spec2 <- garchSim(garchSpec(model=list(omega=0.2, alpha = .7, beta = 0.0)), n=600 ,n.start=600 ,extended=TRUE)

attach(spec2)

summary(spec2)

plot(garch,type='l',ylim=c(-8,8),ylab="ARCH",main="ARCH simulation")

plot(sigma,type='l',ylim=c(-0,8),ylab="Sigma",main="ARCH simulation")
```

Bilag 2. Relevant output fra relevante modeller

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.077702    0.022824   3.4044 0.000663
ar1     0.835279    0.070104  11.9148 0.000000
ma1    -0.874172    0.061273 -14.2668 0.000000
omega   0.103943    0.042844   2.4261 0.015264
alpha1  0.134323    0.031031   4.3287 0.000015
beta1   0.844216    0.036262  23.2812 0.000000
shape   5.394035    0.608343   8.8668 0.000000
LogLikelihood : -4006.235

Information Criteria
------------------------------------

Akaike       3.8239
Bayes        3.8428
Shibata      3.8239
Hannan-Quinn 3.8308

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      3.291 0.06965
Lag[2*(p+q)+(p+q)-1][5]     5.094 0.00239
Lag[4*(p+q)+(p+q)-1][9]     5.895 0.27396
d.o.f=2
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.02001  0.8875
Lag[2*(p+q)+(p+q)-1][5]   1.02780  0.8533
Lag[4*(p+q)+(p+q)-1][9]   2.78530  0.7938
d.o.f=2
Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           1.3979 0.1623
Negative Sign Bias  0.5094 0.6105
Positive Sign Bias  0.1202 0.9043
Joint Effect        6.1290 0.1055


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     20.34       0.3742
2    30     34.17       0.2330
3    40     38.65       0.4858
4    50     50.38       0.4186
```

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution     : ged

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.076452    0.022985   3.3261 0.000881
ar1     0.834069    0.048704  17.1253 0.000000
ma1    -0.870704    0.043270 -20.1224 0.000000
omega   0.089558    0.037762   2.3717 0.017708
alpha1  0.119142    0.027290   4.3658 0.000013
beta1   0.860190    0.032661  26.3369 0.000000
shape   1.268482    0.049630  25.5589 0.000000
LogLikelihood : -4019.662

Information Criteria
------------------------------------

Akaike       3.8367
Bayes        3.8556
Shibata      3.8367
Hannan-Quinn 3.8436

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic  p-value
Lag[1]                      3.073 0.079587
Lag[2*(p+q)+(p+q)-1][5]     4.729 0.008123
Lag[4*(p+q)+(p+q)-1][9]     5.523 0.342418
d.o.f=2
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1639  0.6856
Lag[2*(p+q)+(p+q)-1][5]    1.0037  0.8588
Lag[4*(p+q)+(p+q)-1][9]    2.6240  0.8190
d.o.f=2
Sign Bias Test
------------------------------------
                   t-value    prob sig
Sign Bias          1.32499 0.18532
Negative Sign Bias 0.81873 0.41303
Positive Sign Bias 0.02393 0.98091
Joint Effect       6.54468 0.08792   *


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     28.42      0.07563
2    30     45.46      0.02651
3    40     45.05      0.23358
4    50     52.57      0.33745
```

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,0)
Mean Model      : ARFIMA(0,0,0)
Distribution    : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.066787    0.030962   2.1571 0.030999
omega   2.295454    0.171826  13.3591 0.000000
alpha1  0.463642    0.069171   6.7029 0.000000
shape   4.201726    0.394891  10.6402 0.000000
LogLikelihood : -4078.228

Information Criteria
------------------------------------

Akaike       3.8897
Bayes        3.9005
Shibata      3.8897
Hannan-Quinn 3.8936

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.06817  0.7940
Lag[2*(p+q)+(p+q)-1][2]   0.11552  0.9089
Lag[4*(p+q)+(p+q)-1][5]   3.46686  0.3284
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic   p-value
Lag[1]                      4.345 3.712e-02
Lag[2*(p+q)+(p+q)-1][2]     9.457 2.660e-03
Lag[4*(p+q)+(p+q)-1][5]    18.742 4.293e-05
d.o.f=1
Sign Bias Test
------------------------------------
                   t-value    prob sig
Sign Bias          0.71580 0.47419
Negative Sign Bias 0.05944 0.95261
Positive Sign Bias 2.17086 0.03005  **
Joint Effect       5.00664 0.17131


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     21.92     0.288042
2    30     47.44     0.016802
3    40     70.43     0.001507
4    50     47.33     0.540973
```

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(2,0)
Mean Model       : ARFIMA(0,0,0)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.063651    0.030049   2.1182 0.034157
omega   1.733582    0.140228  12.3626 0.000000
alpha1  0.309829    0.056018   5.5308 0.000000
alpha2  0.295212    0.055445   5.3244 0.000000
shape   4.676411    0.474375   9.8580 0.000000
LogLikelihood : -4046.482

Information Criteria
------------------------------------

Akaike       3.8604
Bayes        3.8738
Shibata      3.8604
Hannan-Quinn 3.8653

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1263  0.7223
Lag[2*(p+q)+(p+q)-1][2]    0.1491  0.8865
Lag[4*(p+q)+(p+q)-1][5]    3.1686  0.3771
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      1.894  0.1688
Lag[2*(p+q)+(p+q)-1][5]     4.406  0.2076
Lag[4*(p+q)+(p+q)-1][9]     6.536  0.2412
d.o.f=2
Sign Bias Test
------------------------------------
                   t-value    prob sig
Sign Bias          0.36829 0.71270
Negative Sign Bias 0.06696 0.94662
Positive Sign Bias 1.86911 0.06175   *
Joint Effect       4.30142 0.23070


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     23.43     0.218944
2    30     33.40     0.261856
3    40     67.84     0.002851
4    50     42.14     0.745424
```

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(3,0)
Mean Model       : ARFIMA(0,0,0)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.067661    0.029935   2.2603 0.023802
omega   1.519498    0.134491  11.2981 0.000000
alpha1  0.283388    0.052758   5.3715 0.000000
alpha2  0.258286    0.052413   4.9279 0.000001
alpha3  0.118538    0.038334   3.0922 0.001987
shape   4.867135    0.507400   9.5923 0.000000

LogLikelihood : -4038.156

Information Criteria
------------------------------------

Akaike       3.8534
Bayes        3.8696
Shibata      3.8534
Hannan-Quinn 3.8593

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1546  0.6942
Lag[2*(p+q)+(p+q)-1][2]    0.1834  0.8645
Lag[4*(p+q)+(p+q)-1][5]    2.1573  0.5817
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                       1.683  0.1945
Lag[2*(p+q)+(p+q)-1][8]      5.213  0.3247
Lag[4*(p+q)+(p+q)-1][14]     8.705  0.3093
d.o.f=3
Sign Bias Test
------------------------------------
                   t-value    prob sig
Sign Bias          0.11977 0.90468
Negative Sign Bias 0.02099 0.98325
Positive Sign Bias 1.78559 0.07431   *
Joint Effect       4.45886 0.21599


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     21.18       0.3269
2    30     32.72       0.2894
3    40     45.16       0.2300
4    50     45.04       0.6342
```

```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(4,0)
Mean Model      : ARFIMA(0,0,0)
Distribution    : std

Optimal Parameters
-----------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.072263    0.029899   2.4169 0.015652
omega   1.418569    0.134002  10.5862 0.000000
alpha1  0.269144    0.051738   5.2021 0.000000
alpha2  0.236784    0.051679   4.5818 0.000005
alpha3  0.109012    0.037441   2.9115 0.003596
alpha4  0.065020    0.032500   2.0006 0.045434
shape   4.970796    0.524622   9.4750 0.000000
LogLikelihood : -4034.847

Information Criteria
-----------------------------------

Akaike       3.8512
Bayes        3.8701
Shibata      3.8512
Hannan-Quinn 3.8581

Weighted Ljung-Box Test on Standardized Residuals
-----------------------------------
                        statistic p-value
Lag[1]                     0.0986  0.7535
Lag[2*(p+q)+(p+q)-1][2]    0.1162  0.9084
Lag[4*(p+q)+(p+q)-1][5]    2.0742  0.6010
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
-----------------------------------
                         statistic p-value
Lag[1]                       1.645  0.1997
Lag[2*(p+q)+(p+q)-1][11]     6.297  0.3967
Lag[4*(p+q)+(p+q)-1][19]    11.224  0.3297
d.o.f=4
Sign Bias Test
-----------------------------------
                   t-value    prob sig
Sign Bias          0.05803 0.95373
Negative Sign Bias 0.01718 0.98629
Positive Sign Bias 1.75300 0.07975   *
Joint Effect       4.49567 0.21268


Adjusted Pearson Goodness-of-Fit Test:
-----------------------------------
  group statistic p-value(g-1)
1    20     19.14       0.4478
2    30     31.49       0.3429
3    40     41.62       0.3574
4    50     46.47       0.5761
```

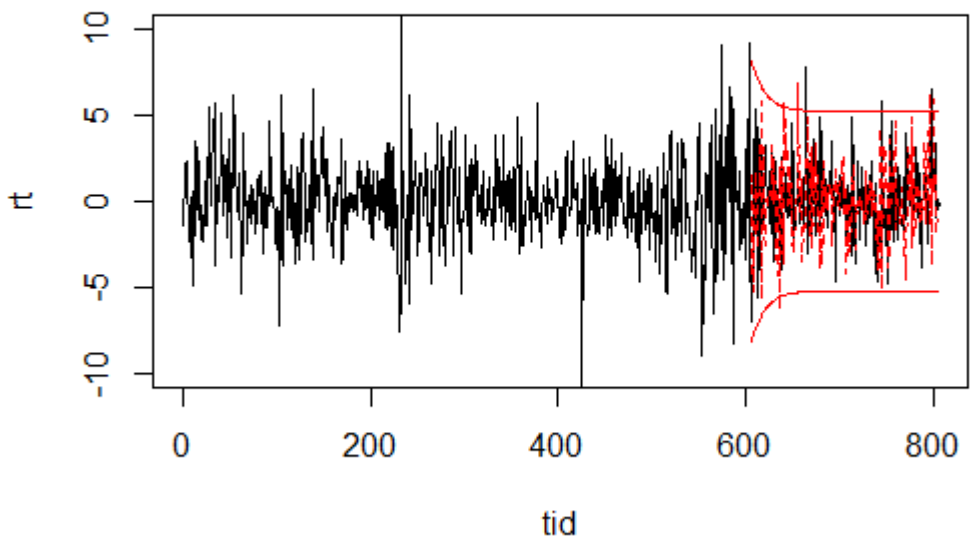
```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(1,3)
Mean Model      : ARFIMA(0,0,0)
Distribution    : ged

Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.041052    0.018093    2.2689 0.023274
omega   0.030473    0.007186    4.2408 0.000022
alpha1 -0.093543    0.022227   -4.2086 0.000026
beta1   0.922088    0.000172 5347.6181 0.000000
beta2  -0.533697    0.056358   -9.4698 0.000000
beta3   0.580337    0.060792    9.5463 0.000000
gamma1  0.279936    0.037736    7.4182 0.000000
shape   1.331170    0.053003   25.1151 0.000000
LogLikelihood : -3996.151

Information Criteria
------------------------------------

Akaike       3.8153
Bayes        3.8368
Shibata      3.8153
Hannan-Quinn 3.8232

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1244  0.7243
Lag[2*(p+q)+(p+q)-1][2]    0.1301  0.8990
Lag[4*(p+q)+(p+q)-1][5]    2.0042  0.6174
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                     0.01608  0.8991
Lag[2*(p+q)+(p+q)-1][11]   1.79748  0.9720
Lag[4*(p+q)+(p+q)-1][19]   3.92116  0.9803
d.o.f=4
Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           0.4261 0.6701
Negative Sign Bias  0.7123 0.4764
Positive Sign Bias  0.5119 0.6088
Joint Effect        0.8385 0.8402


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     31.64      0.03427
2    30     45.69      0.02517
3    40     59.19      0.02006
4    50     66.82      0.04606
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.045730    0.029048   1.5743 0.115416
omega   0.030038    0.010366   2.8977 0.003760
alpha1 -0.069670    0.016082  -4.3321 0.000015
beta1   0.970170    0.008848 109.6454 0.000000
gamma1  0.215926    0.027635   7.8136 0.000000
shape   5.831964    0.685152   8.5119 0.000000
LogLikelihood : -3993.382

Information Criteria
------------------------------------

Akaike       3.8108
Bayes        3.8269
Shibata      3.8107
Hannan-Quinn 3.8167

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2059  0.6500
Lag[2*(p+q)+(p+q)-1][2]    0.2158  0.8446
Lag[4*(p+q)+(p+q)-1][5]    1.8710  0.6492
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.3139  0.5753
Lag[2*(p+q)+(p+q)-1][5]    1.4711  0.7470
Lag[4*(p+q)+(p+q)-1][9]    3.3288  0.7037
d.o.f=2
Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           0.2892 0.7724
Negative Sign Bias  1.1935 0.2328
Positive Sign Bias  0.2107 0.8331
Joint Effect        1.8010 0.6147


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     18.72       0.4747
2    30     27.97       0.5195
3    40     36.25       0.5962
4    50     58.34       0.1697
```

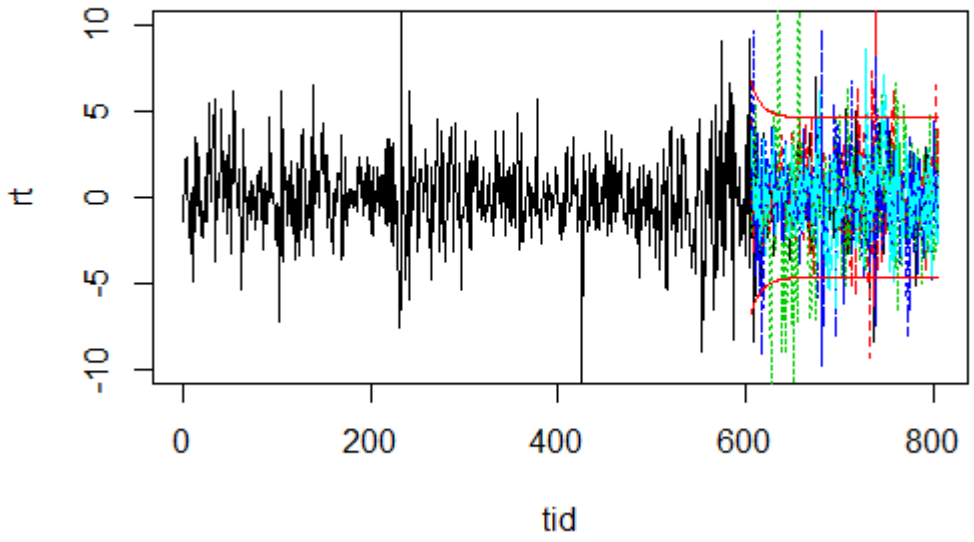
```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : eGARCH(1,2)
Mean Model       : ARFIMA(0,0,0)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.045178    0.029829   1.5145 0.129888
omega   0.031523    0.013476   2.3392 0.019324
alpha1 -0.087888    0.021862  -4.0201 0.000058
beta1   0.664063    0.012731  52.1626 0.000000
beta2   0.304281    0.008894  34.2130 0.000000
gamma1  0.252259    0.044497   5.6691 0.000000
shape   5.889890    0.706176   8.3405 0.000000
LogLikelihood : -3991.301

Information Criteria
------------------------------------

Akaike       3.8097
Bayes        3.8286
Shibata      3.8097
Hannan-Quinn 3.8166

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1398  0.7084
Lag[2*(p+q)+(p+q)-1][2]    0.1444  0.8896
Lag[4*(p+q)+(p+q)-1][5]    1.7800  0.6712
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                     0.01578  0.9000
Lag[2*(p+q)+(p+q)-1][8]    2.32637  0.8059
Lag[4*(p+q)+(p+q)-1][14]   4.07516  0.8730
d.o.f=3
Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           0.4124 0.6801
Negative Sign Bias  0.8498 0.3955
Positive Sign Bias  0.3517 0.7251
Joint Effect        0.9150 0.8218


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     20.17       0.3844
2    30     28.00       0.5180
3    40     35.22       0.6432
4    50     50.62       0.4094
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : eGARCH(1,3)
Mean Model       : ARFIMA(0,0,0)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.042784    0.030012     1.4255 0.154003
omega   0.029567    0.006114     4.8358 0.000001
alpha1 -0.090435    0.022400    -4.0373 0.000054
beta1   0.928710    0.000162 5716.3792 0.000000
beta2  -0.500498    0.018270   -27.3940 0.000000
beta3   0.541546    0.019628    27.5907 0.000000
gamma1  0.281622    0.036494     7.7169 0.000000
shape   5.983780    0.726064     8.2414 0.000000
LogLikelihood : -3984.667

Information Criteria
------------------------------------

Akaike       3.8044
Bayes        3.8259
Shibata      3.8043
Hannan-Quinn 3.8122

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1265  0.7221
Lag[2*(p+q)+(p+q)-1][2]    0.1328  0.8972
Lag[4*(p+q)+(p+q)-1][5]    1.9798  0.6232
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                     0.02517  0.8740
Lag[2*(p+q)+(p+q)-1][11]   1.92660  0.9649
Lag[4*(p+q)+(p+q)-1][19]   4.08428  0.9759
d.o.f=4
Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           0.4336 0.6646
Negative Sign Bias  0.6698 0.5031
Positive Sign Bias  0.5956 0.5515
Joint Effect        0.8836 0.8294


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     21.71       0.2987
2    30     30.69       0.3804
3    40     49.93       0.1128
4    50     55.10       0.2550
```

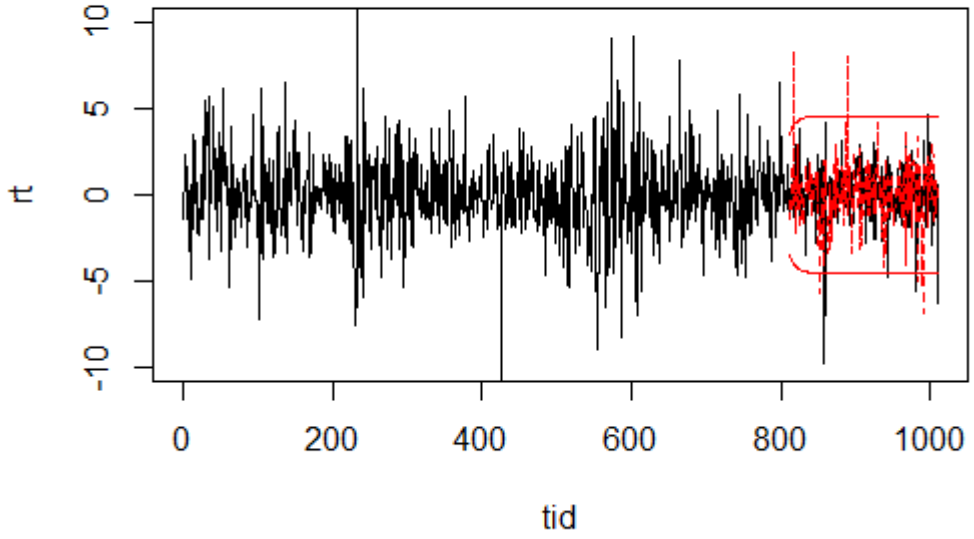
Bilag 3. Simulation for EGARCH



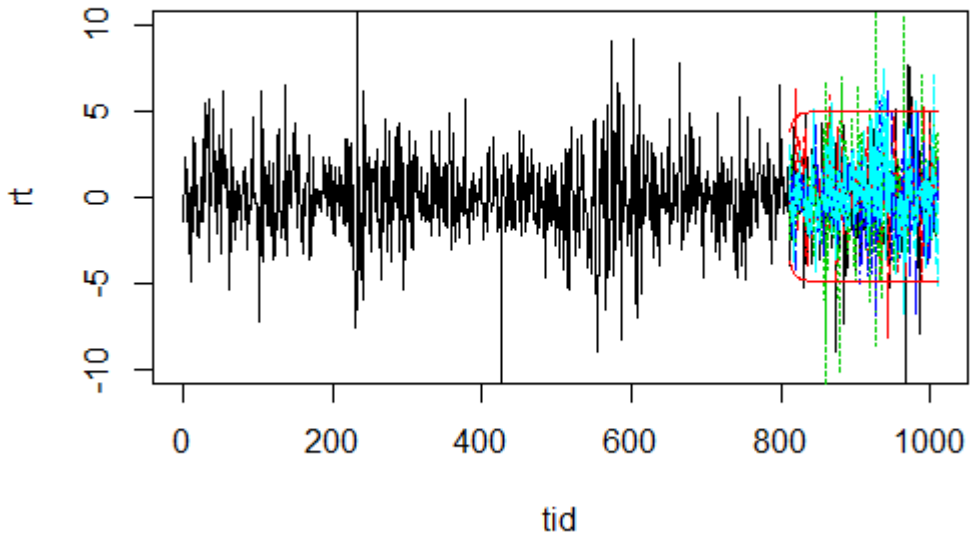
Bilag 3. 1. Simulation fra første forecasting tidspunkt af en EGARCH(1,1) processer med det faktiske afkast til sammenligning samt forecasting grænser.



Bilag 3. 2 Simulation fra første forecasting tidspunkt af flere EGARCH(1,1) processer med det faktiske afkast til sammenligning samt forecasting grænser.



Bilag 3. 3. Simulation fra andet forecasting tidspunkt af en EGARCH(1,1) processer med det faktiske afkast til sammenligning samt forecasting grænser.



Bilag 3. 4. Simulation fra andet forecasting tidspunkt af flere EGARCH(1,1) processer med det faktiske afkast til sammenligning samt forecasting grænser.

Bilag 4. Koden fra Rstudio.

```
library(readxl)
Data <- read_excel("Speciale/Data.xlsx")
Data<-Data[2642:1,]
attach(Data)
summary(Data)
plot(Lukkekurs,type='l',main="Lukkekurs", xlab="Observationer",ylab="Lukkekurs")
plot(Åben,type='l',main="Åben", xlab="Observationer",ylab="Åben")
plot(Høj,type='l',main="Høj", xlab="Observationer",ylab="Høj")
plot(Lav,type='l',main="Lav", xlab="Observationer",ylab="Lav")
rm(list=ls())

library(readxl)
Data <- read_excel("Speciale/Data.xlsx")
Data1<-Data[2100:1,]
attach(Data1)
summary(Data1)
plot(Lukkekurs,type='l',main="Lukkekurs", xlab="Observationer",ylab="Lukkekurs")
plot(Åben,type='l',main="Åben", xlab="Observationer",ylab="Åben")
plot(Høj,type='l',main="Høj", xlab="Observationer",ylab="Høj")
plot(Lav,type='l',main="Lav", xlab="Observationer",ylab="Lav")
Lukkekurs2=as.data.frame(Lukkekurs)
N2=length(Lukkekurs2[,1])
rt=100*(log(Lukkekurs2[2:N2,])-log(Lukkekurs2[1:(N2-1),]))
plot(rt,type='l')
qqnorm(rt)
qqline(rt)
hist(rt,prob=TRUE)
xx<-(-1500:1500)/100
lines(xx,dnorm(xx,mean=mean(rt),sd=sqrt(var(rt))))
boxplot(rt)
Box.test(rt,lag=5)
acf(rt)

plot(rt^2,type='l')
acf(rt^2)
Box.test(rt^2, lag = 20, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)
library(moments)
kurtosis(rt)
kurtosis(rt^2)
skewness(rt)
skewness(rt^2)


###### installer de krævede pakker.
library(fGarch)
library(devtools)
library(shiny)
library(rugarch)
```

```
M1<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(1, 1)),distribution.model = "norm"))
M1
## Nedenstående plot-funktion giver mulighed for at plotte det samme som findes i specialet.
plot(M1)

M2<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(1, 1)),distribution.model = "std"))
M2
plot(M2)

M3<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(1, 1)),distribution.model = "sged"))
M3
plot(M3)

## Yderlige modeller.
M21<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(1,1)),variance.model=list(model='sGARC
H', garchOrder=c(1, 1)),distribution.model = "std"))
M22<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(1,1)),variance.model=list(model='sGARC
H', garchOrder=c(1, 1)),distribution.model = "ged"))
M23<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(1, 0)),distribution.model = "std"))
M24<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(2, 0)),distribution.model = "std"))
M25<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(3, 0)),distribution.model = "std"))
M26<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARC
H', garchOrder=c(4, 0)),distribution.model = "std"))
M27<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='eGARC
H', garchOrder=c(1, 3)),distribution.model = "ged"))
M28<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='eGARC
H', garchOrder=c(1, 1)),distribution.model = "std"))
M29<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='eGARC
H', garchOrder=c(1, 2)),distribution.model = "std"))
M30<-
ugarchfit(data=rt,spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='eGARC
H', garchOrder=c(1, 3)),distribution.model = "std"))
```

```
## Forecasting.
O1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
O2<-
ugarchfit(data=rt[1:810],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
nahead=200
origin1=605
origin2=810
pred1 = ugarchforecast(O1, data = rt[1:605], n.ahead = nahead)
pred2 = ugarchforecast(O2, data = rt[1:810], n.ahead = nahead)
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-12,12),xlim=c(600,1015), main="Forecasting log
returns")
lines((origin2+1):(origin2+nahead),quantile(pred2,0.975),lwd=1,col=4)
lines((origin2+1):(origin2+nahead),quantile(pred2,0.025),lwd=1,col=4)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.025),col=2,lwd=1)

## Zoomer ind på de to starttidspunkter.
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-12,12),xlim=c(600,700), main="Forecasting log
returns")
lines((origin1+1):(origin1+nahead),quantile(pred1,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.025),col=2,lwd=1)
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-12,12),xlim=c(800,900), main="Forecasting log
returns")
lines((origin2+1):(origin2+nahead),quantile(pred2,0.975),lwd=1,col=4)
lines((origin2+1):(origin2+nahead),quantile(pred2,0.025),lwd=1,col=4)

## Forecasting med EGARCH model for samme interval.
O3<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
O4<-
ugarchfit(data=rt[1:810],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
nahead=200
origin1=605
origin2=810
pred3 = ugarchforecast(O3, data = rt[1:605], n.ahead = nahead)
pred4 = ugarchforecast(O4, data = rt[1:810], n.ahead = nahead)
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-12,12),xlim=c(600,1015), main="Forecasting log
returns")
lines((origin2+1):(origin2+nahead),quantile(pred4,0.975),lwd=1,col=5,lty=2)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.025),lwd=1,col=5,lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred3,0.975),col=6,lwd=1,lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred3,0.025),col=6,lwd=1,lty=2)
```

```
##Zoomer ind igen.
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-8,8),xlim=c(600,700), main="Forecasting log returns")
lines((origin1+1):(origin1+nahead),quantile(pred3,0.975),col=6,lwd=1,lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred3,0.025),col=6,lwd=1,lty=2)
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-12,12),xlim=c(800,900), main="Forecasting log returns")
lines((origin2+1):(origin2+nahead),quantile(pred4,0.975),col=5,lwd=1,lty=2)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.025),col=5,lwd=1,lty=2)

## Plot med de 4 forecastinger.
plot(rt,type="l", xlab="tid",ylab="log return", ylim=c(-12,12),xlim=c(600,1015), main="Forecasting log returns")
lines((origin2+1):(origin2+nahead),quantile(pred2,0.975),lwd=1,col=4)
lines((origin2+1):(origin2+nahead),quantile(pred2,0.025),lwd=1,col=4)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.025),col=2,lwd=1)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.975),lwd=1,col=5,lty=2)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.025),lwd=1,col=5,lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred3,0.975),col=6,lwd=1,lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred3,0.025),col=6,lwd=1,lty=2)

## Simulation
## GARCH tidspunkt 1
S1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model='sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
nsim<-200
S1sim<-ugarchsim(S1,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-S1sim@simulation$seriesSim
NN<-length(rt[1:605])
plot(rt[1:605],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=1)

## Kan lave rt længere for at se hvor godt simulationen passer, og samtidig tilføje
## et konfidensinterval fra vores forecasting af samme model.
plot(rt[1:806],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.025),col=2,lwd=1)

## Kan også lave flere simulationer på samme tid.
S1sim2<-ugarchsim(S1,n.sim=nsim,n.start=1,m.sim=5,startMethod='sample')
sim<-S1sim2@simulation$seriesSim
NN<-605
matplot((NN+1):(NN+nsim),sim,xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines(rt[1:605])
lines((origin1+1):(origin1+nahead),quantile(pred1,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.025),col=2,lwd=1)
```

```
##Tidspunkt 2
S2<-
ugarchfit(data=rt[1:810],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
nsim<-200
S2sim<-ugarchsim(S2,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-S2sim@simulation$seriesSim
NN<-length(rt[1:810])
plot(rt[1:810],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=1)

plot(rt[1:1011],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=2)
lines((origin2+1):(origin2+nahead),quantile(pred2,0.975),col=2,lwd=1)
lines((origin2+1):(origin2+nahead),quantile(pred2,0.025),col=2,lwd=1)

S2sim2<-ugarchsim(S2,n.sim=nsim,n.start=1,m.sim=5,startMethod='sample')
sim<-S2sim2@simulation$seriesSim
NN<-810
matplot((NN+1):(NN+nsim),sim,xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines(rt[1:810])
lines((origin2+1):(origin2+nahead),quantile(pred2,0.975),col=2,lwd=1)
lines((origin2+1):(origin2+nahead),quantile(pred2,0.025),col=2,lwd=1)

##EGARCH - Tidspunkt 1
E1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
nsim<-200
E1sim<-ugarchsim(E1,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-E1sim@simulation$seriesSim
NN<-length(rt[1:605])
plot(rt[1:605],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=1)

plot(rt[1:806],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=2)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred1,0.025),col=2,lwd=1)

E1sim2<-ugarchsim(E1,n.sim=nsim,n.start=1,m.sim=5,startMethod='sample')
sim<-E1sim2@simulation$seriesSim
NN<-605
matplot((NN+1):(NN+nsim),sim,xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines(rt[1:605])
lines((origin1+1):(origin1+nahead),quantile(pred3,0.975),col=2,lwd=1)
lines((origin1+1):(origin1+nahead),quantile(pred3,0.025),col=2,lwd=1)
```

```
## Tidspunkt 2
E2<-
ugarchfit(data=rt[1:810],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
nsim<-200
E2sim<-ugarchsim(E2,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-E2sim@simulation$seriesSim
NN<-length(rt[1:810])
plot(rt[1:810],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=1)

plot(rt[1:1011],xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines((NN+1):(NN+nsim),sim,col='red',lty=2)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.975),col=2,lwd=1)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.025),col=2,lwd=1)

E2sim2<-ugarchsim(E2,n.sim=nsim,n.start=1,m.sim=5,startMethod='sample')
sim<-E2sim2@simulation$seriesSim
NN<-810
matplot((NN+1):(NN+nsim),sim,xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='rt',ylim=c(-10,10))
lines(rt[1:810])
lines((origin2+1):(origin2+nahead),quantile(pred4,0.975),col=2,lwd=1)
lines((origin2+1):(origin2+nahead),quantile(pred4,0.025),col=2,lwd=1)

##Sigma sammenligning først for første tidspunkt.
NN<-605
nsim=500
S1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
S1sim<-ugarchsim(S1,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-S1sim@simulation$seriesSim
simsigmaS1<-S1sim@simulation$sigmaSim
plot(S1@fit$sigma,xlim=c(1,NN+nsim),type='l',xlab='tid',ylab='sigma',ylim=c(1,6))
lines((NN+1):(NN+nsim),simsigmaS1,col='red')

E1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
E1sim<-ugarchsim(E1,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-E1sim@simulation$seriesSim
simsigmaE1<-E1sim@simulation$sigmaSim
plot(E1@fit$sigma,xlim=c(1,NN+nsim),type='l',xlab='tid',ylab='sigma',ylim=c(1,6))
lines((NN+1):(NN+nsim),simsigmaE1,col='red')

## For det andet start tidspunkt.
NN<-810
S2<-
ugarchfit(data=rt[1:810],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
```

```
S2sim<-ugarchsim(S2,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-S2sim@simulation$seriesSim
simsigmaS2<-S2sim@simulation$sigmaSim
plot(S2@fit$sigma,xlim=c(1,NN+nsim),type='l',xlab='tid',ylab='sigma',ylim=c(1,6))
lines((NN+1):(NN+nsim),simsigmaS2,col='red')

E2<-
ugarchfit(data=rt[1:810],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
E2sim<-ugarchsim(E2,n.sim=nsim,n.start=1,m.sim=1,startMethod='sample')
sim<-E2sim@simulation$seriesSim
simsigmaE2<-E2sim@simulation$sigmaSim
plot(E2@fit$sigma,xlim=c(1,NN+nsim),type='l',xlab='tid',ylab='sigma',ylim=c(1,6))
lines((NN+1):(NN+nsim),simsigmaE2,col='red')

## Flere simulationer samtidig
NN<-605
nsim=500
S1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'sGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
S1sim2<-ugarchsim(S1,n.sim=nsim,n.start=1,m.sim=200,startMethod='sample')
sim<-S1sim2@simulation$seriesSim
simsigmaS12<-S1sim2@simulation$sigmaSim
matplot((NN+1):(NN+nsim),simsigmaS12,xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='sigma',ylim=c(1,20))
lines(1:NN,S1@fit$sigma)

E1<-
ugarchfit(data=rt[1:605],spec=ugarchspec(mean.model=list(armaOrder=c(0,0)),variance.model=list(model=
'eGARCH', garchOrder=c(1, 1)),distribution.model = "std"))
E1sim2<-ugarchsim(E1,n.sim=nsim,n.start=1,m.sim=200,startMethod='sample')
sim<-E1sim2@simulation$seriesSim
simsigmaE12<-E1sim2@simulation$sigmaSim
matplot((NN+1):(NN+nsim),simsigmaE12,xlim=c(1,NN+nsim),xlab='tid',type='l',ylab='sigma',ylim=c(1,20))
lines(1:NN,S1@fit$sigma)
```