

DCC-GARCH-modeller med ulike avhengighetsstrukturer

Masteroppgave i statistikk

14.mars 2013



Helene Aardal

Universitetet i Bergen

Matematisk institutt

Sammendrag

Hovedfokuset i denne oppgaven er å finne gode metoder for modellering av volatilitet og avhengighetsstruktur i finansielle porteføljer. En spesifikk multivariat GARCH modell, Dynamic Conditional Correlation (DCC-) GARCH, kombineres med copulaer og par-copula-konstruksjoner for å få en mer fleksibel modell til å modellere nettopp dette.

GARCH-modeller er verktøy for å predikere og analysere volatiliteten i tidsrekker når denne varierer over tid, mens copulaer gir mulighet til å modellere avhengighetsstruktur og marginaler hver for seg.

Flere DCC-GARCH-modeller med ulike avhengighetsstrukturer implementeres i oppgaven; En Copula-DCC-GARCH-modell med multivariat Student t copula, en PCC-DCC-GARCH-modell med Student t-copula for alle par av variable og en PCC-DCC-GARCH-modell der par-copula-konstruksjonen består av både Clayton og Student t copulaer.

Takk

Først og fremst ønsker jeg å takke min veileder, Kjersti Aaas, for å ha gitt meg et spennende tema å jobbe med, og for den hjelpen du har gitt meg i arbeidet med denne oppgaven. Selv om den geografiske avstanden til tider har føltes litt stor, har det vært veldig inspirerende og kjekt å ha deg som veileder.

Jeg ønsker også å takke Dag Tjøstheim og Trygve Nilsen. Dere har stått for store deler av den læringen jeg har tilegnet meg gjennom studiet ved UiB, og det er mye på grunn av dere at jeg ble inspirert til å gå videre med statistikk.

Jeg vil også rette en stor takk til alle mine medstudenter for å ha gjort studiehverdagen kjekkere. Spesielt vil jeg nevne Ingrid og Lars. Lars for å være en god faglig støtte, og for å alltid stille opp når jeg kommer med mine av og til ganske dumme spørsmål. Ingrid for gode faglige diskusjoner, men aller mest for alle de usaklige diskusjonene uten faglig innhold. Du har rett og slett gjort det mye morsommere å studere!

En siste takk til familie og venner for generell støtte gjennom hele studietiden.

Innhold

| | | |
|----------|---|-----------|
| 1 | Innledning | 1 |
| 2 | GARCH-modeller | 3 |
| 2.1 | Univariat GARCH | 3 |
| 2.1.1 | GARCH-modellen | 4 |
| 2.2 | Multivariat GARCH (MGARCH) | 5 |
| 2.2.1 | Definisjon | 5 |
| 2.2.2 | DCC-GARCH | 6 |
| 2.3 | Estimering av parametere i univariat GARCH | 8 |
| 3 | Copulaer | 11 |
| 3.1 | Definisjon | 12 |
| 3.2 | Implisitte Copulaer | 12 |
| 3.3 | Eksplisitte Copulaer | 15 |
| 3.3.1 | Clayton copula | 15 |
| 3.3.2 | Gumbel copula | 15 |
| 3.4 | Copulabaserte avhengighetsmål | 17 |
| 3.4.1 | Kendall's tau | 17 |
| 3.4.2 | Spearman's Rho | 19 |
| 3.4.3 | Haleavhengighetskoeffisienten | 20 |
| 3.5 | Simulere fra Copulaer | 22 |
| 3.5.1 | Implisitte copulaer | 22 |
| 3.5.2 | Eksplisitte copulaer | 22 |
| 3.6 | Tilpasse copulaer til data | 23 |
| 4 | Copula DCC-GARCH | 25 |
| 4.1 | Student t-fordelte marginaler og Student t-copula | 25 |
| 4.1.1 | Estimering av parametre | 26 |

| | | |
|----------|---|-----------|
| 4.1.2 | Estimeringsproblemer | 28 |
| 4.1.3 | Simulering av data | 29 |
| 5 | Par-copula-konstruksjoner | 31 |
| 5.1 | Definisjon | 32 |
| 5.2 | Vines | 33 |
| 5.3 | Simulering | 35 |
| 5.4 | Estimering | 36 |
| 6 | PCC-DCC-GARCH | 39 |
| 6.1 | Estimering av parametere | 40 |
| 6.1.1 | D-vine med bivariat Student's t copula for alle par | 41 |
| 6.1.2 | D-vine med både Student's t copula og Clayton copula | 42 |
| 6.2 | Simulering | 43 |
| 7 | Prediksjon og Validering | 45 |
| 7.1 | Prediksjon | 45 |
| 7.2 | Modellvalidering | 46 |
| 7.2.1 | AIC | 47 |
| 7.2.2 | Kupiec test og Christoffersen's Markov test | 47 |
| 8 | Robusthetsstudier | 49 |
| 8.1 | Variasjoner i DCC-parameterene | 49 |
| 8.1.1 | Modell 1: Copula-DCC-GARCH | 50 |
| 8.1.2 | Modell 2: PCC-DCC-GARCH med Student t for alle par | 51 |
| 8.1.3 | Modell 3: PCC-DCC-GARCH, Clayton og Student t | 53 |
| 8.2 | Videre testing av modell 2 | 55 |
| 8.2.1 | Korrelasjonsmatrise 1 | 55 |
| 8.2.2 | Korrelasjonsmatrise 2 | 61 |
| 8.3 | Oppsummering | 62 |
| 9 | Modellene brukt i praksis | 65 |
| 9.1 | Dataanalyse | 65 |
| 9.2 | Valg av par-copula-oppsett | 74 |
| 9.3 | Parameterestimering | 79 |
| 9.3.1 | 4-dimensjonal Student t-copula | 80 |
| 9.3.2 | Par-copula-konstruksjon med Student t copula for alle par av variable | 82 |
| 9.3.3 | Par-copula-konstruksjon med både Clayton- og Student t- copulaer | 84 |

| | | |
|-----------|--|------------|
| 9.4 | Modellvalidering | 86 |
| 9.4.1 | Backtesting Value at Risk; Kuipец test og Christoffersen's Markov test | 86 |
| 9.4.2 | AIC | 90 |
| 10 | Oppsummering og Konklusjon | 93 |
| 11 | Videre arbeid | 95 |
| | Referanser | 99 |
| A | Testdata | 101 |
| A.1 | Copula- DCC- GARCH | 101 |
| A.1.1 | Simulering | 101 |
| A.1.2 | Estimering av parametre | 103 |
| A.2 | PCC- DCC GARCH | 103 |
| A.2.1 | Simulering fra modellen med Student t copula for alle par | 103 |
| A.2.2 | Estimering av parametre | 106 |
| A.3 | Simulering fra modellen med Clayton og Student t par-copulaer | 107 |
| A.3.1 | Estimering av parametre | 109 |
| B | h-funksjoner | 111 |
| B.1 | Student t-copula | 111 |
| B.2 | Clayton copula | 114 |
| C | Algoritme 2 og 4 | 115 |
| C.1 | Algoritme 2 | 115 |
| C.2 | Algoritme 4 | 116 |
| D | R-kode | 119 |
| D.1 | Dataanalyse | 119 |
| D.1.1 | Diverse plott | 119 |
| D.1.2 | Valg av par-copula-konstruksjon | 121 |
| D.2 | Simulering fra de ulike copulaene | 124 |
| D.2.1 | Implisitte copulaer | 124 |
| D.2.2 | Eksplisitte copulaer | 125 |
| D.3 | Simulering fra 3-dimensjonal Copula DCC-GARCH | 125 |
| D.4 | Estimering av parameter i Copula DCC-GARCH | 127 |
| D.4.1 | Optimering | 128 |
| D.4.2 | Likelihoodfunksjon for de univariate rekkene: | 129 |

| | | |
|-------|--|-----|
| D.4.3 | Likelihood for Student t- copulaen i d dimensjoner med dynamisk kor- relasjon | 130 |
| D.5 | Simulering fra PCC-DCC-GARCH | 134 |
| D.5.1 | Student's t-copula for alle par | 134 |
| D.5.2 | Clayton- og Student's t PCC | 135 |
| D.6 | Estimering av parametere i PCC-DCC-GARCH | 135 |
| D.6.1 | Student's t-copula for alle par | 137 |
| D.6.2 | Clayton- og Student's t PCC | 143 |
| D.7 | VaR-evalueringer | 148 |

Kapittel 1

Innledning

I finans er volatilitet et viktig konsept som måler ustabilitet i avkastninger. Dette gjør det til en viktig faktor når man skal beregne hvilken risiko man står ovenfor. Det er et velkjent fenomen at volatiliteten varierer over tid og at den har en tendens til å komme i klynger; perioder med lav volatilitet og perioder med høy volatilitet. Det er også vist at volatiliteten i finansielle tidsrekker er autokorrelert; dagens volatilitet avhenger av tidligere volatilitet.

Siden volatiliteten ikke er direkte observerbar er det essensielt at man har en god modell for å estimere og predikere den. Mye litteratur er skrevet rundt modellering av univariat volatilitet, startet av Engle i [11], og den mest brukte modellen for modellering av volatiliteten i univariate tidsrekker er GARCH-modellen som Bollerslev introduserte i [7].

I de senere år har fokuset forflyttet seg over til simultane studier av store tverrsnitt av aktivaavkastninger. Det er nå kjent at finansielle volatiliteter beveger seg mer eller mindre sammen over tid på tvers av aktivaklasser og markeder. Det er ikke lenger nok å forstå hver risikokilde i isolasjon, men også hvordan alle risikokildene relaterer seg til hverandre og deres potensielle synergi til å danne katastrofale tap når ulike faktorer justeres på et bestemt vis [29]. Vi trenger å forstå den multivariate fordelingen til alle risikofaktorene.

Utvidelse av univariat GARCH til en multivariat GARCH-modell byr på flere utfordringer. Hovedutfordringen er å finne en fleksibel, men ikke for kompleks modell. En tilnærming, er å dekomponere den betingede kovariansmatrisen i betingede standardavvik og en betinget korrelasjonsmatrise. Den første modellen av denne typen var CCC (Constant Conditional Correlation), introdusert av Bollerslev i [31].

Men antagelsen om konstant korrelasjon over tid er ikke overbevisende siden korrelasjonen mellom aktiva varierer over tid. I 2001 introduserte Engle og Sheppard DCC-GARCH-modellen (Dynamical Conditional Correlation) i [13], en utvidelse av CCC-modellen der korrelasjonsmatrisen tillates å være tidsvarierende. Denne modellen har en to-steps algoritme for estimering av parameterene, noe som gjør modellen relativt enkel å bruke i praksis.

I en annen del av litteraturen brukes copulaer for å gi modelleringen av avhengighetsstrukturen mer fleksibilitet. Patton introduserer blant annet, i [24], bivariate modeller med tidsvarierende copulaer som tillater avvik fra normalfordelingen og symmetri i avhengigheten. Ved å bruke copulaer tillates man å modellere avhengighetsstruktur og marginaler hver for seg, og man får dermed mer fleksible modeller.

En gjenstående utfordring i multivariate GARCH-modeller er å finne fleksible fordelinger og modeller som også er gjennomførbare for store sett av aktiva. I [15] foreslår Heinen og Valdesogo en CAPM (Canonical Vine Autoregressive Model), hvor de modellerer avhengigheten i en DCC-modell med en C-vine-par-copula-konstruksjon.

Jeg vil i denne oppgaven kombinere teorien om DCC-GARCH-modellen med teorien om copulaer og par-copula-konstruksjoner. Jeg presenterer både en Copula-DCC-GARCH modell der jeg benytter en multivariat Student's t copula for å modellere avhengighetsstrukturen mellom dataene i tillegg til en PCC-DCC-GARCH-modell med mulighet for ulike copulaer for ulike par av aktiva. Selv om arbeidet til Heinen og Valdesogo i [15] er noe relatert, er det ingen som tidligere har studert den samme kombinasjonen av DCC-GARCH-modellen og par-copula-konstruksjoner som er implementert i denne oppgaven.

Opgaven er strukturert som følger: I kapittel 2 beskrives både univariat GARCH og DCC-GARCH-modellen. Generell teori om copulaer presenteres i kapittel 3, og i kapittel 4 kombinerer jeg teorien i de to foregående kapitlene for å beskrive Copula-DCC-GARCH-modellen. I kapittel 5 presenteres teori rundt par-copula-konstruksjoner og vine-strukturer. Denne teorien blir kombinert med DCC-teorien i kapittel 6, hvor PCC-DCC-GARCH-modellen blir introdusert, og i kapittel 7 presenteres ulike "Goodness of fit"-tester. I kapittel 8 gjøres et robusthetsstudie av modellene. Bruk av modellene på ekte data kommer i kapittel 9. Her gjennomføres også "Goodness of fit"-testene fra kapittel 7. Til slutt presenteres konklusjoner og forslag til videre arbeid i kapittel 10 og 11.

Kapittel 2

GARCH-modeller

I dette kapitlet vil jeg presentere ulike volatilitets- og korrelasjonsmodeller for finansielle tidsrekker, med spesielt fokus på den multivariate DCC-GARCH-modellen, introdusert av Engle og Sheppard i 2001 [13].

2.1 Univariat GARCH

For å kunne studere DCC-GARCH modellen, er det vesentlig å forstå den univariate GARCH-modellen. Ikke bare fordi modellen er en ikke-lineær kombinasjon av univariate GARCH-modeller, men også fordi den dynamiske betingede korrelasjonsmatrisen er basert på hvordan den univariate GARCH(1,1)-prosessen fungerer.[26]

Før ARCH-modellen ble introdusert av Engle i [11], som forøvrig ga han Nobelprisen i økonomi i 2003, var det vanlig å predikere volatiliteten i finansielle tidsrekker ved å bruke et fast antall tidligere observasjoner for å estimere standardavviket. Dette impliserer konstant volatilitet. Som nevnt i innledningen, er ikke variansen i slike prosesser konstant (homoskedastisk), men en prosess der volatiliteten forandrer seg med tiden (heteroskedastisk). ARCH-modellen ble videre generalisert til GARCH av danske Tim Bollerslev i hans doktorgradsavhandling [7] fra 1987. GARCH-modellen skiller seg fra ARCH ved at den i tillegg til heteroskedastisk volatilitet, fanger opp et fenomen kalt volatilitetsklynger. Med volatilitetsklynger mener man at finansielle data har en tendens til å vise perioder med høy volatilitet og perioder med lav volatilitet. Dette får GARCH-modellen frem ved at den lar volatiliteten ved tid t avhenge av foregående volatiliteter, i tillegg til de foregående kvadrerte forventningskorrigerede avkastningene. Den enkleste og mest populære GARCH-modellen er GARCH(1,1).

2.1.1 GARCH-modellen

Jeg bruker samme notasjon som Elisabeth Orskaug i [22], og den generelle GARCH(p,q)-modellen defineres ved ligningene [7]:

$$r_t = \mu_t + a_t \quad (2.1)$$

$$a_t = h_t^{1/2} z_t \quad (2.2)$$

$$h_t = \alpha_0 + \sum_{i=1}^q \alpha_i a_{t-1}^2 + \sum_{j=1}^p \beta_j h_{t-1} \quad (2.3)$$

Notasjon:

| | |
|--|--|
| r_t : | Log-avkastninger av et aktiva ved tid t |
| μ_t : | Forventet verdi for r_t ved tid t |
| a_t : | Forventningsjustert avkastning ved tid t . $a_t = r_t - \mu_t$ |
| h_t : | Varians ved tid t betinget på historien; kvadratet av volatiliteten. |
| z_t : | Følge av uavhengige og identisk fordelte variable med $E[z_t] = 0$ og $\text{Var}[z_t] = 1$. |
| $\alpha_0, \alpha_1, \dots, \alpha_q, \beta_1, \dots, \beta_p$: | Parametre i modellen. |

I likningen for volatiliteten, likning (2.4), har man $p + q + 1$ ukjente parametre. Ved å velge p og q høye kan man få en ganske kompleks modell. I denne oppgaven konsentrerer jeg meg om GARCH(1,1)-modellen. Her er $p = q = 1$. Man har altså 3 ukjente parametre:

$$h_t = \alpha_0 + \alpha_1 a_{t-1}^2 + \beta h_{t-1} \quad (2.4)$$

Som poengtert i [26], består GARCH(1,1)-modellen av tre ledd:

α_0 : Vektet varians i det lange løp.

$\alpha_1 a_{t-1}^2$: Kvadrerte realisjoner av a_{t-1} vektet med α_1 .

βh_{t-1} : Det autoregressive leddet. Volatiliteten ved forrige tidsenhet vektet med β .

Varians er som kjent en positiv størrelse, og for å sikre at h_t er positiv må vi ha $\alpha_0 \geq 0$, $\alpha_1 \geq 0$ og $\beta \geq 0$. For å sikre at modellen er stasjonær må man også ha $\alpha_1 + \beta < 1$. Stasjonærvariansen til modellen er da gitt ved $\frac{\alpha_0}{1-\alpha_1-\beta}$.

Legg merke til at GARCH-modellen ikke fanger opp assymetri i feilene, siden realisjonene av a_{t-1} er kvadrerte. Dette er en svakhet med modellen.

2.2 Multivariat GARCH (MGARCH)

Utvidelse av univariat GARCH til multivariat GARCH gir muligheten til å studere hvordan ulike markeder påvirker hverandre. I [17] nevnes 4 såkalte “stylized facts” ved multivariate avkastningsdata:

- For multivariate avkastningsserier er det lite som tyder på krysskorrelasjon.
- Ser man på absoluttverdien til avkastningene, er det derimot sterke bevis for krysskorrelasjon.
- Korrelasjonen varierer med tiden.
- Ekstreme avkastninger i en tidsrekke sammenfaller ofte med ekstreme avkastninger i andre tidsrekker.

Innsikt i hvordan ulike aksjemarkeder beveger seg i *forhold* til hverandre er viktig for pricing av derivater, porteføljeoptimering, risikohåndtering og hedging. Det finnes mange ulike MGARCH-modeller, og felles for flere av dem er at de kan være veldig komplekse og vanskelige å bruke i praksis. En vanlig måte å analysere hvordan korrelasjonen endrer seg, er derfor å tilpasse ulike modeller for endring i korrelasjon og gjøre statistiske sammenligninger av modellene[17].

2.2.1 Definisjon

Jeg følger igjen samme notasjon som i [22], og multivariat GARCH defineres som:

$$\mathbf{r}_t = \boldsymbol{\mu}_t + \mathbf{a}_t, \quad (2.5)$$

$$\mathbf{a}_t = \mathbf{H}_t^{1/2} \mathbf{z}_t. \quad (2.6)$$

Notasjon:

\mathbf{r}_t : $n \times 1$ vektor med logavkastninger av n aktiva ved tid t .

\mathbf{a}_t : $n \times 1$ vektor med forventningskorrigerede avkastninger av n aktiva ved tid t .

$\boldsymbol{\mu}_t$: $n \times 1$ vektor med med forventningsverdien til \mathbf{r}_t .

\mathbf{H}_t : Den betingede kovariansmatrisen til \mathbf{a}_t ved tid t .

\mathbf{z}_t : $n \times 1$ vektor med iid feilledd slik at $E[\mathbf{z}_t] = 0$ og $E[\mathbf{z}_t \mathbf{z}_t^T] = \mathbf{I}$.

MGARCH-modellene kan deles inn i flere undergrupper, som jeg ikke skal gå inn på her, men forskjellene i de ulike gruppene går på modellering av \mathbf{H}_t [3]. En av de mange multivariate GARCH-modellene er DCC-GARCH.

2.2.2 DCC-GARCH

DCC-GARCH hører til en undergruppe av multivariate GARCH-modeller som modellerer den betingede variansen og korrelasjonen hver for seg, i stedet for å modellere den betingede kovariansmatrisen direkte. Modellen ble introdusert i [13], og har en to-steps algoritme for å estimere parameterene, som gjør den relativt enkel å bruke i praksis.

“DCC-GARCH er en MGARCH-modell som har fleksibiliteten til univariat GARCH, men ikke kompleksiteten til multivariat GARCH[12].”

Spesielt for denne modellen er at den tillater korrelasjonsmatrisen å variere med tiden, samtidig som den betingede variansen modelleres som univariat GARCH. Man modellerer kovariansmatrisen som følger:

$$\mathbf{H}_t = \mathbf{D}_t \mathbf{R}_t \mathbf{D}_t, \quad t = 1, \dots, T. \quad (2.7)$$

Her er \mathbf{D}_t en diagonalmatrise med betingede standardavvik for \mathbf{a}_t ved tid t ,

$$\mathbf{D}_t = \begin{pmatrix} \sqrt{h_{1t}} & 0 & 0 & \dots & 0 \\ 0 & \sqrt{h_{2t}} & 0 & \dots & 0 \\ 0 & 0 & \sqrt{h_{3t}} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & 0 & 0 & \dots & \sqrt{h_{nt}} \end{pmatrix},$$

hvor h_{it} følger (2.4). \mathbf{R}_t er korrelasjonsmatrisen ved tid t ,

$$\mathbf{R}_t = \begin{pmatrix} 1 & \rho_{12,t} & \rho_{13,t} & \dots & \rho_{1n,t} \\ \rho_{21,t} & 1 & \rho_{23,t} & \dots & 0 \\ \rho_{31,t} & \rho_{32,t} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ \rho_{n1,t} & \rho_{n2,t} & \rho_{n3,t} & \dots & 1 \end{pmatrix}.$$

Sammenlignet med modeller der \mathbf{R}_t antas konstant over tid mister man fordelene med enkel numerisk estimering, og korrelasjonsmatrisen må estimeres for hver t i en rekursiv algoritme [30]. For å sikre at \mathbf{H}_t er positiv definit i modeller definert ved (2.7), må man i tillegg til veldefinerte varianser, h_{it} , ha \mathbf{R}_t positiv definit for alle t . For å sørge for dette modelleres \mathbf{R}_t ved:

$$\mathbf{R}_t = \mathbf{Q}_t^{*-1} \mathbf{Q}_t \mathbf{Q}_t^{*-1} \quad (2.8)$$

hvor

$$\mathbf{Q}_t = (1 - a - b) \bar{\mathbf{Q}} + a \boldsymbol{\epsilon}_{t-1} \boldsymbol{\epsilon}_{t-1}^T + b \mathbf{Q}_{t-1}. \quad (2.9)$$

I (2.9) er a og b positive parametre som tilfredstiller $a + b \geq 1$. $\boldsymbol{\epsilon}_t$ er de standardiserte residualene,

$$\boldsymbol{\epsilon}_t = \mathbf{D}_t^{-1} \mathbf{a}_t, \quad t = 1, \dots, T \quad (2.10)$$

og $\bar{\mathbf{Q}} = \text{cov}[\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t^T] = E[\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t^T]$ er den ubetingede kovariansmatrisen til disse. Denne kan estimeres ved

$$\bar{\mathbf{Q}} = \frac{1}{T} \sum_{t=1}^T \boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t^T. \quad (2.11)$$

\mathbf{Q}_t^* er en diagonalmatrise med kvadratroten av diagonalelementene i \mathbf{Q}_t langs diagonalen. Dette for å sikre at absoluttverdien til elementene i korrelasjonsmatrisen, \mathbf{R}_t , er mindre eller lik 1.

$$\mathbf{Q}_t^* = \begin{pmatrix} \sqrt{q_{11t}} & 0 & 0 & \dots & 0 \\ 0 & \sqrt{q_{22t}} & 0 & \dots & 0 \\ 0 & 0 & \sqrt{q_{33t}} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & 0 & 0 & \dots & \sqrt{q_{mmt}} \end{pmatrix}.$$

For å sikre at \mathbf{R}_t er positiv definit, må \mathbf{Q}_t være positiv definit. Begrensingene på a og b sikrer dette. Ligning (2.9) sikrer altså at vi får en positiv definit matrise, mens (2.8) skalerer slik at \mathbf{R}_t blir en korrelasjonsmatrise.

2.3 Estimering av parametere i univariat GARCH

Den mest brukte metoden for å estimere parametre i GARCH-modeller fra historiske data, er maximum likelihood. Det innebærer å velge verdier for parametrene som maksimerer sannsynligheten for akkurat de dataene vi har fått [16]. Likelihooden evalueres betinget på de observerte avkastningene, i tillegg til volatiliteten, som er uobserverbar, og bestemmes rekursivt. Maximum-likelihood-estimering av GARCH-modeller kan gjøres på to måter [17]:

- Quasi Maximum likelihood
- Full Maximum likelihood

Quasi Maximum likelihood brukes når det er selve GARCH-parameterene som er det sentrale i studien. Her antar man ingenting om hvilken fordeling residualene har, men bruker tettheten til standard normalfordelingen i likelihoodfunksjonen. Denne metoden har vist seg å gi gode

estimerer for α - og β -parameterene, men ulempen er at man ikke får noe informasjon om hvilken fordeling dataene følger.

I Full Maximum Likelihood brukes tettheten til den samme fordelingen i likelihoodfunksjonen som skal optimeres. Denne metoden gir mer informasjon, men krever at man kjenner fordelingen til dataene.

I begge metodene er man nødt til å velge en startverdi for σ_0 . Denne kan for eksempel være “sample-variansen” til de observerte avkastningene, eller 0.

Kapittel 3

Copulaer

Det definitivt mest brukte målet på avhengighet i den finansielle verden, og ellers, er den lineære korrelasjonskoeffisienten. Problemet med denne er at det kun er i tilfeller med normalfordelte variable er slik at korrelasjon lik 0 betyr uavhengighet. Ellers sier den oss bare noe om *lineær* uavhengighet. Korrelasjonen plukker altså ikke opp de ikke-lineære avhengighetene som eksisterer mellom mange risikofaktorer i den virkelige verden. Det er kun i tilfeller der risikofaktorene har en normalfordeling som simultanfordeling at det gir mening å bruke korrelasjonsmatrisen som en oppsummering av avhengighetsstrukturen mellom risikofaktorene. Mens simultanfordelingen da bestemmes fullstendig av marginalfordelinger og korrelasjon, er det utenfor denne verdenen uendelig mange fordelinger som kan passe.

Siden mye av den tidlige teorien i finans bygger på antagelser om multinormalfordelte avkastninger, har bruken av korrelasjon alene som mål på avhengigheten mellom de ulike variablene i disse tilfellene vært tilsynelatende uproblematisk. Det viser seg imidlertid at de fleste avhengige risikofaktorer i den virkelige verden ikke er normalfordelte og bruk av korrelasjon som eneste avhengighetsmål kan gi misvisende konklusjoner.

Til tross for at man i økonometrien har mange univariate fordelinger å velge mellom, er det for multivariate fordelinger få konkurrenter til multivariat normalfordeling og multivariat t-fordeling. Samtidig vet man at normalfordelingen ikke er forenlig med kunnskapen man har om kurtose og asymmetri i finansielle data. Student t-fordelingen har høyere kurtose, men ulempen med denne fordelingen er at man tvinger alle seriene til å følge samme marginalfordeling. Hvis man tilpasser univariate fordelinger til ulike typer aktiva i en portefølje, ser man for eksempel at haletyngden i de ulike marginalfordelingene varierer. Det ville derfor vært hensiktsmessig om man først kunne tilpasse hvert aktiva til den univariate fordelingen som passer best, og deretter kunne knytte disse sammen med en passende avhengighetsstruktur. Copulaer gir oss muligheten til dette.

Hovedkildene i dette kapitlet er [2],[17],[25] og [10].

3.1 Definisjon

En d -dimensjonal copula er en fordelingsfunksjon, C , på $[0, 1]^d$ med standard uniformt fordelte marginaler.

Sklar's teorem:

Dersom F er en simultanfordeling med marginaler F_1, F_2, \dots, F_d , eksisterer det en Copula $C : [0, 1]^d \rightarrow [0, 1]$ slik at, for alle x_1, \dots, x_d i $[-\infty, \infty]$, er

$$F(x_1, x_2, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)). \quad (3.1)$$

Direkte ut fra dette, følger det at den simultane tetthetsfunksjonen kan skrives som

$$f(x_1, x_2, \dots, x_d) = c(F_1(x_1), \dots, F_d(x_d))f_1(x_1)f_2(x_2) \dots f_d(x_d). \quad (3.2)$$

Hvis man kjenner copulaen og marginalfordelingene til variablene x_1, x_2, \dots, x_d , kan man altså enkelt definere simultanfordelingen til $\mathbf{X} = (x_1, x_2, \dots, x_d)$. Man kan definere marginalene og copulaen uavhengig av hverandre og likevel være sikret en ekte fordeling.

Det finnes to hovedgrupper av Copulaer; Implisitte Copulaer og Eksplisitte Copulaer.

3.2 Implisitte Copulaer

Implisitte copulaer har ingen enkel lukket form, men følger av velkjente multivariate fordelinger. Dersom disse kjente fordelingene er fordelinger med elliptiske konturer, kalles copulaen for elliptisk copula. De mest vanlige copulaene av denne typen er den Gaussiske copulaen og Student t-copulaen.

Tettheten til en implisitt copula korresponderende til en kontinuerlig fordelingsfunksjon, F , med kontinuerlige marginalfordelinger F_1, F_2, \dots, F_d , er gitt ved:

$$c(\mathbf{u}) = \frac{g(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))}{f_1(F_1^{-1}(u_1)) \dots f_d(F_d^{-1}(u_d))}, \quad (3.3)$$

hvor g er tettheten til den korresponderende simultanfordelingen, f_1, f_2, \dots, f_d er marginaltetthetene, og $F_1^{-1}, \dots, F_d^{-1}$ er de inverse kumulative fordelingsfunksjonene. Tettheten til den Gaussiske- og Student t- copulaen kan utledes ved bruk av denne formelen.

For student t copulaen med ν frihetsgrader er marginalfordelingene i nevneren i (3.3), f_1, f_2, \dots, f_d , iid $\sim t(\nu)$. Telleren er den standardiserte multivariate student t fordelingen, der $\boldsymbol{\mu}$ er nullvektoren, og skaleringsmatrisen, \mathbf{S} , er lik korrelasjonsmatrisen \mathbf{R} .

En stokastisk variabel, X , er univariat Student t-fordelt med ν frihetsgrader dersom tettheten er gitt ved

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\nu/2)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}. \quad (3.4)$$

En d -dimensjonal stokastisk vektor, $\mathbf{X} = (x_1, \dots, x_d)$, er multivariat Student t- fordelt med ν frihetsgrader, forventning $\boldsymbol{\mu}$ og skalamatrikse, \mathbf{R} , dersom tettheten er gitt ved

$$g(\mathbf{x}) = \frac{\Gamma[(\nu + d)/2]}{(\pi\nu)^{d/2}\Gamma(\nu/2)|\mathbf{R}|^{1/2}} \left(1 + \frac{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{R}^{-1}(\mathbf{x} - \boldsymbol{\mu})}{\nu}\right)^{-\frac{\nu+d}{2}}. \quad (3.5)$$

Hvis vi nå setter inn for f og g i (3.3) får vi altså:

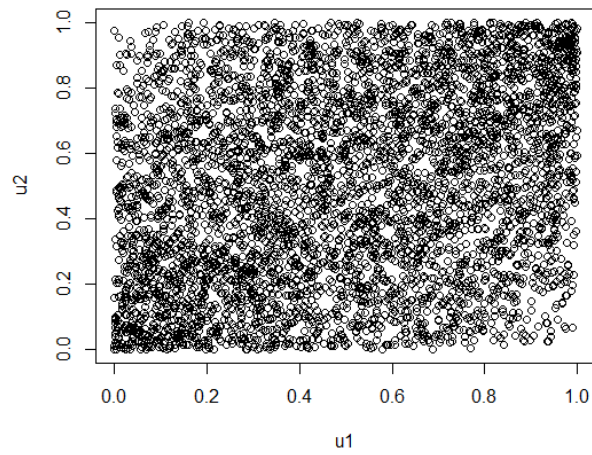
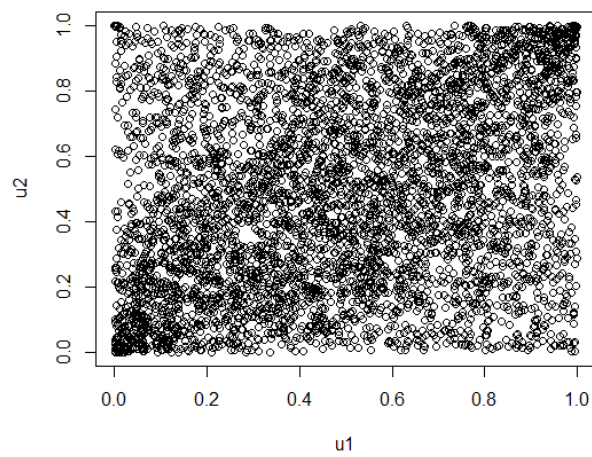
$$\begin{aligned} c(\mathbf{u}) &= \frac{g(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))}{f_1(F_1^{-1}(u_1)) \times \dots \times f_d(F_d^{-1}(u_d))} \\ &= \frac{g(x_1, \dots, x_d)}{f(x_1) \times \dots \times f(x_d)} \\ &= \frac{\frac{\Gamma(\frac{\nu+d}{2})}{(\pi\nu)^{d/2}\Gamma(\frac{\nu}{2})|\mathbf{R}|^{1/2}} \left(1 + \frac{\mathbf{x}^T \mathbf{R}^{-1} \mathbf{x}}{\nu}\right)^{-\frac{\nu+d}{2}}}{\prod_{i=1}^d \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})} \left(1 + \frac{x_i^2}{\nu}\right)^{-\frac{\nu+1}{2}}} \\ &= \frac{\Gamma(\frac{\nu+d}{2})\Gamma(\frac{\nu}{2})^{d-1} \left(1 + \frac{\mathbf{x}^T \mathbf{R}^{-1} \mathbf{x}}{\nu}\right)^{-\frac{\nu+d}{2}}}{\Gamma(\frac{\nu+1}{2})^d |\mathbf{R}|^{1/2} \prod_{j=1}^d \left(1 + \frac{x_j^2}{\nu}\right)^{-\frac{\nu+1}{2}}}. \end{aligned} \quad (3.6)$$

På samme måte, med standardisert multinormalfordeling i teller og iid standard normalfordelte marginalfordelinger i nevner i (3.3), får vi at tettheten til den gaussiske copulaen er:

$$c(\mathbf{u}) = \frac{\frac{1}{(2\pi)^{d/2}|\mathbf{R}|^{1/2}} e^{-1/2\mathbf{x}^T \mathbf{R}^{-1} \mathbf{x}}}{\prod_{j=1}^d \frac{1}{(2\pi)^{1/2}} e^{-1/2x_j^2}} = \frac{e^{-1/2\mathbf{x}^T \mathbf{R}^{-1} \mathbf{x}}}{|\mathbf{R}|^{1/2} e^{-1/2 \sum_{j=1}^d x_j^2}}. \quad (3.7)$$

Student t-copulaen gir positiv sannsynlighet for simultane ekstreme hendelser, mens den gaussiske copulaen gir sannsynlighet 0 for at ekstreme hendelser skal sammenfalle. I finansverden er det mye som tyder på at ekstreme hendelser skjer samtidig. En nyhet påvirker gjerne flere markeder samtidig, så den gaussiske copulaen vil sannsynligvis være for begrenset for modellering av avhengighetsstrukturen mellom slike data.

Figur 3.1 viser simuleringer fra de to copulaene. Man ser at for Student's t copulaen er det tettere med datapunkter i hjørnene, noe som viser sterkere avhengighet i halene.

(a) Gaussisk copula med $\rho = 0.3$ (b) t-copula med $\rho = 0.3$ og $\nu_C = 5$

Figur 3.1: 5000 simulerte datapunkt fra Gaussisk copula og Student's t-copula.

3.3 Eksplisitte Copulaer

Den andre typen copulaer er eksplisitte copulaer. Disse er ikke utledet fra kjente multivariate fordelinger, men fordelingsfunksjonen er gitt eksplisitt. En undergruppe av disse er arkimediske copulaer, og de mest vanlige er Clayton-copulaen og Gumbel-copulaen. Clayton-copulaen gir positiv sannsynlighet for simultane ekstreme negative hendelser, men ikke for simultane positive ekstreme hendelser. For Gumbel-copulaen er det omvendt. Dersom man tror på assymetri i avhengighetsstrukturen kan altså Student t-copulaen være for restriktiv.

3.3.1 Clayton copula

Clayton-copulaen er en assymmetrisk copula med sterkere avhengighet i den negative halen enn i den positive. Det er ikke helt opplagt hvordan man skal definere den d-dimensjonale Clayton-copulaen, men den 2-dimensjonale Clayton-copulaen er gitt ved

$$C_\delta(u_1, u_2) = (u_1^{-\delta} + u_2^{-\delta} - 1)^{-1/\delta}. \quad (3.8)$$

Jeg deriverer dette uttrykket med hensyn på u_1 og u_2 for å få tettheten:

$$\begin{aligned} c(u_1, u_2) &= \frac{\partial^2 C(u_1, u_2)}{\partial u_1 \partial u_2} \\ &= \frac{\partial}{\partial u_2} \left(\frac{\partial C(u_1, u_2)}{\partial u_1} \right) \\ &= \frac{\partial}{\partial u_2} \left(-\frac{1}{\delta} (u_1^{-\delta} + u_2^{-\delta} - 1)^{-1/\delta-1} (-\delta u^{-\delta-1}) \right) \\ &= \frac{\partial}{\partial u_2} \left((u_1^{-\delta} + u_2^{-\delta} - 1)^{1/\delta-1} u^{-(\delta+1)} \right) \\ &= -\left(\frac{1}{\delta} + 1\right) (u_1^{-\delta} + u_2^{-\delta} - 1)^{-1/\delta-2} (-\delta u_2^{-(\delta+1)}) u_1^{-(\delta+1)} \\ &= (1 + \delta) (u_1 u_2)^{-1-\delta} (u_1^{-\delta} + u_2^{-\delta} - 1)^{-1/\delta-2}. \end{aligned} \quad (3.9)$$

Her er $0 < \delta < \infty$ en parameter som måler avhengighet. Perfekt avhengighet oppnås for $\delta \rightarrow \infty$, mens $\delta \rightarrow 0$ gir uavhengighet.

3.3.2 Gumbel copula

Gumbel-copulaen er en assymmetrisk copula med sterkere avhengighet i den positive halen enn den negative. Gumbel-copulaen i to dimensjoner er gitt ved

$$C_\delta(u_1, u_2) = \exp\left(-\left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta}\right), \quad (3.10)$$

og tettheten blir

$$\begin{aligned} c(u_1, u_2) &= \frac{\partial C(u_1, u_2)}{\partial u_1 \partial u_2} = \frac{\partial}{\partial u_2} \left(\frac{\partial C(u_1, u_2)}{\partial u_1} \right) \\ &= \frac{\partial}{\partial u_2} \left(-C(u_1, u_2) \times \frac{1}{\delta} \left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta-1} \times \delta(-\log u_1)^{\delta-1} \times -\frac{1}{u_1} \right) \\ &= \frac{1}{u_1} (-\log u_1)^{\delta-1} \times \frac{\partial}{\partial u_2} \left(-C(u_1, u_2) \times \left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta-1} \right) \\ &= \frac{1}{u_1} (-\log u_1)^{\delta-1} \left(C(u_1, u_2) \times \frac{1}{\delta} \left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta-1} \right. \\ &\quad \times \delta(-\log u_2)^{\delta-1} \times -\frac{1}{u_2} \times \left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta-1} \\ &\quad \left. + -C(u_1, u_2) \times (1/\delta - 1) \times \left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta-2} \right. \\ &\quad \left. \times \delta(-\log u_2)^{\delta-1} \times \frac{1}{u_2} \right) \\ &= \frac{C(u_1, u_2)}{u_1 u_2} (\log u_1 \log u_2)^{\delta-1} \left((-\log u_1)^\delta + (-\log u_2)^\delta \right)^{-2+2/\delta} \\ &\quad + (\delta - 1) \times \left[(-\log u_1)^\delta + (-\log u_2)^\delta\right]^{1/\delta-2} \\ &= \frac{C(u_1, u_2)}{u_1 u_2} (\log u_1 \log u_2)^{\delta-1} \left((-\log u_1)^\delta + (-\log u_2)^\delta \right)^{-2+2/\delta} \\ &\quad \times \left(1 + (\delta - 1) \left((-\log u_1)^\delta + (-\log u_2)^\delta \right)^{-1/\delta} \right). \end{aligned} \quad (3.11)$$

$\delta \in (0, 1]$ er en parameter som måler avhengighet. Perfekt avhengighet oppnås for $\delta \rightarrow 0$, mens man har uavhengighet for $\delta = 1$.

Figur 3.2 viser simuleringer fra en todimensjonal Clayton og en todimensjonal Gumbel copula. Her ser man at datapunktene for Gumbel copulaen viser en sterkere avhengighet i øvre hale (stor ansamling av datapunkter i hjørnet øverst til høyre), mens for Clayton er det omvendt.

3.4 Copulabaserte avhengighetsmål

Ved siden av Pearsons korrelasjonskoeffisient, er rangkorrelasjon og haleavhengighetskoeffisienten avhengighetsmål basert på copulaer. Som nevnt tidligere inneholder den velkjente Pearsons korrelasjonskoeffisienten svært begrenset informasjon om de simultane egenskapene til stokastiske variable. Det er viktig å være klar over at korrelasjonen kun er et tilfredsstillende mål i tilfeller med normalfordeling.

3.4.1 Kendall's tau

Kendall's rangkorrelasjon kan ansees som et mål på konkordanse mellom bivariate tilfeldige vektorer. To vektorer (x_1, x_2) og $(\tilde{x}_1, \tilde{x}_2)$ kalles konkordant dersom $(x_1 - \tilde{x}_1)(x_2 - \tilde{x}_2) > 0$, og diskordant hvis $(x_1 - \tilde{x}_1)(x_2 - \tilde{x}_2) < 0$.

Kendall's tau er ganske enkelt sannsynligheten for konkordanse minus sannsynligheten for diskordanse mellom en tilfeldig vektor (X_1, X_2) og en uavhengig tilfeldig trekning fra samme fordeling:

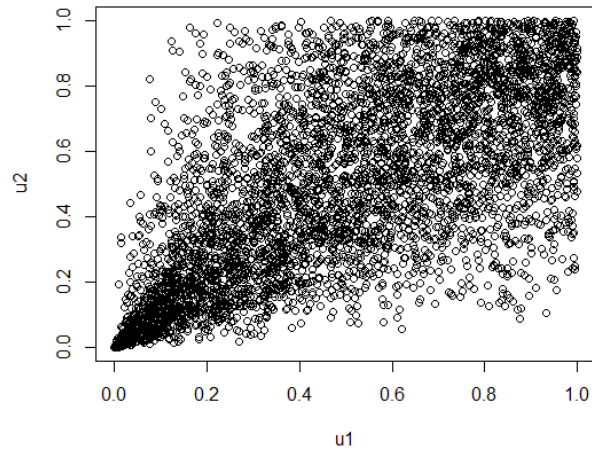
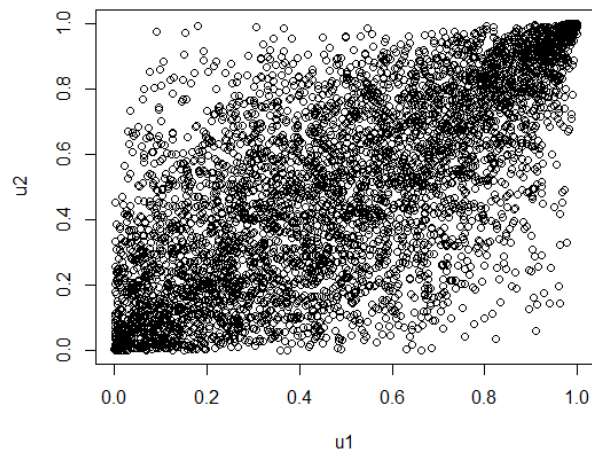
$$\begin{aligned}\rho_\tau(X_1, X_2) &= P((X_1 - \tilde{X}_1)(X_2 - \tilde{X}_2) > 0) - P((X_1 - \tilde{X}_1)(X_2 - \tilde{X}_2) < 0) \\ &= E(\text{sign}((X_1 - \tilde{X}_1)(X_2 - \tilde{X}_2))).\end{aligned}\quad (3.12)$$

Gitt at X_1 og X_2 har kontinuerlige marginalfordelinger og en unik copula C , kan Kendall's tau skrives som

$$\rho_\tau(X_1, X_2) = 4 \int_0^1 \int_0^1 C(u_1, u_2) dC(u_1, u_2) - 1. \quad (3.13)$$

Forholdet mellom den lineære korrelasjonskoeffisienten og Kendall's tau er for elliptiske copulaer gitt ved

$$\text{cor}(X_1, X_2) = \sin\left(\frac{\pi}{2}\rho_\tau\right). \quad (3.14)$$

(a) Clayton copula med $\delta = 2$ (b) Gumbel copula med $\delta = 2$

Figur 3.2: 5000 simulerte datapunkt fra Clayton copula og Gumbel copula.

For Arkimediske copulaer kan Kendall's tau relateres til avhengighetsparameteren, δ . Den er for Clayton-copulaen gitt ved

$$\rho_{\tau}(X_1, X_2) = \frac{\delta}{\delta + 2}, \quad (3.15)$$

og for Gumbel-copulaen er Kendall's tau gitt ved

$$\rho_{\tau}(X_1, X_2) = 1 - \frac{1}{\delta}. \quad (3.16)$$

3.4.2 Spearman's Rho

Et annet mål for rankkorrelasjon er Spearman's Rho. For tilfeldige variable X_1 og X_2 med fordelingsfunksjoner F_1 og F_2 , defineres denne som

$$\rho_S(X_1, X_2) = \text{cor}(F_1(X_1), F_2(X_2)). \quad (3.17)$$

Spearman's rho er med andre ord den lineære korrelasjonen mellom de sannsynlighetstransformerte variablene, og for kontinuerlige fordelinger er ρ_S gitt ved

$$\rho_S(X_1, X_2) = 12 \int_0^1 \int_0^1 (C(u_1, u_2) - u_1 u_2) du_1 du_2. \quad (3.18)$$

For den Gaussiske og Student t-copulaen, er forholdet mellom Spearman's rho og den lineære korrelasjonskoeffisienten gitt ved:

$$\text{cor}(X_1, X_2) = 2 \sin\left(\frac{\pi}{6}\rho_S\right). \quad (3.19)$$

Mens den lineære korrelasjonskoeffisienten måler graden av lineær avhengighet, er $\rho_\tau(X_1, X_2)$ og $\rho_S(X_1, X_2)$ mål på monoton avhengighet. I motsetning til den lineære korrelasjonskoeffisienten er disse avhengighetsmålene også invariante under monotone transformasjoner. Selv om dette indikerer at Spearman's rho og Kendall's tau er bedre å bruke, bør man i følge [17] heller velge en modell for avhengighetsstrukturen som reflekterer mer detaljert kunnskap om risikohåndteringsproblemet man står overfor enn å oppsummere avhengigheten med et enkelt tall. Som den lineære korrelasjonskoeffisienten har også rank-korrelasjon sine begrensninger. Det er for eksempel flere copulaer som har samme rangkorrelasjon.

3.4.3 Haleavhengighetskoeffisienten

Som de to foregående avhengighetsmålene er også haleavhengighet et mål på parvis avhengighet som bare avhenger av copulaen til et par av stokastiske variable, X_1 og X_2 . Haleavhengigheten fokuserer kun på deler av de simultane egenskapene, og måler avhengigheten for ekstremverdier; hvor stor grad av avhengighet er det i øvre og nedre kvadrant av en bivariat fordeling? Dette er viktig i risikohåndtering for å kunne beskytte seg mot samtidige uheldige begivenheter.

Den øvre haleavhengighetskoeffisienten defineres som

$$\lambda_u(X_1, X_2) = \lim_{\alpha \rightarrow 1} P(X_2 > F_2^{-1}(\alpha) | X_1 > F_1^{-1}(\alpha)), \quad (3.20)$$

og gir sannsynligheten for å observere en stor verdi for X_1 gitt at X_2 er stor. Analogt, er den nedre haleavhengighetskoeffisienten gitt ved

$$\lambda_l(X_1, X_2) = \lim_{\alpha \rightarrow 0} P(X_2 \geq F_2^{-1}(\alpha) | X_1 \geq F_1^{-1}(\alpha)). \quad (3.21)$$

De elliptiske copulaene har lik haleavhengighet i øvre og nedre hale. Som nevnt tidligere gir den gaussiske copulaen 0 i sannsynlighet for simultane ekstreme begivenheter,

$$\lambda_u(X_1, X_2) = \lambda_l(X_1, X_2) = 2 \lim_{x \rightarrow -\infty} \Phi \left(x \frac{\sqrt{1-\rho}}{\sqrt{1+\rho}} \right) = 0, \quad (3.22)$$

mens Student t-copulaen gir positiv sannsynlighet for det samme. Haleavhengighetskoeffisienten for Student t- copulaen er

$$\lambda_u(X_1, X_2) = \lambda_l(X_1, X_2) = 2t_\nu \left(-\sqrt{\nu+1} \sqrt{\frac{1-\rho}{1+\rho}} \right). \quad (3.23)$$

Clayton copulaen har som kjent avhengighet i nedre hale, og haleavhengighetskoeffisientene er

$$\lambda_u(X_1, X_2) = 0 \quad (3.24)$$

$$\lambda_l(X_1, X_2) = 2^{1/\delta}. \quad (3.25)$$

Tilsvarende viser koeffisientene for Gumbel-copulaen, som forventet, at denne har positiv avhengighet i øvre hale.

$$\lambda_l(X_1, X_2) = 0 \quad (3.26)$$

$$\lambda_u(X_1, X_2) = 2 - 2^{1/\delta}. \quad (3.27)$$

3.5 Simulere fra Copulaer

3.5.1 Implisitte copulaer

På grunn av måten de implisitte copulaene er konstruert på, er disse forholdsvis enkle å simulere fra gitt at vi kan trekke tilfeldige variable fra fordelingen de er utledet fra. Hvis vi kan generere en vektor \mathbf{X} med fordelingsfunksjon F , kan vi transformere hver komponent med sin egen marginalfordeling for å få en vektor $\mathbf{U} = (u_1, u_2, \dots, u_d)$ med fordelingsfunksjon C , copulaen til \mathbf{X} .

gaussisk copula: Tilfeldige trekninger fra en gaussisk copula skjer i to steg:

- (i) Generer $\mathbf{Z} \sim N_d(\mathbf{0}, \mathbf{R})$.
- (ii) Returner $\mathbf{U} = (\Phi(Z_1), \Phi(Z_2), \dots, \Phi(Z_d))$.

Student t copula: Som for gaussisk copula, trekker man tilfeldige variabler fra student t-copulaen gjennom to steg:

- (i) Generer $\mathbf{X} \sim t_d(\nu, \mathbf{0}, \mathbf{R})$.
- (ii) Returner $\mathbf{U} = (t_\nu(X_1), t_\nu(X_2), \dots, t_\nu(X_d))$.

3.5.2 Eksplisitte copulaer

Eksplisitte copulaer er en større utfordring å simulere fra. Både Gumbel- og Clayton-copulaen tilhører gruppen av Arkimediske copulaer. Fra [17, Theorem 5.42] har vi at hvis vi lar ϕ være en økende og konveks funksjon, $\phi : [0, 1] \rightarrow [0, \infty]$, kjent som generatoren til copulaen, så kan en bivariat Arkimedisk copula skrives som

$$C(u_1, u_2) = \phi^{[-1]}(\phi(u_1) + \phi(u_2)). \quad (3.28)$$

Dersom ϕ i tillegg er slik at den er den inverse til Laplace-transformasjonen til en fordelingsfunksjon G på \mathbb{R}_+ , hvor $G(0) = 0$, kan følgende simuleringsalgoritme benyttes for å generere trekninger fra en bivariat arkimedisk copula:

- (i) Generer en variabel x med fordelingsfunksjon G , slik at Laplace transformasjonen, \hat{G} , til G er den inverse til generatoren ϕ .
- (ii) Generer 2 uavhengige uniforme variable u_1, u_2 .

(iii) Returner $\mathbf{U} = (\hat{G}(-\ln(u_1/x)), \hat{G}(-\ln(u_2/x)))$.

Clayton copula: For Clayton-copulaen har vi at

$$\phi(t) = (t^{-\delta} - 1).$$

Dette medfører at

$$\phi(t)^{-1} = (t + 1)^{-1/\delta},$$

som er lik Laplace-transformasjonen, \hat{G} , til en gammavariabel $\text{Ga}(\alpha, \beta)$ med $\alpha = 1$ og $\beta = 1/\delta$. Simuleringsalgoritmen blir dermed:

- (i) Generer en gammavariabel $X \sim \text{Ga}(1/\delta, 1)$.
- (ii) Generer 2 uavhengige uniforme variable u_1, u_2 .
- (iii) Returner $\mathbf{U} = ((1 - \frac{\log(u_1)}{X})^{1/\delta}, (1 - \frac{\log(u_2)}{X})^{1/\delta})$.

3.6 Tilpasse copulaer til data

Når hovedmålet er estimering av en parametrisk copula, vil de ukjente marginalfordelingene til dataene opptre som “nuisance”/”støy”-parametre. Vanligvis starter man med å transformere dataene over til “copula-skalaen” ved å estimere de ukjente marginalene og deretter bruke den empiriske kumulative fordelingsfunksjonen for å få uniforme data. De ukjente marginalene estimeres enten ved å tilpasse parametriske fordelinger, eller ved å modellere marginalene ikke-parametrisk ved bruk av en empirisk fordelingsfunksjon, eller ved en blanding av disse to.

Når fordelingsfunksjonene er estimert og vi har fått pseudo-copula data, kan vi bruke Maximum Likelihood til å estimere parametrene i copulaen. La \hat{U}_i være de estimerte uniforme dataene. Likelihood- og log-likelihood-funksjonene er da gitt ved

$$L(\boldsymbol{\theta}; \hat{U}_1, \dots, \hat{U}_n) = \prod_{i=1}^n c_{\boldsymbol{\theta}}(\hat{U}_i), \quad (3.29)$$

og

$$l(\boldsymbol{\theta}; \hat{U}_1, \dots, \hat{U}_n) = \sum_{i=1}^n \log c_{\boldsymbol{\theta}}(\hat{U}_i). \quad (3.30)$$

Denne maksimeres med hensyn på θ , parametrene i copulaen, for å finne estimatene. I høyere dimensjoner kan dette være ganske komplekst, siden man må maksimere over alle mulige korrelasjonsmatriser. En enklere metode basert på Kendall's tau kan da benyttes for å estimere korrelasjonsmatrisen. Den betraktes da som kjent når man deretter maksimerer likelihood-funksjonen med hensyn på de resterende parametrene.

Kapittel 4

Copula DCC-GARCH

Jeg vil i dette kapittelet kombinere teorien fra kapittel 1 og 2 og studere en såkalt Copula-DCC-GARCH-modell.

Forskjellene mellom en vanlig DCC-GARCH modell med kjente multivariate fordelinger og en Copula-DCC-GARCH modell går kun på hvordan avhengighetsstrukturen til dataene beskrives. Korrelasjonsmatrisen og volatiliteten varierer med tiden på samme måte som før. Jeg vil først vise hvordan parametrene i en Copula-DCC-GARCH modell kan estimeres og deretter hvordan man kan generere data fra modellen.

4.1 Student t-fordelte marginaler og Student t-copula

Siden flere studier har vist at Student's t fordelingen er god til å modellere univariate finansdata, er det denne som brukes i de modellene jeg setter opp i denne oppgaven. Flere artikler, som for eksempel [21], har også vist at for denne typen data er Student t-copulaen overlegen i forhold til andre n-dimensjonale copulaer. Jeg velger derfor å sette opp en Copula-DCC-modell med Student's t fordelte feilledd i de univariate GARCH-rekkene og Student's t-copula for å modellere avhengighetsstrukturen.

I (2.2) er $z(t)$ definert til å ha forventning 0 og varians 1. Forventningen i Student's t fordelingen er 0, men siden variansen er lik $\frac{\nu}{\nu-2}$, må marginalfordelingene standardiseres ved å gjøre en transformasjon.

$$a_{it} = \sqrt{\frac{\nu_i - 2}{\nu_i}} \sqrt{h_{it}} y_{it}, \quad (4.1)$$

der $y_{it} \sim iid t(\nu_i)$, med $f(y_{it})$ som i (3.4).

$$y_{it}(a) = \frac{1}{\sqrt{h_{it}}} \sqrt{\frac{\nu_i}{\nu_i - 2}} a_{it} \quad (4.2)$$

$$y'_{it}(a) = \frac{1}{\sqrt{h_{it}}} \sqrt{\frac{\nu_i}{\nu_i - 2}} \quad (4.3)$$

For å finne fordelingen til a_{it} bruker jeg formelen for transformasjoner med en variabel :

$$\begin{aligned} g(a_{it}, \nu_i) &= f(z_{it}(a)) z'_{it}(a) \\ &= \frac{\Gamma(\frac{\nu_i+1}{2})}{\sqrt{\nu_i \pi} \Gamma(\nu_i/2)} \left(1 + \frac{z_{it}(a)^2}{\nu_i}\right)^{-\frac{\nu_i+1}{2}} z'_{it}(a) \\ &= \frac{\Gamma(\frac{\nu_i+1}{2})}{\sqrt{\nu_i \pi} \Gamma(\nu_i/2)} \left(1 + \frac{\frac{1}{h_{it}} \frac{\nu_i}{\nu_i-2} a_{it}^2}{\nu_i}\right)^{-\frac{\nu_i+1}{2}} \frac{1}{\sqrt{h_{it}}} \sqrt{\frac{\nu_i}{\nu_i-2}} \\ &= \frac{\Gamma((\nu_i+1)/2)}{\sqrt{(\nu_i-d)\pi} \sqrt{h_{it}} \Gamma(\nu_i/2)} \left(1 + \frac{a_{it}^2}{(\nu_i-2)h_{it}}\right)^{(\nu_i+1)/2}. \end{aligned} \quad (4.4)$$

Ved å bruke en copula til å binde sammen marginalene i stedet for å tvinge dem inn i en multivariat fordeling, kreves det ikke lenger at frihetsgradene i de ulike rekkene er de samme. Den multivariate fordelingen til avkastningsdataene modelleres i dette tilfellet med følgende simultanfordeling,

$$F(a_1, a_2, \dots, a_d) = C_\nu(F_1(a_1), \dots, F_d(a_d)),$$

hvor C_ν er den multivariate Student's t copulaen med ν frihetsgrader og dynamisk korrelasjonsmatrise etter ligning (2.8). F_1, \dots, F_d er de marginale fordelingene til a_t med tetthet som beskrevet over og frihetsgrader ν_1, \dots, ν_d . Volatiliteten i de univariate rekkene varierer etter ligning (2.4).

4.1.1 Estimering av parametre

Med copula som avhengighetsmål i DCC-GARCH modellen, kan man som nevnt tidligere tilpasse marginalfordelingene uavhengig av avhengighetsstrukturen. Man tilpasser dataene for hvert aktiva til en passende marginalfordeling og finner parametrene i \mathbf{D}_t i (2.7) som for univariat GARCH, beskrevet i avsnitt 2.3.

De standardiserte residualene, $\boldsymbol{\epsilon}_t = \mathbf{D}_t^{-1} \mathbf{a}_t$, har forventning 0, og avhengighetsstrukturen til $\epsilon_{1t}, \dots, \epsilon_{dt}$ er gitt ved en copula. Denne, sammen med parametrene i (2.9) gjenstår å estimere. De univariate parametrene antas kjent når disse skal estimeres.

De statistiske egenskapene til slike to-steps estimeringsalgoritmer for copulaer er diskutert i [24]. Her vises det at estimatene er konsistente og asymptotisk normalfordelte under standard betingelser.

Bidraget til likelihoodfunksjonen fra marginalfordelingene er:

$$\begin{aligned}
 L_{i,m}(\Phi_i) &= \prod_{t=1}^N g(a_{it}, \nu_i) \\
 &= \prod_{t=1}^N \frac{\Gamma((\nu_i + 1)/2)}{\sqrt{(\nu_i - d)\pi} \sqrt{h_{it}} \Gamma(\nu_i/2)} \left(1 + \frac{a_{it}^2}{(\nu_i - 2)h_{it}}\right)^{(\nu_i+1)/2} \\
 l_{i,m}(\Phi_i) = \log(L_{i,m}(\Phi_i)) &= \sum_{t=1}^N \left\{ \log \Gamma\left(\frac{\nu_i + 1}{2}\right) - \log \Gamma\left(\frac{\nu_i}{2}\right) - \frac{1}{2} \log(\nu_i - 2)\pi \right. \\
 &\quad \left. - \frac{\nu_i + 1}{2} \log\left(1 + \frac{a_{it}}{(\nu_i - 2)h_{it}}\right) - \frac{1}{2} \log h_{it} \right\}. \quad (4.5)
 \end{aligned}$$

Tettheten til t- copulaen er gitt ved (3.6), og bidraget til likelihooden fra Student t copulaen blir:

$$\begin{aligned}
 L_c(\Psi) &= \prod_{t=1}^N c(\mathbf{u}) \\
 &= \prod_{t=1}^N \frac{\Gamma(\frac{\nu+d}{2}) \Gamma(\frac{\nu}{2})^{d-1} \left(1 + \frac{\mathbf{x}_t^T \mathbf{R}_t^{-1} \mathbf{x}_t}{\nu}\right)^{-\frac{\nu+d}{2}}}{\Gamma(\frac{\nu+1}{2})^d |\mathbf{R}_t|^{\frac{1}{2}} \prod_{i=1}^d \left(1 + \frac{x_{it}^2}{\nu}\right)^{-\frac{\nu+1}{2}}} \\
 l_c(\Psi) = \log(L_c(\Psi)) &= \sum_{t=1}^N \left\{ -\frac{1}{2} \log |\mathbf{R}_t| + \log \Gamma\left(\frac{\nu+d}{2}\right) - \frac{\nu+d}{2} \log\left(1 + \frac{1}{\nu} \mathbf{x}_t^T \mathbf{R}_t^{-1} \mathbf{x}_t\right) \right. \\
 &\quad \left. + (d-1) \log \Gamma\left(\frac{\nu}{2}\right) - d \log \Gamma\left(\frac{\nu+1}{2}\right) + \frac{\nu+1}{2} \sum_{i=1}^d \log\left(1 + \frac{x_{it}}{\nu}\right) \right\}. \quad (4.6)
 \end{aligned}$$

I denne modellen er estimeringen av marginalfordelingene vel så viktig som estimering av

copulaen. I stedet for å bruke de empiriske kumulative fordelingsfunksjonene når dataene skal transformeres over til “copula-skalaen”, bruker jeg derfor heller de estimerte parametriske fordelingsfunksjonene. Denne metoden kalles IFM-metoden[19].

$$\mathbf{u} = (F_{\nu_1}(\epsilon_1), \dots, F_{\nu_d}(\epsilon_d)),$$

og

$$\mathbf{x} = (F_{\nu_c}^{-1}(u_1), \dots, F_{\nu_c}^{-1}(u_d)).$$

Her er $F_{\nu_i}(\epsilon_i)$ den kumulative fordelingen til Student t-fordelingen med ν_i frihetsgrader. Tilsvarende er $F_{\nu_c}^{-1}(u_i)$ den inverse kumulative fordelingen til Student t-fordelingen med ν_c frihetsgrader.

Setter vi sammen (4.5) og (4.6), blir log-likelihoodfunksjonen altså:

$$l(\Phi, \Psi) = \sum_{i=1}^d l_{i,m}(\Phi_i) + l_c(\Psi). \quad (4.7)$$

Her er parametrene α_{i0} , α_{i1} og β_i fra de univariate GARCH-modellene, samt parameteren ν_i , som svarer til frihetsgradene i de korresponderende t-fordelingene inneholdt i Φ_i , mens Ψ inneholder parametrene a og b og frihetsgradene, ν_c , i copulaen.

De ulike leddene i (4.7) kan optimeres hver for seg. Først finner man parametrene i marginalfordelingene, og deretter dem i copula-delen betinget på Φ . Når Φ_i er estimert for hvert aktiva, i , er også h_{it} estimert for $t = 0, 1, \dots, n$, og vi er i stand til å beregne $\epsilon_t = \mathbf{D}_t^{1/2} \mathbf{a}_t$ og $\bar{\mathbf{Q}} = \sum_{t=1}^n \epsilon_t \epsilon_t^T$, som settes inn i (4.6) før den maksimeres med hensyn på ν_c , a og b .

En svakhet med denne modellen er at korrelasjonen ikke tilsvarende korrelasjonen mellom avkastningene, men mellom de transformerte avkastningene. I (4.6) er $\mathbf{x}_t = F_{\nu_c}^{-1}(u_i)$, og \mathbf{R}_t blir korrelasjonen mellom disse. Det er mulig å finne de faktiske korrelasjonene mellom avkastningene ved å bruke Monte Carlo- simuleringer som i [13], men dette vil være omfattende og vanskelig, og estimeringsfeilen er så liten at den er uvesentlig, spesielt i to-dimensjonale tilfeller.

4.1.2 Estimeringsproblemer

I multivariate modeller er ofte valg av startverdier viktig. Når antall parametere er høyt, blir likelihoodfunksjonen flat, og man står i fare for å finne et lokalt maksimum. For å være sikker på at parameterene man finner virkelig er et globalt maksimum er det lurt å kjøre estimeringen med mange forskjellige startverdier for parameterene.

4.1.3 Simulering av data

Når man har alle parametrene man trenger og vet hvilken copula og hvilke marginalfordelinger man vil bruke, er det relativt enkelt å simulere fra en Copula-DCC-GARCH-modell.

Dersom volatiliteten er estimert for de univariate rekkene ved tidligere tidspunkt og man er interessert i å predikere utviklingen videre, er det vanlig å bruke den siste estimerte verdien som startverdi for volatiliteten. Hvis ikke, er det vanlig å bruke stasjonærvariansen; $h_{i0} = \frac{\alpha_i 0}{1 - \alpha_i - \beta_i}$. Deretter, for hvert tidspunkt t :

- Beregn volatiliteten i de univariate rekkene med (2.4), og lag matrisen \mathbf{D}_t .
- Beregn korrelasjonsmatrisen, \mathbf{R}_t , med (2.9) og (2.8).
- Generer en tilfeldig trekning fra den d-dimensjonale copulaen med korrelasjonsmatrise \mathbf{R}_t .
- Beregn ϵ_t ved å benytte de inverse kumulative fordelingsfunksjonene på de uniforme variablene fra copulaen.
- Beregn de forventningskorrigerede log-avkastningene: $\mathbf{a}_t = \mathbf{D}_t \epsilon_t$.

Kapittel 5

Par-copula-konstruksjoner

Selv om man i teorien kan konstruere flere copulaer med mer en to variable, har de fleste av disse kun en parameter, slik at man får flerdimensjonale problemer. I praksis har man derfor bare den gaussiske og Student t-copulaen å velge mellom. Siden den gaussiske copulaen i bunn og grunn bare er Student t-copulaen med et høyt antall frihetsgrader, står vi altså bare igjen med å bestemme frihetsgradene i Student t-copulaen.

For finansielle data har Student t-copulaen vist seg å være klart bedre enn andre n-dimensjonale copulaer, men en svakhet med denne er at den kun har én parameter for haleavhengighet, uavhengig av dimensjon. I store datasett er det naturlig at ikke alle par av risikofaktorer har lik avhengighetsstruktur, og derfor heller ikke lik haleavhengighet.

Jeg vil i dette kapittelet introdusere såkalte par-copula-konstruksjoner. Denne måten å dekomponere en multivariat fordeling på, gir mulighet til å tilpasse ulike par av risikofaktorer til ulike bivariate copulaer, samtidig som man tilpasser de univariate fordelingene uavhengig av avhengighetsstrukturen som før.

“Den grunnleggende ideen bak par-copula konstruksjoner er å dekomponere en tilfeldig fordelingsfunksjon ned til enkle bivariate byggeklosser og sy dem formålstjenlig sammen. Disse bivariate klossene er bivariate copulaer, og vi har mange å velge mellom.” Oversatt fra [29]

Slike konstruksjoner er en enkel og fleksibel måte å spesifisere multivariate avhengighetsmodeller på. Senere i oppgaven kombinerer jeg denne teorien med DCC-GARCH-modellen. Hovedkildene her er [1], [8] og [29].

5.1 Definisjon

Ideen bak par-copula- konstruksjoner er relativt enkel, og baserer seg på den velkjente egenskapen at enhver tetthetsfunksjon $f(x_1, x_2, \dots, x_n)$ kan faktoriseres på følgende måte:

$$f(x_1, x_2, \dots, x_n) = f(x_1) \cdot f(x_2|x_1) \cdot f(x_3|x_2, x_1) \cdots f(x_n|x_1, \dots, x_{n-1}). \quad (5.1)$$

I tillegg har vi Sklar's teorem som sier at enhver multivariat fordelingsfunksjon F , med marginaler F_1, F_2, \dots, F_n kan skrives som

$$F(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)),$$

og ved partiellderivering av denne, får vi følgende uttrykk for tettheten:

$$f(x_1, x_2, \dots, x_n) = c_{12\dots n}(F_1(x_1), \dots, F_n(x_n)) \cdot f_1(x_1) \cdots f_n(x_n), \quad (5.2)$$

hvor $c_{12\dots n}$ er den n-dimensjonale copula-tettheten.

Hver av faktorene i (5.1) kan dekomponeres videre ved å bruke betingede copulaer. Hvis man forenkler til det bivariate tilfellet og setter inn høyresiden i (5.1) for venstre siden i (5.2), ser man for eksempel lett at den betingede tettheten kan skrives som

$$f(x_1|x_2) = c_{12}(F_1(x_1), F_2(x_2)) \cdot f_1(x_1).$$

På samme måte, er en mulig (blant flere) dekomponering av den andre betingede tettheten i (5.1)

$$\begin{aligned} f(x_3|x_2, x_1) &= c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \cdot f(x_3|x_2) \\ &= c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \cdot c_{23}(F_1(x_1), F_3(x_3)) \cdot f(x_3) \end{aligned}$$

Vi kan mer generelt dekomponere hvert ledd i (5.1) til marginalfordelinger og tilfredsstillende bivariate copulaer ved å bruke formelen

$$f(x|\mathbf{v}) = c_{x,v_j|\mathbf{v}_{-j}}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))f(x|\mathbf{v}_{-j}), \quad (5.3)$$

hvor \mathbf{v} er en d-dimensjonal vektor, v_j er en tilfeldig valgt komponent av \mathbf{v} og \mathbf{v}_{-j} er den (d-1)-dimensjonale vektoren \mathbf{v} uten komponenten v_j . Konstruksjonen er iterativ av natur, og gitt en spesiell faktorisering, finnes det fortsatt mange forskjellige reparameteriseringer.

Merk at man i disse konstruksjonene må evaluere mange av par-copulaene ved en betinget fordeling på formen $F(x|\mathbf{v})$. Utregningen av disse betingede fordelingene er også rekursive, og i følge [18] har vi at

$$F(x|\mathbf{v}) = \frac{\partial C_{xv_j|\mathbf{v}_{-j}}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))}{\partial F(v_j|\mathbf{v}_{-j})}. \quad (5.4)$$

For spesialtilfellet der \mathbf{v} kun har en komponent, får vi

$$F(x|v) = \frac{\partial C_{xv}(F_x(x), F_v(v))}{\partial F_v(v)}, \quad (5.5)$$

og dersom vi videre antar at marginalfordelingene $F_x(x)$ og $F_v(v)$ er uniforme, reduseres (5.4) til

$$F(x|v) = \frac{\partial C_{xv}(x, v)}{\partial v}. \quad (5.6)$$

Denne sammenhengen brukes mye i simulerings- og optimeringsalgoritmer, så vi definerer funksjonen $h(\cdot)$ som

$$h(x, v; \Theta) = F(x|v) = \frac{\partial C_{xv}(x, v; \Theta)}{\partial v}, \quad (5.7)$$

hvor det tredje argumentet Θ er parameterene i copulaen som representerer simultanfordelingen til x og v . Den inverse til h med hensyn på den første variabelen, betegnes h^{-1} ; altså den inverse til den betingede fordelingsfunksjonen.

Alle de bivariate copulaene i en par-copula-konstruksjon må ikke komme fra samme “familie”. Man kan fritt velge den copulaen som passer dataene best, blant alle mulige bivariate copulaer.

5.2 Vines

For høy dimensjon, d , er antall mulige par-copula-konstruksjoner veldig stor (240 konstruksjoner ved $d=5$), så man trenger en måte å organisere dem på. “The regular vine” ble introdusert i [4], og er en grafisk måte å gjøre nettopp dette.

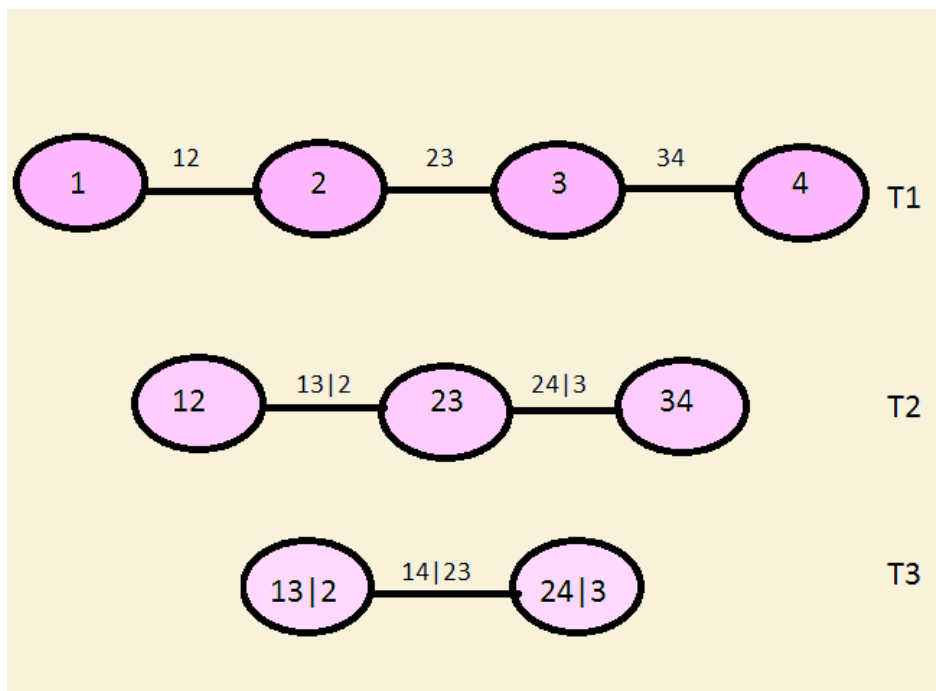
En regulær vine er, kort sagt, et sett av nivåer hvor kantene i nivå nummer i er noder i nivå $i + 1$, og hvor to kanter i nivå i er knyttet sammen med en kant i nivå $i + 1$ hvis, og bare

hvis, de har en felles node. Antall noder som en bestemt node er knyttet til, kalles *graden* til noden.

Jeg konsentrerer meg her om to typer regulære vines; kanonisk vine (C-vine) og D-vine. En regulær vine kalles:

- (a) C-vine dersom hvert nivå, T_i , $i = 1, \dots, d - 1$, har en unik node av grad $d - 1$. (Stjerne)
- (b) D-vine dersom hver node i T_1 har grad 2 eller mindre. (Sti)

“Intuisjonen bak kanoniske vines er at en variabel spiller en stor rolle i avhengighetsstrukturen, så man linker alle til denne. For en D-vine er ting mer symmetrisk”[29].



Figur 5.1: D-vine i 4 dimensjoner.

Figur 5.1 og 5.2 er eksempler på henholdsvis en D-vine og en C-vine i fire dimensjoner. Den 4-dimensjonale D-vinen består av fire nivå; T_j , $j = 1, 2, 3, 4$. Nivå T_j har $5 - j$ noder og $4 - j$ kanter. Hver kant i D-vine-nivået korresponderer til en par-copula-tetthet. Kant $13|2$ korresponderer for eksempel til copulatettheten $c_{13|2}$, og hele dekomponeringen er definert ved de $n(n - 1)/2$ kantene og marginaltetthetene til hver variabel. Nodene brukes kun for å

betemme kantene i neste nivå. Denne trestrukturen er ikke nødvendig for å bruke par-copula-metodologien, men er et nyttig verktøy for å identifisere de ulike par-copula-dekomposisjonene.

Tettheten $f(x_1, \dots, x_n)$ som svarer til en D-vine kan skrives som

$$\prod_{k=1}^n f(x_k) \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{i,i+j|i+1,\dots,i+j-1}(F(x_i|x_{i+1}, \dots, x_{i+j-1}), \dots, F(x_j|x_{i+1}, \dots, x_{i+j-1})), \quad (5.8)$$

hvor indeks j identifiserer nivåene, mens i løper over kantene i hvert nivå. Tilsvarende, er den n -dimensjonale tettheten til en C-vine gitt ved

$$\prod_{k=1}^n f(x_k) \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{j,j+i|1,\dots,j-1}(F(x_j|x_1, \dots, x_{j-1}), \dots, F(x_{j+i}|x_1, \dots, x_{j-1})). \quad (5.9)$$

For D-vines bestemmer arrangementen av nodene i det første nivået, T_1 , hele konstruksjonen. For en n -dimensjonal D-vine kan denne arrangeres på $n!$ måter. Siden kantene er uten retning; $c_{ij|D} = c_{ji|D}$, kan vi reversere rekkefølgen i T_1 uten at vi forandrer den korresponderende vinen, og vi har derfor bare $\frac{n!}{2}$ forskjellige første nivå, og $\frac{n!}{2}$ ulike D-vines med n noder.

For en kanonisk vine er det slik at betingingen på alle kanter i hvert nivå er den samme. I T_2 på figur 5.2 betinges det for eksempel på 1. Utvides dette til n noder ser man at det er n valg for betingelsessettet $\{b_2\}$ i T_2 , $n-1$ valg for betingelsessettet $\{b_2, b_3\}$ i T_3 når $\{b_2\}$ er valgt i T_2 osv. Tilsammen har vi $n(n-1) \cdots 3 = \frac{n!}{2}$ ulike kanoniske vines med n noder.

5.3 Simulering

Simulering fra C-vines og D-vines er relativt rett fram og er basert på følgende simulerings-algoritme for n avhengige $U(0,1)$ -variable. Trekk først n uavhengige uniforme tilfeldige tall, $u_i \in [0, 1]$, og sett

$$\begin{aligned} x_1 &= u_1 \\ x_2 &= F_{2|1}^{-1}(u_2|x_1) \\ x_3 &= F_{3|2,1}^{-1}(u_3|x_1, x_2) \\ x_4 &= F_{4|1,2,3}^{-1}(u_4|x_1, x_2, x_3) \\ &\vdots \\ x_n &= F_{n|1,2,\dots,n-1}^{-1}(u_n|x_1, \dots, x_{n-1}). \end{aligned}$$

For å implementere denne algoritmen må vi ha uttrykk for de betingede fordelingene $F_{x|\mathbf{v}}(x|\mathbf{v})$ og deres inverse. Simuleringsalgoritmen er i prinsipp den samme for D-vines og C-vines. Forskjellen mellom de to vinene er hvordan man velger v_j -variabelen når man skal finne de betingede fordelingene ved ligning (5.4). For en C-vine velger vi alltid den siste betingelsesvariabelen,

$$F(x_j|x_1, x_2, \dots, x_{j-1}) = \frac{\partial C_{j,j-1|1,\dots,j-2}(F(x_j|x_1, \dots, x_{j-2}), F(x_{j-1}|x_1, \dots, x_{j-2}))}{\partial F(x_j|x_1, \dots, x_{j-2})},$$

mens man for en D-vine alltid velger den første betingelsesvariabelen;

$$F(x_j|x_1, x_2, \dots, x_{j-1}) = \frac{\partial C_{j,1|2,\dots,j-1}(F(x_j|x_1, \dots, x_{j-2}), F(x_1|x_2, \dots, x_{j-1}))}{\partial F(x_j|x_2, \dots, x_{j-1})}.$$

Algoritme 1 og 2, hentet fra [1], gir kode for å generere realisjoner fra henholdsvis en C-vine og en D-vine. Algoritme 2 finnes også i appendiks C. Her er marginalfordelingene for enkelhets skyld antatt uniformt fordelte, men det er enkelt å utvide til andre marginalfordelinger.

5.4 Estimering

Når man har bestemt passende par-copula-familier for dataene man jobber med, kan man utføre parameterestimeringen med maximum likelihood. For en D-vine er likelihood-funksjonen og log-likelihood-funksjonen gitt ved

$$L = \prod_{t=1}^t \prod_{k=1}^n f(x_k) \\ \times \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{i,i+j|i+1,\dots,i+j-1}(F(x_i|x_{i+1}, \dots, x_{i+j-1}), \dots, F(x_j|x_{i+1}, \dots, x_{i+j-1})),$$

og

$$l = \sum_{t=1}^t \sum_{k=1}^n \log(f(x_k)) \tag{5.10}$$

$$+ \sum_{j=1}^{n-1} \sum_{i=1}^{n-j} \log \left(c_{i,i+j|i+1,\dots,i+j-1}(F(x_i|x_{i+1}, \dots, x_{i+j-1}), \dots, F(x_j|x_{i+1}, \dots, x_{i+j-1})) \right), \tag{5.11}$$

og tilsvarende for en C-vine:

$$L = \prod_{t=1}^T \prod_{k=1}^n f(x_k) \\ \times \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{j,j+i|1,\dots,j-1}(F(x_j|x_1,\dots,x_{j-1}),\dots,F(x_{j+i}|x_1,\dots,x_{j-1}))$$

og

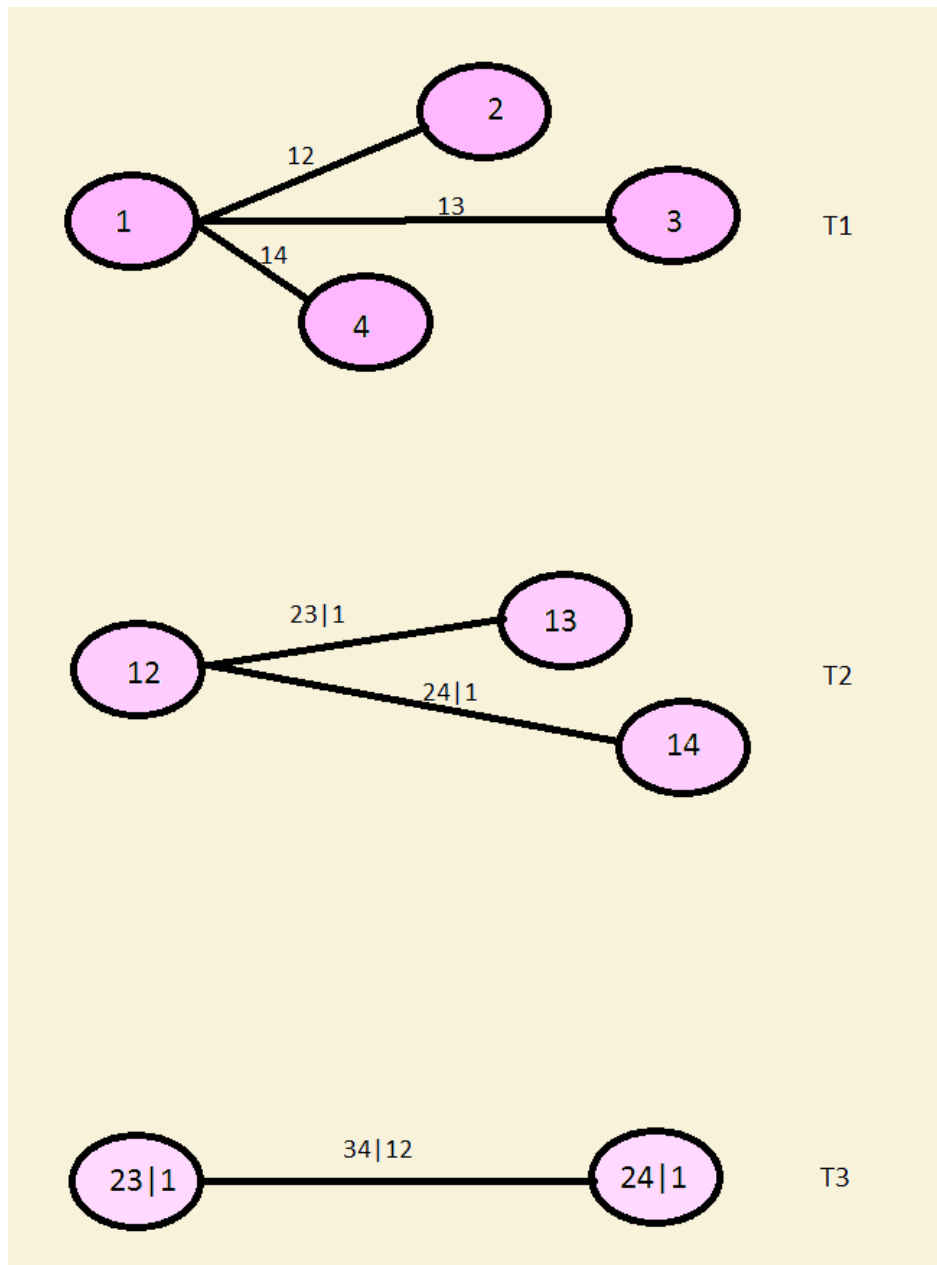
$$l = \sum_{t=1}^T \sum_{k=1}^n \log(f(x_k)) \\ + \sum_{j=1}^{n-1} \sum_{i=1}^{n-j} \log \left(c_{j,j+i|1,\dots,j-1}(F(x_j|x_1,\dots,x_{j-1}),\dots,F(x_{j+i}|x_1,\dots,x_{j-1})) \right). \quad (5.12)$$

For hver copula i (5.11) og (5.12) er det minst en parameter som skal estimeres, avhengig av hvilken copula-type man bruker. Log-likelihoodfunksjonene må optimeres numerisk, og algoritme 3 og algoritme 4, hentet fra [1], er rutiner som evaluerer likelihooden for henholdsvis en C-vine og en D-vine.

Disse algoritmene er ofte sensitive for initialverdier, og man er avhengig av å finne verdier som er rimelig nær den sanne verdien når man optimerer. En vanlig måte å bestemme startverdier på er ved en sekvensiell algoritme der man estimerer parameterene i par-copula-konstruksjonene nivå for nivå. Dette gjøres på følgende måte:

- (i) Estimer parameterene til copulaene i nivå 1 fra originaldataene.
- (ii) Beregn observasjoner for nivå 2 (betingede fordelingsfunksjoner) ved å bruke copula-parameterene fra nivå 1 og h -funksjonen.
- (iii) Estimer parameterene for copulaene i nivå 2 ved å bruke observasjonene fra (ii).
- (iv) Beregn observasjoner for nivå 3 ved å bruke parameterene fra nivå 2 og h -funksjonen.
- (v) Estimer parametere for copulaen i nivå 3 ved å bruke observasjonene fra (iv).
- (vi) Fortsett.

Disse estimeringene er enkle og raske siden man kun har to dimensjoner for hver copula.



Figur 5.2: Kanonisk vine i 4 dimensjoner.

Kapittel 6

PCC-DCC-GARCH

I dette kapitlet vil jeg, på samme måte som for multivariat Student's t-copula i Kapittel 3, vise hvordan man kan kombinere teorien om par-copula-konstruksjoner med DCC-GARCH modellen for å gjøre modellen enda mer generell. Jeg kommer i mine modeller til å bruke et D-vine-oppsett, men dersom dataene man jobber med passer bedre til et C-vine-oppsett er det ikke store endringer som må gjøres. Kombinert med GARCH modeller for de marginale fordelingene til avkastningene, er en D-vine et veldig fleksibelt verktøy for å modellere avhengighet.

En utfordring med denne modellen er at man må estimere og simulere fra betingede copulaer. For å få dynamiske parametere ikke bare i første nivå, men også i de betingede copulaene, er man avhengig av å finne en metode som beregner de ønskede parameterene ut fra korrelasjonsmatrisen, \mathbf{R}_t , som for hvert tidspunkt, t , oppdateres ved (2.8) og (2.9).

Å beregne betinget korrelasjon kan være komplisert. Partiell korrelasjon kan derimot enkelt beregnes ut fra korrelasjoner med følgende rekursive formel fra [32]:

$$\rho_{12 \cdot 3, \dots, n} = \frac{\rho_{12 \cdot 3, \dots, (n-1)} - \rho_{1n \cdot 2, \dots, (n-1)} \rho_{2n \cdot 1, 3, \dots, (n-1)}}{\sqrt{1 - \rho_{1n \cdot 2, \dots, (n-1)}^2} \sqrt{1 - \rho_{2n \cdot 1, 3, \dots, (n-1)}^2}}. \quad (6.1)$$

I flere tilfeller vil den partielle korrelasjonen være lik den betingede. I [20] bevises blant annet at dette gjelder for elliptiske copulaer. Selv om det ikke er helt riktig velger jeg for enkelthets skyld også å anta at det gjelder for alle copulaene jeg bruker i denne oppgaven.

I tilfeller der korrelasjonen er en egen parameter i copulaen (Student's t og Gaussisk copula), lar jeg denne variere med tiden, mens andre parametere, som for eksempel frihetsgradene i t-copulaen, holdes konstant. Brukes derimot copulaer som ikke har korrelasjon som en av parameterene, er forholdet mellom copulaparameterene og den lineære korrelasjonen gitt ved

likningene (3.14) og (3.13). Når man har beregnet korrelasjonen, kan man altså gå veien om Kendall's tau for å finne verdier for de ønskede copulaparameterene. Dette er illustrert i figur 1, s.14 i [15].

6.1 Estimering av parametere

På samme måte som i Kapittel 3, vil vi for en PCC-DCC-modell få en likelihood på formen

$$l(\Phi, \Psi) = \sum_{i=1}^d l_{i,m}(\Phi_i) + l_c(\Psi),$$

der Φ_i er parametrene α_{i0} , α_{i1} og β_i fra de univariate GARCH-modellene, samt parameteren ν_i som svarer til frihetsgradene i de korresponderende t-fordelingene, mens Ψ inneholder parametrene a og b og de resterende copulaparameterene.

Estimering av de univariate modellene er identisk som for Copula-DCC-GARCH-modellen. Jeg tilpasser hver av modellene ved å optimere likelihoodfunksjonen gitt i ligning (4.5). Når disse verdiene er estimert, behandles de som kjent når jeg skal finne parameterene i par-copula-konstruksjonen.

I fire dimensjoner vil en D-vine-konstruksjon ha følgende tetthet:

$$\begin{aligned} c(u_1, \dots, u_4) &= c_{12}(u_1, u_2)c_{23}(u_2, u_3)c_{34}(u_3, u_4) \\ &\quad \times c_{13|2}(F(u_1|u_2), F(u_3|u_2))c_{24|3}(F(u_2|u_3), F(u_4|u_3)) \\ &\quad \times c_{14|23}(F(u_1|u_2, u_3), F(u_4|u_2, u_3)). \end{aligned} \quad (6.2)$$

Her korresponderer hver linje i (6.2) til et nivå i D-vinen, og de betingede fordelingsfunksjonene beregnes iterativt ved ligning (5.4). Log-likelihood-funksjonen, $l_c(\Psi)$, kan skrives som

$$\begin{aligned} l_c(\Psi) &= \sum_{t=1}^T \log(c_{12}(u_1, u_2)c_{23}(u_2, u_3)c_{34}(u_3, u_4)) \\ &\quad + \sum_{t=1}^T \log(c_{13|2}(F(u_1|u_2), F(u_3|u_2))c_{24|3}(F(u_2|u_3), F(u_4|u_3))) \\ &\quad + \sum_{t=1}^T \log(c_{14|23}(F(u_1|u_2, u_3), F(u_4|u_2, u_3))), \end{aligned} \quad (6.3)$$

og det er denne det optimeres over for å finne parameterene inneholdt i Ψ . Algoritme 4 er en evaluering av likelihooden i n dimensjoner, og med noen justeringer for å få inn aspektet

med tidsvarierende korrelasjon, er det denne jeg optimerer for å få estimater for copulaparameterene. For hver t oppdateres Θ -verdiene i algoritmen ved å først beregne \mathbf{R}_t og deretter bruker denne til å finne de betingede korrelasjonene og eventuelt andre dynamiske copulaparameterene. Disse behandles da som konstanter når de sendes inn i likelihoodfunksjonen for dette tidspunktet. Copula-parameterene som ikke skal variere med tiden optimeres på vanlig vis.

Denne algoritmen er sensitiv til startverdier. Som nevnt tidligere, er det vanlig å bruke en sekvensiell algoritme for å finne startverdier for parameterene som er nær de sanne verdiene. I den sekvensielle algoritmen optimeres hver linje i (6.3) hver for seg som beskrevet i avsnitt 5.4. For å få startverdier også for a og b , og ikke bare copula-parameterene, lar jeg korrelasjonsmatrisen være dynamisk i første nivå. I de resterende nivåene holdes den konstant.

6.1.1 D-vine med bivariat Student's t copula for alle par

Tidligere studier, som blant annet [1], viser at for finansielle data, vil en konstruksjon der man har bivariate Student's t-copulaer for alle par ofte være en passende modell.

Tettheten til den bivariate Student's t copulaen er gitt ved

$$c(u_i, u_j) = \frac{\Gamma(\frac{\nu_{ij}+2}{2})/\Gamma(\frac{\nu_{ij}}{2})}{\nu_{ij} \pi t_{\nu_{ij}}(x_i) t_{\nu_{ij}}(x_j) \sqrt{1-\rho_{ij}^2}} \left(1 + \frac{x_i^2 + x_j^2 - 2\rho_{12}x_i x_j}{\nu_{ij}(1-\rho_{ij}^2)} \right)^{-\frac{\nu_{ij}+2}{2}}, \quad (6.4)$$

der $x_i = t_{\nu_{ij}}^{-1}(u_i)$ og $x_j = t_{\nu_{ij}}^{-1}(u_j)$, og den korresponderende likelihoodfunksjonen er

$$\begin{aligned} L_{c_{\nu_{ij}}}(\Theta) &= \prod_{t=1}^T c(u_i, u_j) \\ &= \prod_{t=1}^T \frac{\Gamma(\frac{\nu_{ij}+2}{2})/\Gamma(\frac{\nu_{ij}}{2})}{\nu_{ij} \pi t_{\nu_{ij}}(x_i) t_{\nu_{ij}}(x_j) \sqrt{1-\rho_{ij}^2}} \left(1 + \frac{x_i^2 + x_j^2 - 2\rho_{12}x_i x_j}{\nu_{ij}(1-\rho_{ij}^2)} \right)^{-\frac{\nu_{ij}+2}{2}} \\ l_{c_{\nu_{12}}}(\Theta) &= \sum_{t=1}^T \left\{ \log \left(\Gamma \left(\frac{\nu_{ij}+2}{2} \right) \right) - \log \left(\Gamma \left(\frac{\nu_{ij}}{2} \right) \right) - \log(\nu_{ij} \pi) - \log(t_{\nu_{ij}}(x_i)) \right. \\ &\quad \left. - \log(t_{\nu_{ij}}(x_j)) - \frac{1}{2} \log(1-\rho_{ij}^2) \right. \\ &\quad \left. - \frac{\nu_{ij}+2}{2} \log \left(1 + \frac{x_i^2 + x_j^2 - 2\rho_{12}x_i x_j}{\nu_{ij}(1-\rho_{ij}^2)} \right) \right\}. \end{aligned} \quad (6.5)$$

For en modell der man har Student's t-copula for alle par i PCC-en er det tettheten til denne som settes inn for alle $L(\cdot)$ i algoritme 4. Korrelasjonene i (6.5) antas nå å variere med tiden. I første nivå hentes korrelasjonene rett ut fra korrelasjonsmatrisen, \mathbf{R}_t , mens for de betingede copulaene beregnes korrelasjonene ved å bruke (6.1) på \mathbf{R}_t . Frihetsgradene antas å være konstante over tid.

6.1.2 D-vine med både Student's t copula og Clayton copula

I noen tilfeller kan man tenke seg at man har en portefølje med større sannsynlighet for samtidige negative utslag enn samtidige positive utslag. Når dette er tilfelle kan det være lurt å bytte ut Student's t-copulaen med Clayton-copulaen for noen par av aktiva.

I [15] nevnes det at korrelasjon betinget på positive avkastninger typisk er lavere enn korrelasjon betinget på negative avkastninger. Dette er trender som kan fanges opp av Clayton-copulaen.

Hvilke par man skal tilpasse til hvilken copula kan bestemmes ut fra en kvantitativ analyse av dataene. Dersom man først transformerer dataene over til copulaskalaen ved å bruke de estimerte marginalfordelingene, kan man, ut fra et scatterplott av parene, se om det for noen av parene ser ut til å være assymetri i avhengighetsstrukturen.

Tettheten til den bivariate Clayton copulaen er gitt ved

$$c(u_i, u_j) = (1 + \delta)(u_i u_j)^{-1-\delta}(u_i^{-\delta} + u_j^{-\delta} - 1)^{-1/\delta-2}, \quad (6.6)$$

og den gir følgende likelihood- og log-likelihood-funksjon:

$$\begin{aligned} L_{c_{vij}}(\Theta) &= \prod_{t=1}^T c(u_i, u_j) \\ &= \prod_{t=1}^T (1 + \delta)(u_i u_j)^{-1-\delta}(u_i^{-\delta} + u_j^{-\delta} - 1)^{-1/\delta-2} \\ l_{c_{vij}}(\Theta) &= \sum_{t=1}^T \{\log((1 + \delta)(u_i u_j)^{-1-\delta}(u_i^{-\delta} + u_j^{-\delta} - 1)^{-1/\delta-2}) \\ &= \sum_{t=1}^T \{\log((1 + \delta)) - (1 + \delta) \log(u_i u_j)\} - (1/\delta + 2) \log(u_i^{-\delta} + u_j^{-\delta} - 1)\}. \end{aligned} \quad (6.7)$$

Ved å bytte ut likelihooden for Student's t-copulaen med (6.7) for de aktuelle parene i algoritme 4, vil man få en algoritme som evaluerer likelihooden for en PCC med både Student t-copulaer og Clayton-copulaer.

Clayton-copulaen har ikke korrelasjon som en av sine parametere, og som nevnt tidligere må man dermed gå veien om Kendall's tau for å finne en verdi for δ_t som korresponderer til den tidsvarierende korrelasjonen. For Clayton-copulaen har vi

$$\delta_t = 2\tau_t/(1 - \tau_t), \quad (6.8)$$

hvor

$$\tau_t = 2 \arcsin(\rho_t)/\pi. \quad (6.9)$$

Denne ikke-lineære transformasjonen fra ρ_t til τ_t er en del av hvordan dynamikken i modellen spesifiseres. Transformasjonen i likning (6.9) gjelder strengt tatt kun for elliptiske copulaer, men i [15] brukes den for alle copulaer uten at avvikene kommenteres, så jeg velger å anta at det kan gjøres også i mitt tilfelle.

Clayton-copulaen er begrenset til å modellere positiv avhengighet. Copulaen er bare gyldig for $\delta > 0$, noe som betyr at bruk av Clayton-copulaen for modellering av avhengighetsstruktur begrenser seg til tilfeller med positiv korrelasjon. Negativ korrelasjon gir negativ verdi for δ når man utfører transformasjonen. Som i [15] erstattes i disse tilfellene q_t , "ikke-diagonal"-elementene i Q_t i ligning (2.9), med $\max(q_t, 0)$.

“Dette er en noe “ad hoc” måte å garantere at modellen er veldefinert i alle situasjoner på, men ansees ikke som et problem, siden man ikke vil ende opp med å velge slike modeller.”
(oversatt fra [15])

6.2 Simulering

Når man vet hvilke parametere og hvilke copulaer man vil bruke, simulerer man fra en PCC-DCC-GARCH-modell på følgende måte:

For hvert tidspunkt t :

- (i) Velg startverdi for volatiliteten i de univariate rekkene (siste estimerte varians, eller stasjonærvariansen).
- (ii) Beregn volatiliteten i de univariate rekkene med (2.4) og lag matrisen \mathbf{D}_t .
- (iii) Beregn korrelasjonsmatrisen, \mathbf{R}_t , med (2.8) og (2.9).
- (iv) Beregn korrelasjonen for de betingede copulaene med (6.1).
- (v) Finn Kendall's tau med (6.9).
- (vi) Beregn copulaparameterene med (3.13).
- (vii) Generer en tilfeldig trekning fra den d-dimensjonale par-copula-konstruksjonen med de gitte parameterene.
- (viii) Beregn $\boldsymbol{\epsilon}_t$ ved å benytte de inverse kumulative fordelingsfunksjonene på de uniforme variablene fra copulaen.
- (ix) Beregn de forventningskorrigerede log-avkastningene; $\mathbf{a}_t = \mathbf{D}_t \boldsymbol{\epsilon}_t$.

Kapittel 7

Prediksjon og Validering

I dette kapittelet skal jeg se på hva estimatene fra DCC-modellene kan brukes til. Jeg skal også beskrive tester for å vurdere de ulike modellene opp mot hverandre på et gitt datasett.

7.1 Prediksjon

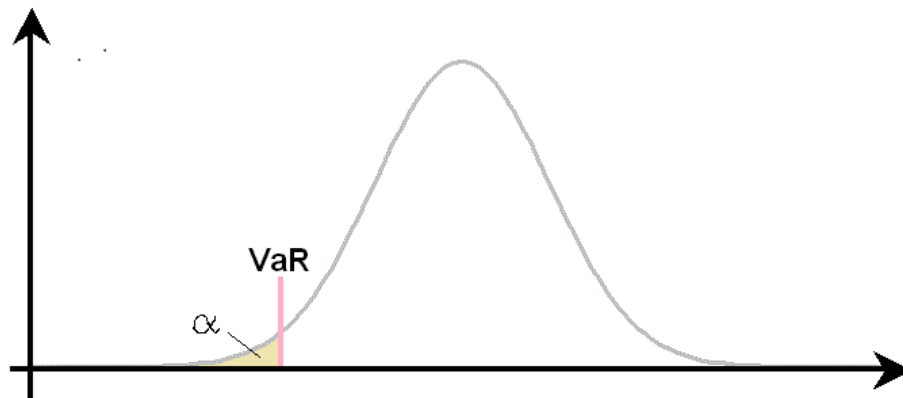
Når parameterene i de ulike modellene er estimert, kan disse estimatene brukes til å predikere den fremtidige kovariansmatrisen, $\mathbf{H}_{t+k} = \mathbf{D}_{t+k}\mathbf{R}_{t+k}\mathbf{D}_{t+k}$, for porteføljen. Dette gjøres ved å predikere \mathbf{D}_{t+k} og \mathbf{R}_{t+k} hver for seg. For detaljer rundt hvordan dette kan gjøres i praksis se [22].

En annen form for fremtidsprediksjon er “Value at Risk”. Value at Risk er et mye brukt mål for å oppsummere risikoen i en portefølje. Value at Risk kan sees på som en kvantil i tapsfordelingen til en portefølje og er definert som den største verdien, l , som er slik at sannsynligheten for at tapet er større enn l er $1 - \alpha$:

$$VaR_\alpha = \inf\{l \in \mathbb{R} : F_L(l) \geq \alpha\}.$$

Her er F_L den gitte tapsfordelingen som man vil finne Value at Risk for. Alternativt kan man finne VaR for fordelingen til avkastningene. Da blir VaR_α den minste verdien, a , som er slik at sannsynligheten for at avkastningen er mindre enn a er $1 - \alpha$. Figur 7.1 illustrerer VaR for en avkastningsfordeling. Typiske verdier for α er $\alpha = 0.95$ eller $\alpha = 0.99$.

En svakhet med dette målet på risiko er mangelen på subadditivitet. Gitt (L_1, VaR_1) og (L_2, VaR_2) er det ikke gitt at $VaR(L_1 + L_2) \leq VaR_1 + VaR_2$. Man er altså avhengig av å ha



Figur 7.1: Value at Risk

et uttrykk for fordelingen til den samlede avkastningen for hele porteføljen for at Value at Risk skal være et fornuftig mål på risikoen. Den samlede avkastningen er gitt ved $p_t = \omega^T \mathbf{a}_t$, hvor ω^T er en vektor med vekter for de ulike aktivaene.

En annen svakhet er at Value at Risk ikke sier noe om størrelsen på de tapene som skjer med en sannsynlighet på $1 - \alpha$. Vi vet at tapene er større enn en gitt verdi med en gitt sannsynlighet, men ikke noe om *hvor* store de er. Flere har derfor foreslått å bruke Expected Shortfall som risikomål i stedet for Value at Risk. Expected Shortfall defineres som forventet tap gitt at Value at Risk overskrides. Men siden VaR er det mest brukte målet i praksis (blant annet i Solvens II [23]), velger jeg å bruke det her.

Jeg kommer tilbake til hvordan VaR beregnes for de tre DCC-GARCH-modellene når jeg skal bruke modellene på ekte data i kapittel 8.

7.2 Modellvalidering

Jeg har i denne oppgaven listet opp modeller hvor jeg hele tiden har argumentert for at den ene er bedre enn den andre. Denne argumentasjonen baserer seg for det meste på logikk, intuisjon og andres erfaringer, men andre argumenter enn magesfølelse og erfaring er gjerne nyttig hvis man også skal kunne overbevise andre enn seg selv. Jeg vil i dette avsnittet presentere et par metoder for å sammenligne hvor godt ulike modeller beskriver de samme dataene. Det er viktig å være klar over at ulike datasett har ulike egenskaper, og selv om en modell er den beste for et gitt datasett, er det ikke gitt at det er den beste for et annet.

7.2.1 AIC

En metode som er mye brukt for modellseleksjon, spesielt blant regressjonsmodeller, er Akaikes Informasjonskriterie (AIC). Akaikes Informationskriterie er en metode for å velge den modellen som best beskriver dataene fra et sett av modeller. Den valgte modellen (modellen med lavest AIC-verdi) er den modellen som minimerer Kullback-Leibler avstanden mellom modellen og sannheten[28]. AIC beregnes ved ligningen

$$AIC_i = -2 \log L_i(\hat{\theta}_i|x) + 2k_i, \quad (7.1)$$

hvor $L_i(\cdot)$ er likelihoodfunksjonen til den i -te modellen og k_i er antall parametere. For at AIC skal være gyldig må antall observasjoner være stor sammenlignet med k .

AIC er mest robust for nestede modeller, men en fordel med AIC er at det kan brukes til å sammenligne både nøstede og ikke-nøstede modeller. Av modellene som er foreslått i denne oppgaven er modellen med multivariat Student t copula og modellen med bivariate Student t copulaer for alle par i par-copula-konstruksjonen nøstede siden multivariat Student's t copula er et spesialtilfelle av PCC-en [14]. Den siste modellen er imidlertid ikke nøstet med de andre.

AIC gir ikke mening i seg selv, men for å sammenligne ulike modeller for de samme dataene er den interessant. Ulempen med AIC er at den ikke gir noen advarsel dersom alle modellene passer dårlig til dataene. I tilfeller med bare dårlige modeller, vil AIC-verdien bare hjelpe til å velge den som er "minst dårlig". Det kreves derfor at man gjør grundige vurderinger før man setter opp modellene. I mange tilfeller er det lurt å utføre såkalte "goodness of fit"-tester i tillegg til AIC. Jeg vil imidlertid ikke komme inn på slike tester i denne oppgaven.

7.2.2 Kupiec test og Christoffersen's Markov test

En annen måte å teste modeller på er å undersøke hvor gode de er til å estimere Value at Risk. For å teste dette, kan man bruke "backtesting". Man deler dataene inn i et estimeringssett og et testsett og bruker estimeringssettet til å beregne VaR-verdier for testsett-perioden. Man beregner så hvor stor prosent av porteføljeavkastningene, p_t , som har verdier lavere enn VaR-estimatene. Denne prosentandelen sammenlignes med signifikansnivået som Value at Risk er beregnet med.

Dersom modellen er riktig spesifisert skal følgende to egenskaper være oppfylt [22]:

1. Antall avvik skal være lik det forventede antallet.
2. Avvikene skal være uavhengig fordelte over tid.

Kuipec testen ble introdusert i [27] og er en test for å avgjøre om den observerte frekvensen av verdier som faller under VaR er konsistent med den forventede frekvensen [6]. Denne tester altså punkt 1, mens punkt 2 kan testes med Christoffersen's Markov test, introdusert i [9].

Kuipec test

Oppsettet for Kuipic testen er følgende:

| | |
|---------------------|--|
| Nullhypotese: | H_0 : Forventet andel overskridelser av VaR er lik α . |
| Alternativhypotese: | H_1 : Forventet andel overskridelser av VaR er ikke lik α . |

Testobservator:

$$Kuipec = 2 \ln \left(\left(\frac{x}{N} \right)^x \left(1 - \frac{x}{N} \right)^{N-x} \right) - 2 \ln(\alpha(1 - \alpha)^{N-x}), \quad (7.2)$$

hvor N er antall observasjoner, x er antall overskridelser og α er signifikansnivået som VaR er beregnet under. Testobservatoren er asymptotisk $\chi^2(1)$ -fordelt.

Dersom den estimerte p-verdien er over signifikansnivået (vanligvis 5 %), aksepterer vi modellen. Dersom p-verdien er lavere enn signifikansnivået, forkaster vi modellen og konkluderer med at den ikke passer til det gitte datasettet.

Christoffersen's Markov test

Christoffersen's Markov test har følgende oppsett:

| | |
|---------------------|---|
| Nullhypotese: | H_0 : Overskridelsene er uavhengig fordelt over tid. |
| Alternativhypotese: | H_1 : Overskridelsene er ikke uavhengig fordelt over tid. |

Testobservator:

$$Chris = 2 \ln((1 - \pi_{01})^{A_{00}} \pi_{01}^{A_{01}} (1 - \pi_{11})^{A_{10}} \pi_{11}^{A_{11}}) - 2 \ln((1 - \pi)^{A_{00} + A_{10}} \pi^{A_{01} + A_{11}}). \quad (7.3)$$

Hvis vi definerer $I_t = 0$ for $p_t \geq VaR_t$ og $I(\alpha)_t = 1$ ellers, så er A_{ij} antall opservasjoner av I_t med verdi i etterfulgt av j ; $i, j=0, 1$. $\pi_{ij} = \frac{A_{ij}}{\sum_{j=0}^1 A_{ij}}$ og $\pi = \frac{A_{01} + A_{11}}{A}$. Også denne testobservatoren er asymptotisk $\chi^2(1)$ -fordelt.

En svakhet med Christoffersen's Markov test er at den kun ser på avhengighet fra en dag til den neste, altså at markovegenskapen holder.

Kapittel 8

Robusthetsstudier

Før modellene blir tatt i bruk på ekte data gjøres testkjøringer hvor jeg for hver modell simulerer 10000 datapunkter for deretter å estimere meg tilbake til simuleringsestimatene. Dette for å forsikre meg om at algoritmene er riktig implementert. Et utvalg av testkjøringene finnes i Appendiks A.

Jeg får uventede resultater i noen av testkjøringene. For PCC-DCC-modellen med Student's t copula for alle par har jeg i tilfeller med høy a-verdi fått unøyaktige estimat for frihetsgradene, spesielt på 2. og 3.nivå i par-copula konstruksjonen. Dette er ikke kommentert i noen av de tidligere studiene jeg har lest og jeg vil derfor i dette kapitlet gjøre et studie av robustheten til modellene. Det kjøres tester for ulike parametervalg for å undersøke hva unøyaktighetene skyldes. Det første jeg tester er om verdiene på DCC-parameterene, a og b, har noe å si for nøyaktigheten til de estimerte frihetsgradene.

8.1 Variasjoner i DCC-parameterene

Ut fra tidligere estimeringsforsøk kan det, som sagt, virke som om dette er tilfelle for PCC-DCC-modellen med Student's t copula for alle par. Selv om jeg ikke har lagt merke til det samme for de to andre modellene, velger jeg å kjøre tester på disse også.

Følgende univariate parametere er brukt i alle testforsøk:

| | α_0 | α_1 | β | ν |
|---------|------------------------|------------|---------|-------|
| rekke 1 | 2.28×10^{-06} | 0.116 | 0.854 | 4 |
| rekke 2 | 1.39×10^{-06} | 0.091 | 0.903 | 6 |
| rekke 3 | 3.99×10^{-06} | 0.117 | 0.863 | 8 |
| rekke 4 | 2.57×10^{-06} | 0.095 | 0.899 | 10 |

8.1.1 Modell 1: Copula-DCC-GARCH

Jeg gjør 20 simuleringer der jeg gradvis øker a-verdien og minker b-verdien. For hver a- og b-verdi gjør jeg 5000 trekninger med $\nu_C = 7$ i alle. Deretter kjører jeg estimeringsalgoritmen for å se om jeg får verdier nær de jeg simulerer med. Resultatene vises i tabell 8.1, og jeg konkluderer ut fra den med at nøyaktigheten til estimatet for frihetsgradene i den multivariate Student's t copulaen er uavhengig av verdien på DCC-parameterene.

| a | b | \hat{a} | \hat{b} | ν_C |
|-------|------|-----------|-----------|---------|
| 0.005 | 0.9 | 0.00777 | 0.90286 | 6.746 |
| 0.015 | 0.89 | 0.0166 | 0.90111 | 6.679 |
| 0.025 | 0.88 | 0.02308 | 0.87671 | 6.643 |
| 0.035 | 0.87 | 0.03365 | 0.84019 | 7.549 |
| 0.045 | 0.86 | 0.04664 | 0.86025 | 7.341 |
| 0.055 | 0.85 | 0.0578 | 0.8349 | 6.969 |
| 0.065 | 0.84 | 0.06572 | 0.84641 | 6.419 |
| 0.075 | 0.83 | 0.07659 | 0.83692 | 6.716 |
| 0.085 | 0.82 | 0.08861 | 0.8058 | 6.795 |
| 0.095 | 0.81 | 0.10376 | 0.80848 | 6.657 |
| 0.105 | 0.80 | 0.11047 | 0.80113 | 6.928 |
| 0.115 | 0.79 | 0.10778 | 0.80861 | 7.291 |
| 0.125 | 0.78 | 0.12058 | 0.79607 | 6.845 |
| 0.135 | 0.77 | 0.12295 | 0.78334 | 6.865 |
| 0.145 | 0.76 | 0.14708 | 0.7661 | 6.917 |
| 0.155 | 0.75 | 0.1476 | 0.7632 | 7.013 |
| 0.165 | 0.74 | 0.16259 | 0.74193 | 6.404 |
| 0.175 | 0.73 | 0.16645 | 0.74293 | 6.663 |
| 0.185 | 0.72 | 0.20169 | 0.7055 | 6.885 |
| 0.195 | 0.71 | 0.19884 | 0.70847 | 6.685 |
| 0.205 | 0.70 | 0.19859 | 0.70799 | 6.604 |

Tabell 8.1: Estimerte parametere fra 20 C-DCC-GARCH-simuleringer med varierende DCC-parametere.

8.1.2 Modell 2: PCC-DCC-GARCH med Student t for alle par

Jeg gjør det samme for PCC-DCC-modellen. Her har jeg frihetsgrader 3, 6, 9, 12, 15 og 18 i alle forsøk og varierer estimatene for a og b på samme måte som for C-DCC- modellen. Dette gir meg tabell 8.2.

I denne tabellen ser det ut som at nøyaktigheten av parameterene i par-copula-konstruksjonen blir mer påvirket av DCC-parameterene. Frihetsgradene på nivå 2 og 3 blir tilsynelatende litt og litt dårligere estimert jo lenger ned i tabellen man kommer. En grunn til dette kan være at man for høy verdi for a trenger flere datapunkter for å få et mer nøyaktig resultat. For å sjekke om det er tilfellet, gjør jeg en kjøring med 100000 simuleringer for det siste parametersettet ($a = 0.205$ og $b = 0.70$). Resultatene av denne er vist i tabell 8.3.

Det ser ikke ut til at det hjelper å doble antall datapunkter. Estimaterne i nivå 2 og 3 er fortsatt for lave i forhold til de sanne verdiene.

| a | b | \hat{a} | \hat{b} | ν_{12} | ν_{23} | ν_{34} | $\nu_{13 2}$ | $\nu_{24 3}$ | $\nu_{14 23}$ |
|-------|------|-----------|-----------|------------|------------|------------|--------------|--------------|---------------|
| 0.005 | 0.9 | 0.00565 | 0.9255 | 2.836 | 5.636 | 9.405 | 10.529 | 15.767 | 20.768 |
| 0.015 | 0.89 | 0.01153 | 0.90671 | 2.911 | 5.586 | 8.734 | 11.203 | 17.98 | 22.039 |
| 0.025 | 0.88 | 0.02732 | 0.86742 | 3.185 | 5.006 | 8.366 | 9.939 | 16.903 | 14.416 |
| 0.035 | 0.87 | 0.03468 | 0.88371 | 2.888 | 6.22 | 8.548 | 8.622 | 12.491 | 10.326 |
| 0.045 | 0.86 | 0.04479 | 0.86551 | 3.135 | 5.763 | 8.863 | 14.318 | 16.498 | 21.612 |
| 0.055 | 0.85 | 0.05073 | 0.86843 | 2.775 | 5.493 | 8.054 | 11.085 | 11.582 | 17.73 |
| 0.065 | 0.84 | 0.06901 | 0.83809 | 2.93 | 5.572 | 10.706 | 10.46 | 11.77 | 16.071 |
| 0.075 | 0.83 | 0.08459 | 0.80994 | 2.903 | 5.546 | 9.653 | 8.371 | 10.715 | 16.079 |
| 0.085 | 0.82 | 0.08944 | 0.82373 | 3.143 | 5.555 | 8.333 | 10.112 | 14.847 | 13.816 |
| 0.095 | 0.81 | 0.09655 | 0.81414 | 3.141 | 5.688 | 18.656 | 11.518 | 13.01 | 15.901 |
| 0.105 | 0.80 | 0.109 | 0.80403 | 2.94 | 5.554 | 8.449 | 8.429 | 8.926 | 14.993 |
| 0.115 | 0.79 | 0.11733 | 0.78919 | 2.91 | 5.408 | 7.49 | 8.474 | 13.642 | 17.515 |
| 0.125 | 0.78 | 0.1346 | 0.77977 | 2.843 | 5.089 | 17.129 | 7.883 | 12.133 | 14.902 |
| 0.135 | 0.77 | 0.13859 | 0.77565 | 2.988 | 5.524 | 9.459 | 8.28 | 13.33 | 10.778 |
| 0.145 | 0.76 | 0.15055 | 0.76309 | 3.11 | 5.088 | 8.209 | 6.761 | 9.521 | 9.251 |
| 0.155 | 0.75 | 0.1483 | 0.76347 | 3.039 | 4.749 | 7.89 | 7.242 | 10.552 | 12.494 |
| 0.165 | 0.74 | 0.18178 | 0.73658 | 2.946 | 5.096 | 20.567 | 9.298 | 12.824 | 14.348 |
| 0.175 | 0.73 | 0.18086 | 0.72572 | 2.948 | 5.882 | 9.524 | 8.481 | 13.171 | 14.992 |
| 0.185 | 0.72 | 0.1974 | 0.71749 | 2.952 | 4.797 | 7.463 | 8.316 | 8.708 | 13.134 |
| 0.195 | 0.71 | 0.20575 | 0.71052 | 2.79 | 4.284 | 6.994 | 6.296 | 7.591 | 9.088 |
| 0.205 | 0.70 | 0.21099 | 0.7059 | 3.103 | 4.809 | 8.389 | 9.036 | 14.028 | 9.204 |

Tabell 8.2: Estimerte parametere fra 20 PCCC-DCC-GARCH-simuleringer med varierende DCC-parametere.

| a | b | \hat{a} | \hat{b} | ν_{12} | ν_{23} | ν_{34} | $\nu_{13 2}$ | $\nu_{24 3}$ | $\nu_{14 23}$ |
|-------|------|-----------|-----------|------------|------------|------------|--------------|--------------|---------------|
| 0.205 | 0.70 | 0.21826 | 0.69439 | 2.908 | 5.424 | 7.383 | 6.073 | 7.863 | 9.059 |

Tabell 8.3: Kjøring med samme parametersett som siste linje i tabell 8.2 og 10000 datapunkter

8.1.3 Modell 3: PCC-DCC-GARCH, Clayton og Student t

I den siste modellen vil jeg i tillegg til å se på hvordan DCC-parameterene påvirker estimatene, også se på hvor høy a -verdien kan være før man får negativ korrelasjon for noen av tidspunktene. Jeg har derfor, for hver kjøring, talt opp antall negative korrelasjonsverdier i optimeringsalgoritmen. Disse er lagt inn i de to siste kolonnene i tabellen. Antall negative parametere vil både være avhengig av hvor høye korrelasjoner man starter med og av størrelsen på DCC-parameterene, så jeg velger å gjøre kjøring for to forskjellige Q_1 -matriser:

$$Q_{1,1} = \begin{pmatrix} 1 & 0.7 & 0.76 & 0.3 \\ 0.7 & 1 & 0.8 & 0.4 \\ 0.76 & 0.8 & 1 & 0.3 \\ 0.3 & 0.4 & 0.3 & 1 \end{pmatrix}$$

$$Q_{1,2} = \begin{pmatrix} 1 & 0.45 & 0.6 & 0.3 \\ 0.45 & 1 & 0.8 & 0.4 \\ 0.6 & 0.8 & 1 & 0.3 \\ 0.3 & 0.4 & 0.3 & 1 \end{pmatrix}$$

Det simuleres i begge tilfeller 5000 treknninger med frihetsgrader lik 3, 6, 9 og 12. Resultatene vises i tabell B.4 og B.5.

| a | b | \hat{a} | \hat{b} | ν_{23} | $\nu_{13 2}$ | $\nu_{24 3}$ | $\nu_{14 23}$ | neg1 | neg2 |
|--------|------|-----------|-----------|------------|--------------|--------------|---------------|------|------|
| 0.0025 | 0.95 | 0.001 | 0.973 | 3.284 | 7.688 | 7.662 | 19.376 | 0 | 0 |
| 0.0125 | 0.94 | 0.015 | 0.938 | 3.134 | 6.035 | 6.816 | 10.624 | 0 | 0 |
| 0.0225 | 0.93 | 0.025 | 0.937 | 2.882 | 5.501 | 9.572 | 13.325 | 0 | 0 |
| 0.0325 | 0.92 | 0.035 | 0.921 | 3.043 | 4.721 | 8.371 | 10.794 | 0 | 0 |
| 0.0425 | 0.91 | 0.054 | 0.895 | 2.913 | 4.857 | 7.3 | 13.855 | 0 | 0 |
| 0.0525 | 0.9 | 0.056 | 0.899 | 2.802 | 6.036 | 8.806 | 8.255 | 0 | 0 |
| 0.0625 | 0.89 | 0.064 | 0.888 | 2.939 | 6.23 | 6.71 | 7.654 | 0 | 0 |
| 0.0725 | 0.88 | 0.073 | 0.885 | 2.977 | 4.955 | 9.268 | 8.189 | 0 | 0 |
| 0.0825 | 0.87 | 0.092 | 0.866 | 3.146 | 5.507 | 9.278 | 13.744 | 14 | 3 |
| 0.0925 | 0.86 | 0.098 | 0.859 | 3.094 | 5.866 | 18.269 | 14.332 | 44 | 18 |
| 0.1025 | 0.85 | 0.099 | 0.866 | 2.747 | 5.007 | 6.039 | 5.472 | 30 | 48 |
| 0.1125 | 0.84 | 0.125 | 0.832 | 2.89 | 7.454 | 7.677 | 8.772 | 50 | 15 |

Tabell 8.4: Estimerte parametere fra 12 PCCC-DCC-GARCH-simuleringer med varierende DCC-parametere og $Q_{1,1}$ som startmatrise for Q_t .

| a | b | \hat{a} | \hat{b} | ν_{23} | $\nu_{13 2}$ | $\nu_{24 3}$ | $\nu_{14 23}$ | neg1 | neg2 |
|--------|------|-----------|-----------|------------|--------------|--------------|---------------|------|------|
| 0.0025 | 0.95 | 0.005 | 0.924 | 2.819 | 5.477 | 5.822 | 8.472 | 0 | 0 |
| 0.0125 | 0.94 | 0.015 | 0.924 | 3.029 | 5.722 | 7.127 | 9.708 | 0 | 0 |
| 0.0225 | 0.93 | 0.027 | 0.931 | 3.066 | 5.856 | 7.024 | 11.466 | 0 | 0 |
| 0.0325 | 0.92 | 0.036 | 0.919 | 3.201 | 5.673 | 7.546 | 12.522 | 5 | 0 |
| 0.0425 | 0.91 | 0.043 | 0.92 | 2.946 | 5.975 | 6.386 | 17.604 | 0 | 0 |
| 0.0525 | 0.9 | 0.058 | 0.901 | 2.844 | 6.439 | 16.414 | 9.946 | 19 | 0 |
| 0.0625 | 0.89 | 0.068 | 0.883 | 3.125 | 6.145 | 6.586 | 7.556 | 52 | 1 |
| 0.0725 | 0.88 | 0.071 | 0.887 | 3.183 | 4.971 | 6.739 | 11.537 | 54 | 0 |
| 0.0825 | 0.87 | 0.087 | 0.871 | 3.11 | 5.401 | 7.332 | 19.487 | 150 | 2 |
| 0.0925 | 0.86 | 0.1 | 0.857 | 2.716 | 5.34 | 8.328 | 7.212 | 216 | 19 |

Tabell 8.5: Estimerte parametere fra 10 PCCC-DCC-GARCH-simuleringer med varierende DCC-parametere og $Q_{1,1}$ som startmatrise for Q_t .

Ut fra tabell 8.4 og 8.5 ser vi at startkorrelasjonene har en del å si for om copulaparameterene skal bevege seg utenfor gyldighetsområdet. I tilfellet med lavere korrelasjoner får vi negative parametere i copulaen når a -verdien er så lav som 0.0325. I det andre tilfellet holder antall ugyldige parametere seg lik 0 frem til $a = 0.0825$. Selv denne a -verdien er relativt lav. Det er derfor viktig å være obs på både de estimerte DCC-parameterene og den estimerte \bar{Q} -matrisen når man bruker modeller med copulaer som krever positive parametere.

Når det kommer til nøyaktigheten av parameterene i denne modellen, ser det også her ut til at frihetsgradene blir litt og litt dårligere estimert jo lenger ned i tabellen man kommer, men det er ikke like tydelig som for modell 2.

8.2 Videre testing av modell 2

For å teste om det kan være andre ting enn verdien på DCC-parameterene som gjør at estimatene blir unøyaktige, velger jeg å kjøre flere tester der jeg også varierer de andre parameterene. Jeg antar at funnene vil være de samme for de to PCC-modellene og velger å gjøre testene bare på modell 2.

Siden multivariat Student's t copula er et spesialtilfelle av en par-copula-konstruksjon med Student t- copula for alle par,(se [14]), virker det rart at den ene modellen skal bli påvirket av a - og b - estimatene, mens den andre ikke blir det. En av faktorene jeg vil teste er derfor om det kan være størrelsen eller spredningen i frihetsgradene som gjør at det blir forskjeller når DCC-parameterene varierer. En annen faktor kan være størrelse på korrelasjonen.

Siden en av faktorene jeg vil sjekke er om størrelsen på korrelasjonene har noe å si, kjører jeg de første testene to ganger, med to forskjellige startmatriser for korrelasjonen; en med relativt høye korrelasjoner, og en med lavere.

8.2.1 Korrelasjonsmatrise 1

$$Q_{1,1} = \begin{pmatrix} 1 & 0.7 & 0.76 & 0.3 \\ 0.7 & 1 & 0.8 & 0.4 \\ 0.76 & 0.8 & 1 & 0.3 \\ 0.3 & 0.4 & 0.3 & 1 \end{pmatrix}$$

En multivariat Students t copula med ν frihetsgrader tilsvarer en PCC med Student's t-

copula for alle par der frihetsgradene på første nivå er lik ν , frihetsgradene på andre nivå er lik $\nu + 1$ og frihetsgradene på tredje nivå er lik $\nu + 2$. For å utelukke at unøyaktighetene skyldes trykkfeil i simulerings- eller optimeringskodene gjør jeg en slags kryssjekk hvor jeg kjører to forsøk der jeg simulerer fra Copula-DCC-modellen med ν frihetsgrader, men bruker likelihoodfunksjonen for PCC-DCC-GARCH-modellen i estimeringen. Deretter gjør jeg det motsatte; Simulerer fra PCC og estimerer med multivariat Student's t. Hvis algoritmene er riktige skal det altså ikke ha noe å si hvilken likelihood (PCC eller multivariat Student's t-copula) man bruker når man skal estimere seg tilbake til frihetsgradene man simulerer med.

I første testforsøk velger jeg lave frihetsgrader og simulerer med $\nu = 3$, $a = 0.2$ og $b = 0.7$. Estimeringsresultatene vises i tabell 8.6. I denne tabellen vises også log-likelihood-verdien for de sanne parameterene og de estimerte verdiene.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.201 | 0.2 |
| b | 0.707 | 0.7 |
| $\nu_{c_{12}}$ | 2.78 | 3 |
| $\nu_{c_{23}}$ | 2.74 | 3 |
| $\nu_{c_{34}}$ | 3.34 | 3 |
| $\nu_{c_{13 2}}$ | 4.05 | 4 |
| $\nu_{c_{24 3}}$ | 4.89 | 4 |
| $\nu_{c_{14 23}}$ | 7.98 | 5 |
| likelihood | 8320.96 | 8333.01 |

Tabell 8.6: Simulert fra C-DCC, estimert med PCCDCC; Lave frihetsgrader, høy korrelasjon, høy a, lav b.

I tabell 8.7 vises estimeringsresultater fra en simulering med samme DCC-parametere og frihetsgrader 3, 3, 3, 4, 4 og 5. Optimeringen er gjort over likelihooden for modell 1.

Begge disse optimeringene gir gode estimater for parameterene selv om DCC-verdiene er lik de verdiene som gav unøyaktige estimater i forrige avsnitt.

| Parameter | Estimat | Forventet verdi |
|-----------|---------|-----------------|
| a | 0.200 | 0.2 |
| b | 0.707 | 0.7 |
| ν_c | 3.12 | 3 |

Tabell 8.7: Simulert fra PCCDCC, estimert med C-DCC; Lave frihetsgrader, høy korrelasjon, høy a , lav b .

Jeg øker så frihetsgradene og simulerer med 15 frihetsgrader i Student t-copulaen. De resterende parameterene er som før. Estimeringsresultatene vises i tabell 8.8 og tabell 8.9.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.206 | 0.2 |
| b | 0.709 | 0.7 |
| $\nu_{c_{12}}$ | 11.38 | 15 |
| $\nu_{c_{23}}$ | 11.61 | 15 |
| $\nu_{c_{34}}$ | 17.7 | 15 |
| $\nu_{c_{13 2}}$ | 14.22 | 16 |
| $\nu_{c_{24 3}}$ | 13.68 | 16 |
| $\nu_{c_{14 23}}$ | 16.45 | 17 |
| likelihood | 6908.76 | 6898.50 |

Tabell 8.8: Simulert fra C-DCC, estimert med PCCDCC; Høye frihetsgrader, høy korrelasjon, høy a , lav b .

| Parameter | Estimat | Forventet verdi |
|-----------|---------|-----------------|
| a | 0.206 | 0.2 |
| b | 0.704 | 0.7 |
| ν_c | 13.11 | 15 |

Tabell 8.9: Simulert fra PCCDCC, estimert med C-DCC; Høye frihetsgrader, høy korrelasjon, høy a , lav b .

Resultatene blir her litt dårligere enn i forrige forsøk og det kan se ut som modellen gjør det bedre for lave frihetsgrader enn for høye. Jeg vil så sjekke om dette også gjelder når a -verdien er lav. Jeg gjør derfor en trekning med $a=0.01$, $b=0.9$ og $\nu_C = 15$. Resultatene vises i tabell 8.10 og tabell 8.11.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.009 | 0.01 |
| b | 0.892 | 0.9 |
| $\nu_{c_{12}}$ | 11.99 | 15 |
| $\nu_{c_{23}}$ | 15.77 | 15 |
| $\nu_{c_{34}}$ | 15.87 | 15 |
| $\nu_{c_{13 2}}$ | 14.01 | 16 |
| $\nu_{c_{24 3}}$ | 13.82 | 16 |
| $\nu_{c_{14 23}}$ | 23.31 | 17 |
| likelihood | 6081.57 | 6061.46 |

Tabell 8.10: Simulert fra C-DCC, estimert med PCCDCC; Høye frihetsgrader, høy korrelasjon, lav a , høy b .

Estimatene som kommer ut er igjen litt mer nøyaktige, men fortsatt dårligere enn for de lave frihetsgradene.

De foregående eksperimentene bekrefter at simulerings og estimeringsalgoritmene er riktige i og med at jeg, i de tilfellene der modellen ser ut til å være mest nøyaktig, får gode estimater når jeg simulerer og estimerer med ulike algoritmer for den samme copulaen. Videre vil jeg teste om stor variasjon av frihetsgradene kan være avgjørende for nøyaktigheten på estimatene.

| Parameter | Estimat | Forventet verdi |
|-----------|---------|-----------------|
| a | 0.012 | 0.01 |
| b | 0.912 | 0.9 |
| ν_c | 15.16 | 15 |

Tabell 8.11: Simulert fra PCCDCC, estimert med C-DCC; Høye frihetsgrader, høy korrelasjon, lav a , høy b .

Jeg må da bruke PCCDCC-modellen både til simulering og estimering.

Jeg går tilbake til $a = 0.2$ og $b = 0.7$ og simulerer med frihetsgrader 50, 30, 20, 12, 7, 3. Estimeringsresultatene vises i tabell 8.12.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.208 | 0.2 |
| b | 0.707 | 0.7 |
| $\nu_{c_{12}}$ | 52.42 | 50 |
| $\nu_{c_{23}}$ | 23.73 | 30 |
| $\nu_{c_{34}}$ | 25.45 | 20 |
| $\nu_{c_{13 2}}$ | 9.93 | 12 |
| $\nu_{c_{24 3}}$ | 5.32 | 7 |
| $\nu_{c_{14 23}}$ | 2.88 | 3 |
| likelihood | 7104.98 | 7118.98 |

Tabell 8.12: Simulert fra PCCDCC, estimert med PCCDCC; varierende frihetsgrader fra høy til lav, høy korrelasjon, høy a , lav b .

I neste eksperiment snus variasjonen i frihetsgrader motsatt vei så de laveste frihetsgradene kommer på første nivå. Resultatene vises i tabell 8.13.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.210 | 0.2 |
| b | 0.696 | 0.7 |
| $\nu_{c_{12}}$ | 2.87 | 3 |
| $\nu_{c_{23}}$ | 6.48 | 8 |
| $\nu_{c_{34}}$ | 9.8 | 12 |
| $\nu_{c_{13 2}}$ | 18.41 | 20 |
| $\nu_{c_{24 3}}$ | 20.033 | 30 |
| $\nu_{c_{14 23}}$ | 18.19 | 50 |
| likelihood | 6979.03 | 6985.87 |

Tabell 8.13: Simulert fra PCCDCC, estimert med PCCDCC; varierende frihetsgrader fra lav til høy, høy korrelasjon, høy a , lav b .

Det ser ut til at modellen håndterer variasjonene best når de høye frihetsgradene kommer på de første nivåene. Jeg kjører den siste testen en gang til for å se om estimatene blir bedre med andre DCC-parametere; $a = 0.01$ og $b = 0.9$. Dette gir estimatene i tabell 8.14.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.011 | 0.01 |
| b | 0.839 | 0.9 |
| $\nu_{c_{12}}$ | 2.63 | 3 |
| $\nu_{c_{23}}$ | 6.18 | 8 |
| $\nu_{c_{34}}$ | 14.18 | 12 |
| $\nu_{c_{13 2}}$ | 29.48 | 20 |
| $\nu_{c_{24 3}}$ | 29.60 | 30 |
| $\nu_{c_{14 23}}$ | 82.62 | 50 |
| likelihood | 5604.77 | 5605.85 |

Tabell 8.14: Simulert fra PCCDCC, estimert med PCCDCC; varierende frihetsgrader fra lav til høy, høy korrelasjon, lav a , høy b .

8.2.2 Korrelasjonsmatrise 2

$$Q_{1,1} = \begin{pmatrix} 1 & 0.05 & 0.2 & 0.15 \\ 0.05 & 1 & 0.35 & 0.4 \\ 0.2 & 0.35 & 1 & 0.1 \\ 0.15 & 0.4 & 0.1 & 1 \end{pmatrix}$$

Som nevnt tidligere kjører jeg de to første forsøkene på nytt med en korrelasjonsmatrise med lavere verdier. Estimeringsresultatene er vist i tabell 8.15, 8.16, 8.17 og 8.18. Ut fra disse ser det ikke ut til at estimatene blir verken dårligere eller bedre ved å endre verdien på korrelasjonene.

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.215 | 0.2 |
| b | 0.688 | 0.7 |
| $\nu_{c_{12}}$ | 3.28 | 3 |
| $\nu_{c_{23}}$ | 3.18 | 3 |
| $\nu_{c_{34}}$ | 3.67 | 3 |
| $\nu_{c_{13 2}}$ | 6.65 | 4 |
| $\nu_{c_{24 3}}$ | 3.93 | 4 |
| $\nu_{c_{14 23}}$ | 5.79 | 5 |
| likelihood | 4278.57 | 4308.73 |

Tabell 8.15: Simulering fra C-DCC, optimering med PCCDCC; Lave frihetsgrader, lav korrelasjon, høy a , lav b .

| Parameter | Estimat | Forventet verdi |
|-----------|---------|-----------------|
| a | 0.196 | 0.2 |
| b | 0.703 | 0.7 |
| ν_c | 2.95 | 3 |

Tabell 8.16: Simulering fra PCCDCC, optimering med C-DCC; Lave frihetsgrader, lav korrelasjon, høy a , lav b .

| Parameter | Estimat | Forventet verdi |
|-------------------|---------|-----------------|
| a | 0.208 | 0.2 |
| b | 0.679 | 0.7 |
| $\nu_{c_{12}}$ | 14.26 | 15 |
| $\nu_{c_{23}}$ | 21.01 | 15 |
| $\nu_{c_{34}}$ | 13.21 | 15 |
| $\nu_{c_{13 2}}$ | 13.17 | 16 |
| $\nu_{c_{24 3}}$ | 11.4 | 16 |
| $\nu_{c_{14 23}}$ | 14.00 | 17 |
| likelihood | 2716.88 | 2729.66 |

Tabell 8.17: Simulert fra C-DCC, estimert med PCCDCC; Høye frihetsgrader, lav korrelasjon, høy a , lav b .

| Parameter | Estimat | Forventet verdi |
|-----------|---------|-----------------|
| a | 0.215 | 0.2 |
| b | 0.686 | 0.7 |
| ν_c | 16.07 | 15 |

Tabell 8.18: Simulert fra PCCDCC, estimert med C-DCC; Høye frihetsgrader, lav korrelasjon, høy a , lav b .

8.3 Oppsummering

Når man oppsummerer de foregående testene ser det ut til at verdien på DCC-parameterene påvirker nøyaktigheten av estimatene på frihetsgradene når disse er høye eller har stor spredning (mest når verdiene i 3.nivå er størst). Verdien på korrelasjonene ser ikke ut til å ha noe å si for de resterende estimatene. Ser man derimot på log-likelihoodverdiene er disse svært nær log-likelihoodverdiene for de sanne parameterene selv når estimatene er unøyaktige.

Å estimere frihetsgradene helt eksakt i alle tilfeller ser altså ut til å være vanskelig, men

hvis man ser på log-likelihoodverdiene er disse i alle tilfeller veldig like for de estimerte og de sanne verdiene. Dette antyder at det kanskje ikke er så farlig at estimatene er litt unøyaktige.

Kapittel 9

Modellene brukt i praksis

I dette kapitlet tilpasses de ulike modellene til et 4-dimensjonalt datasett fra den virkelige verden. Datasettet består av indeksverdier til følgende aksjeindekser fra Oslo Børs:

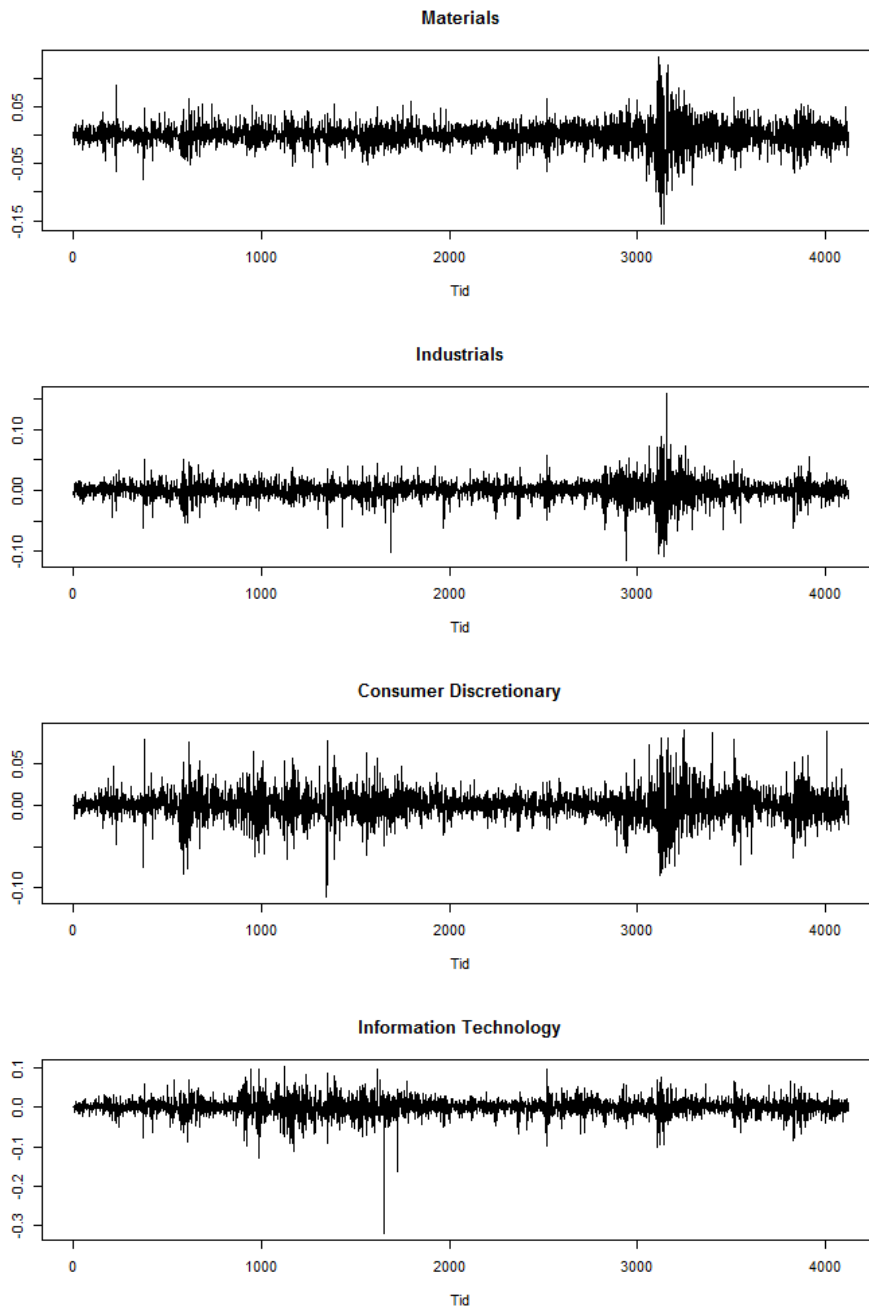
1. OSE15GI: Materials.
2. OSE20GI: Industrials.
3. OSE25GI: Consumer Discretionary.
4. OSE45GI: Information Technology.

Dataene er daglige og verdiene er fra mai 1996 til september 2012.

9.1 Dataanalyse

Jeg starter med å gjøre en analyse av det aktuelle datasettet for å underbygge de valgene jeg gjør i forhold til modellene jeg bruker. De fire tidsrekkene er vist i Figur 9.1. Som forventet, siden datasettet består av finansielle tidsrekker, ser vi at alle rekkene har volatilitetsklynger; perioder med høy volatilitet og perioder med lav volatilitet. Dette indikerer at en GARCH modell kan brukes til å tilpasse dataene. Det kan også se ut som periodene med høy volatilitet kommer noenlunde samtidig for alle rekkene, og en modell som er i stand til å modellere avhengighet vil være nødvendig.

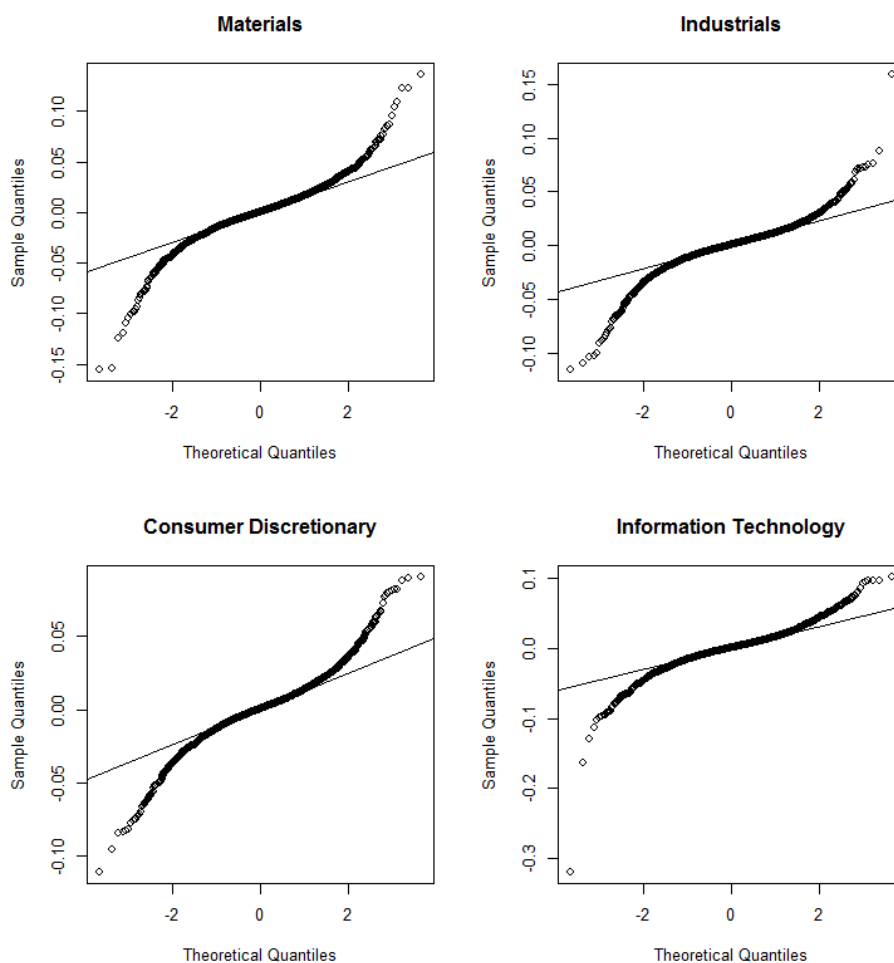
For å sjekke antagelsen om mer tunghalede fordelinger enn normalfordelingen gjelder for dette datasettet, lager jeg et QQ-plott for hver av tidsrekkene. Disse er vist i figur 9.2. Dersom



Figur 9.1: Logaritmiske avkastninger for de 4 indeksene i perioden 03.05.1996 til 28.09.2012.

dataene er normalfordelte skulle punktene på QQ-plottet ha ligget langs den rette linjen hele veien. Dette er ikke tilfelle for noen av indeksene. At indeksverdiene avviker fra linjen i begge endene bekrefter at de følger mer tunghalede fordelinger, som for eksempel Student's t -fordelingen.

Figur 9.3 viser autokorrelasjonsfunksjonen til \mathbf{a}_t , mens figur 9.4 viser autokorrelasjons-



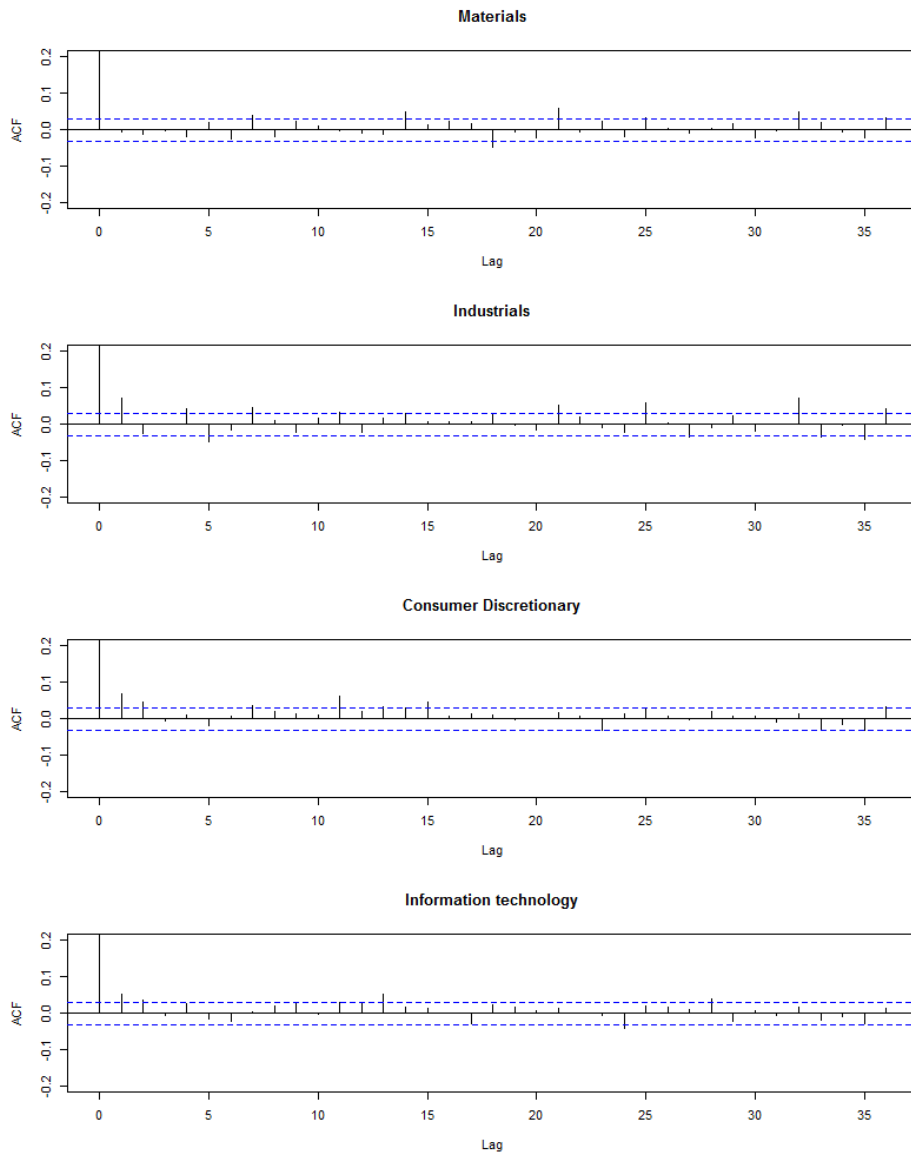
Figur 9.2: QQ-plott for de fire tidsrekkene.

funksjonen til \mathbf{a}_t^2 . Autokorrelasjon beskriver samvariasjon mellom observasjoner på to ulike tidspunkter. For at \mathbf{a}_t skal være uavhengige i tid, må \mathbf{a}_t^2 være ukorrelert.

Dersom \mathbf{a}_t er ukorrelert, vil ferre enn 5 % av verdiene komme på utsiden av de prikkede linjene på acf-plottet. For 30 perioder skal altså bare 1.5 verdier falle utenfor. Fra plottene ser vi at mer enn 2 verdier er utenfor for alle indeksene bortsett fra IT, men det finnes ikke noe bestemt mønster, og overskridelsene er små, så jeg konkluderer med at \mathbf{a}_t er tilnærmet ukorrelert for alle tidsrekkene.

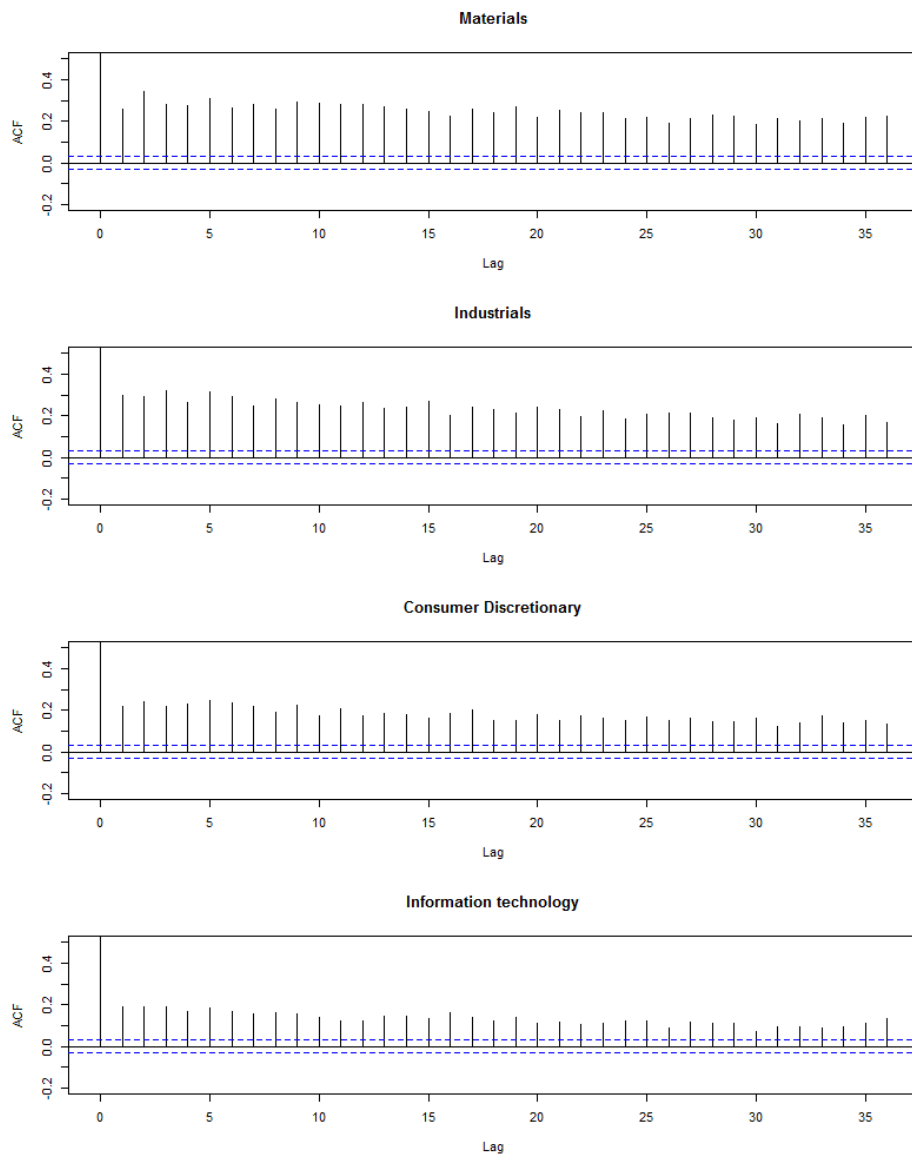
Det samme gjelder derimot ikke for \mathbf{a}_t^2 . For alle tidsrekkene overskrider mer enn 5 % av verdiene den prikkede linjen. Det ser i tillegg ut til å være et mønster med høyere utslag i starten enn på slutten. \mathbf{a}_t er altså ukorrelert og avhengig, noe som betyr at GARCH-modellen er et naturlig valg for modellering av \mathbf{a}_t for alle indeksene.

Figur 9.5 og 9.6 er plott av krysskorrelasjonen mellom to og to tidsrekker, mens figur

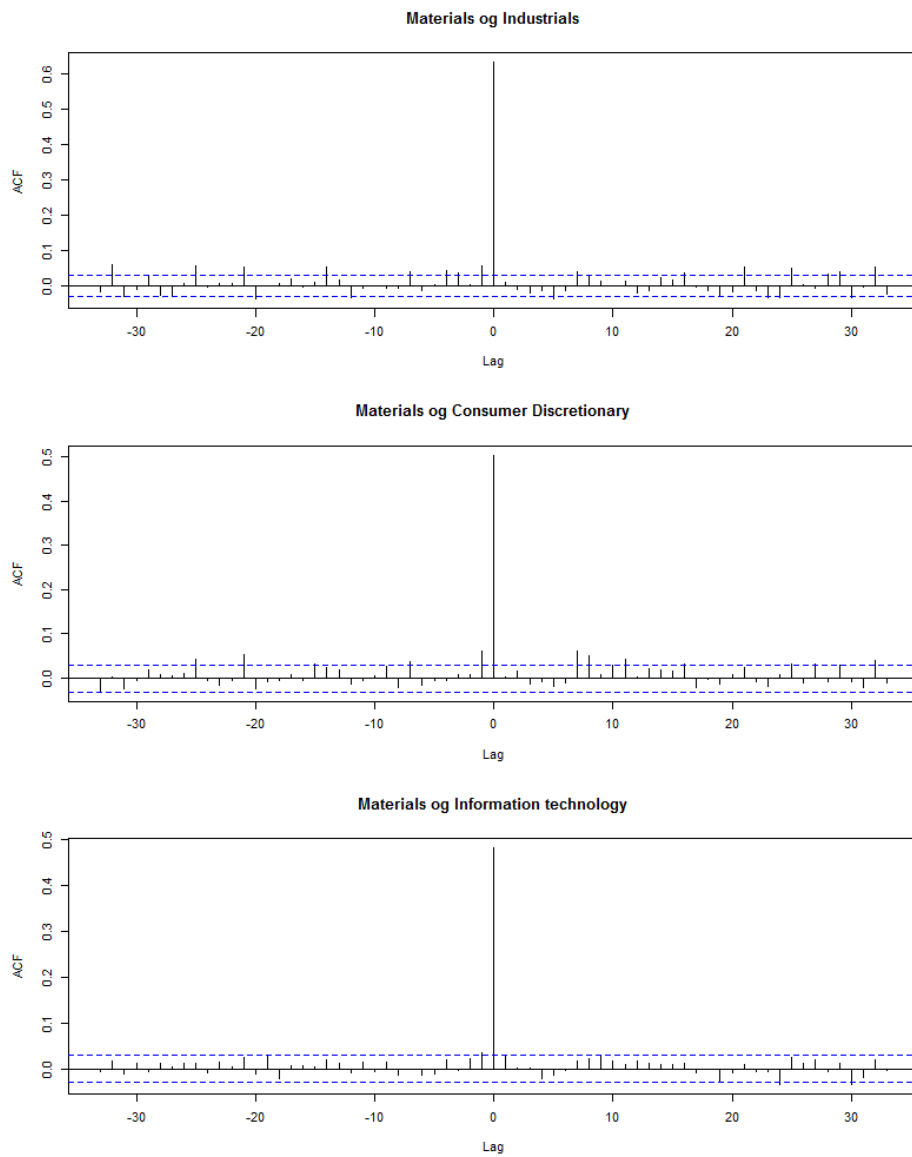


Figur 9.3: Autokorrelasjonsplott for de samme dataene.

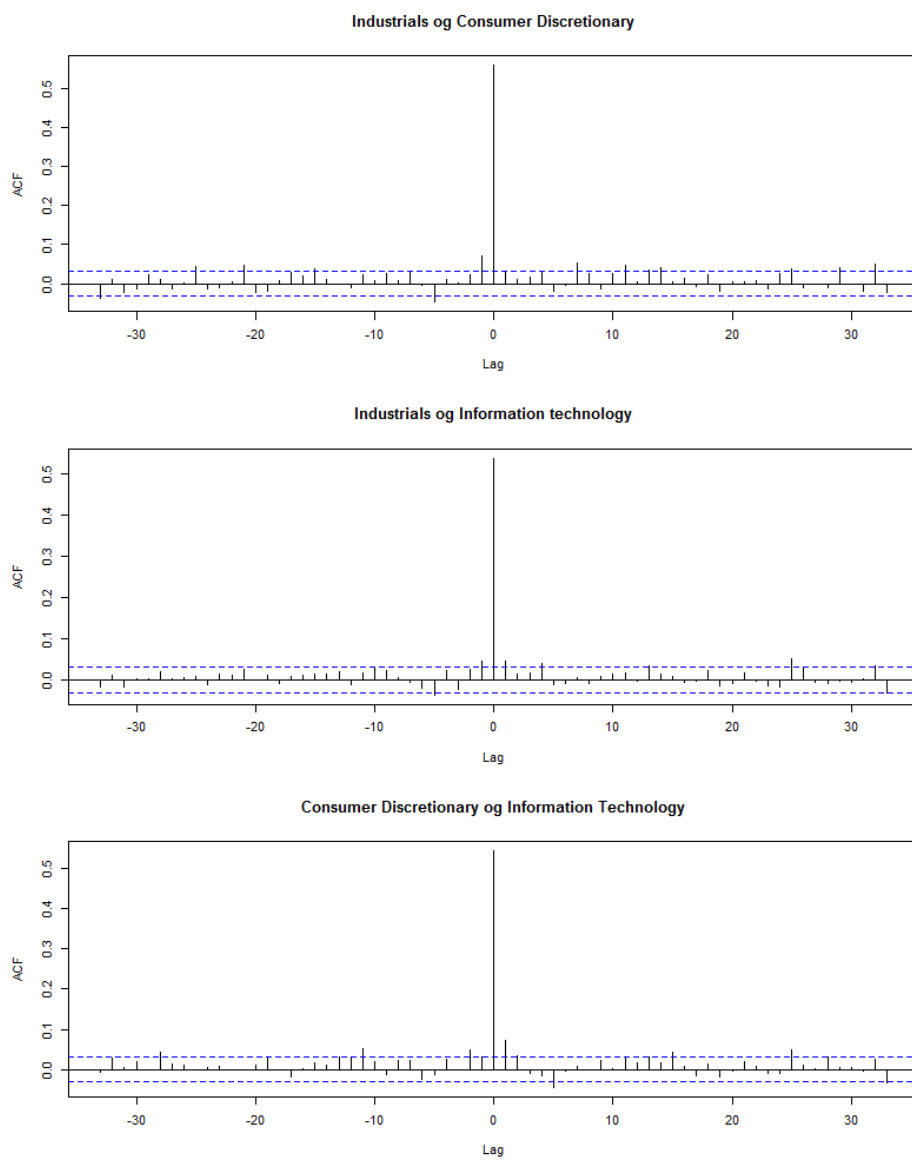
9.7 og 9.8 viser krysskorrelasjonsplott for absoluttverdien til de samme dataene. Ut fra figur 9.5 og 9.6 konkluderer jeg med at vi har tilnærmet ingen krysskorrelasjon mellom dataene, men på de to neste figurene er situasjonen annerledes. Plottene viser tydelig krysskorrelasjon mellom alle parene. Dette indikerer at en multivariat modell vil være nødvendig for å ikke miste aspektet med avhengighet på tvers av rekkene.



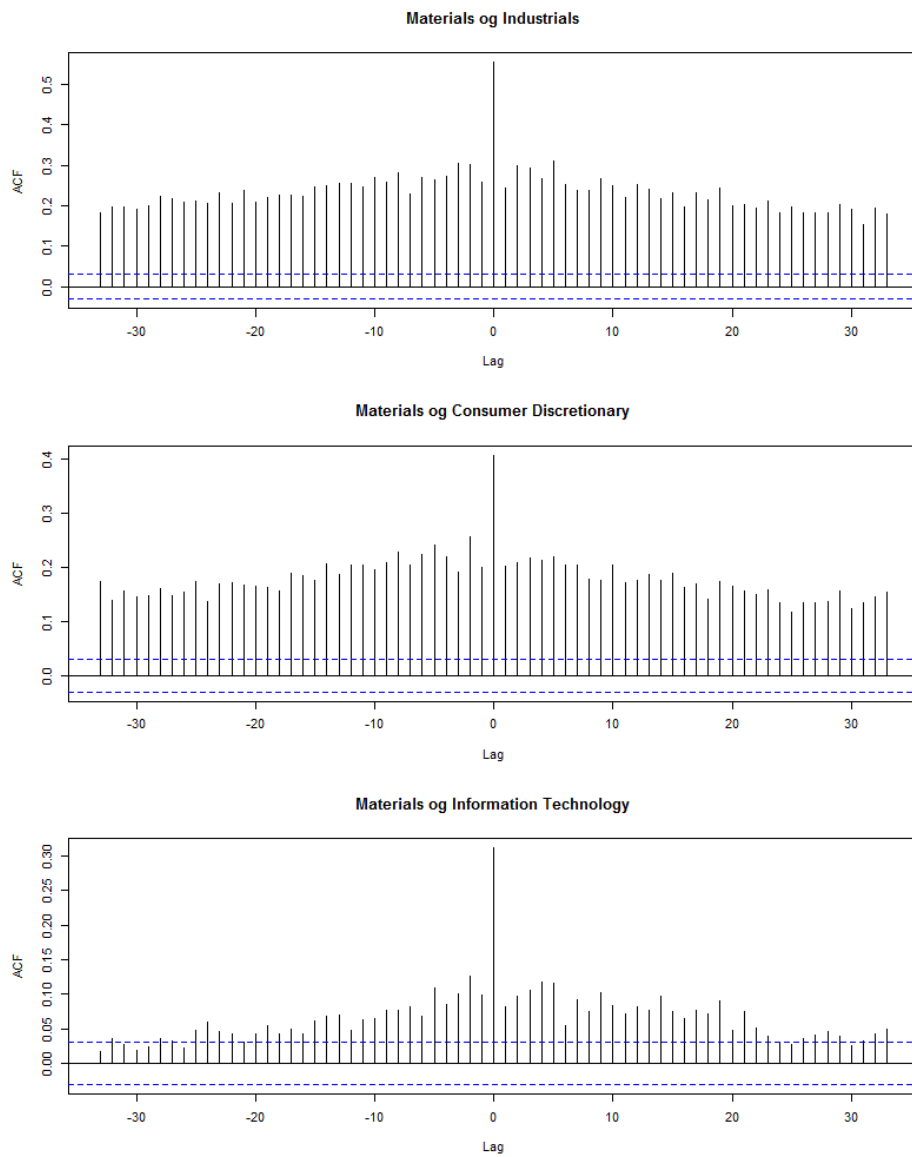
Figur 9.4: ACF-plott av log-avkastningsdataene kvadrert.



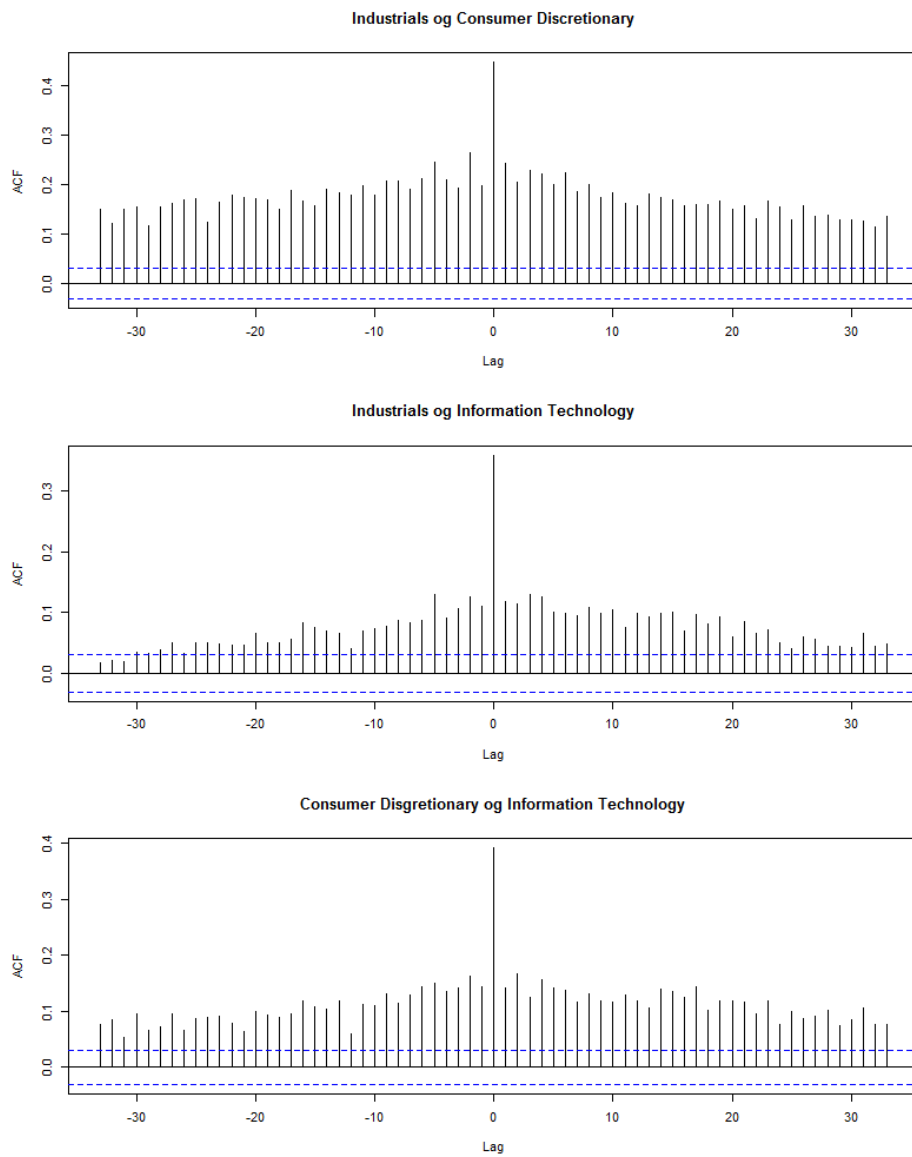
Figur 9.5: Krysskorrelasjonsplott mellom 3 av parene.



Figur 9.6: Krysskorrelasjonsplott mellom de resterende parene.



Figur 9.7: Krysskorrelasjonsplott for absoluttverdien til de tre første parene.



Figur 9.8: Krysskorrelasjonsplott for absoluttverdien til de samme parene.

9.2 Valg av par-copula-oppsett

Som et første steg for å kunne tilpasse dataene til en PCC-DCC-GARCH-modell må jeg velge hvilket par-copula-oppsett som skal tilpasses. Jeg starter med å transformere dataene over på copula-skalaen. Dette gjør jeg ved å først tilpasse univariate GARCH-modeller til de univariate rekkene og bruker de estimerte marginalmodellene til å gjøre transformasjonen.

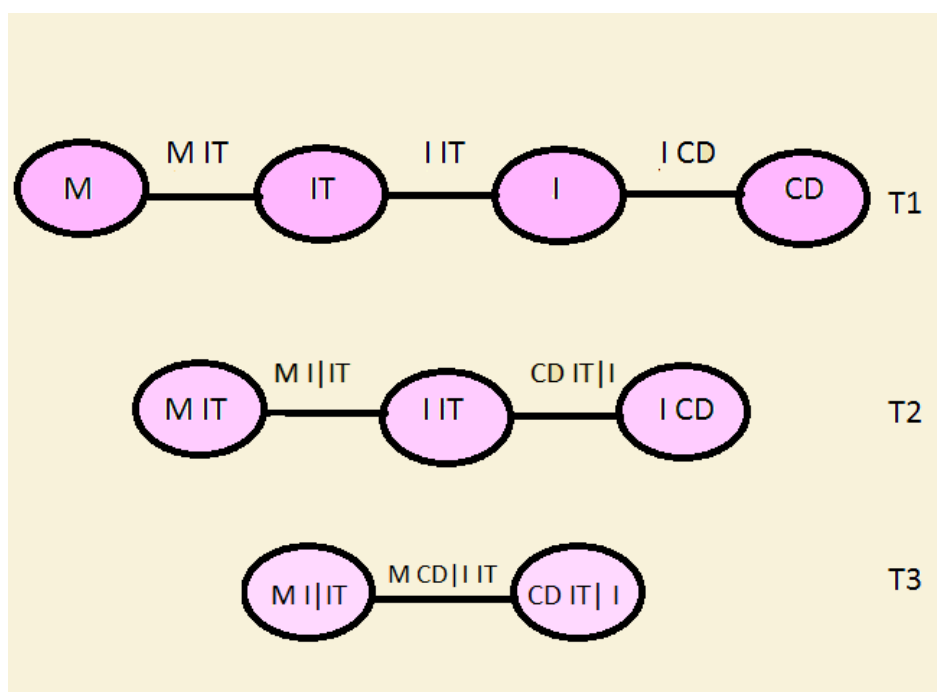
For å velge riktig vine-struktur i par-copula-konstruksjonen tilpasser jeg hvert par av risikofaktorer til en bivariat Student's t copula. Deretter beregnes haleavhengigheten for hvert av parene med ligning (3.23). Dette for å undersøke mellom hvilke variable avhengigheten er størst og om det eventuelt er en av variablene som har spesielt høy avhengighet til alle de andre slik at det vil være naturlig med en C-vine-struktur.

I denne estimeringen holdes korrelasjonen konstant. De estimerte haleavhengighetene er vist i tabell 9.1.

| Mellom | M | I | CD |
|--------|-------|-------|-------|
| IT | 0.069 | 0.156 | 0.095 |
| M | | 0.093 | 0.049 |
| I | | | 0.109 |

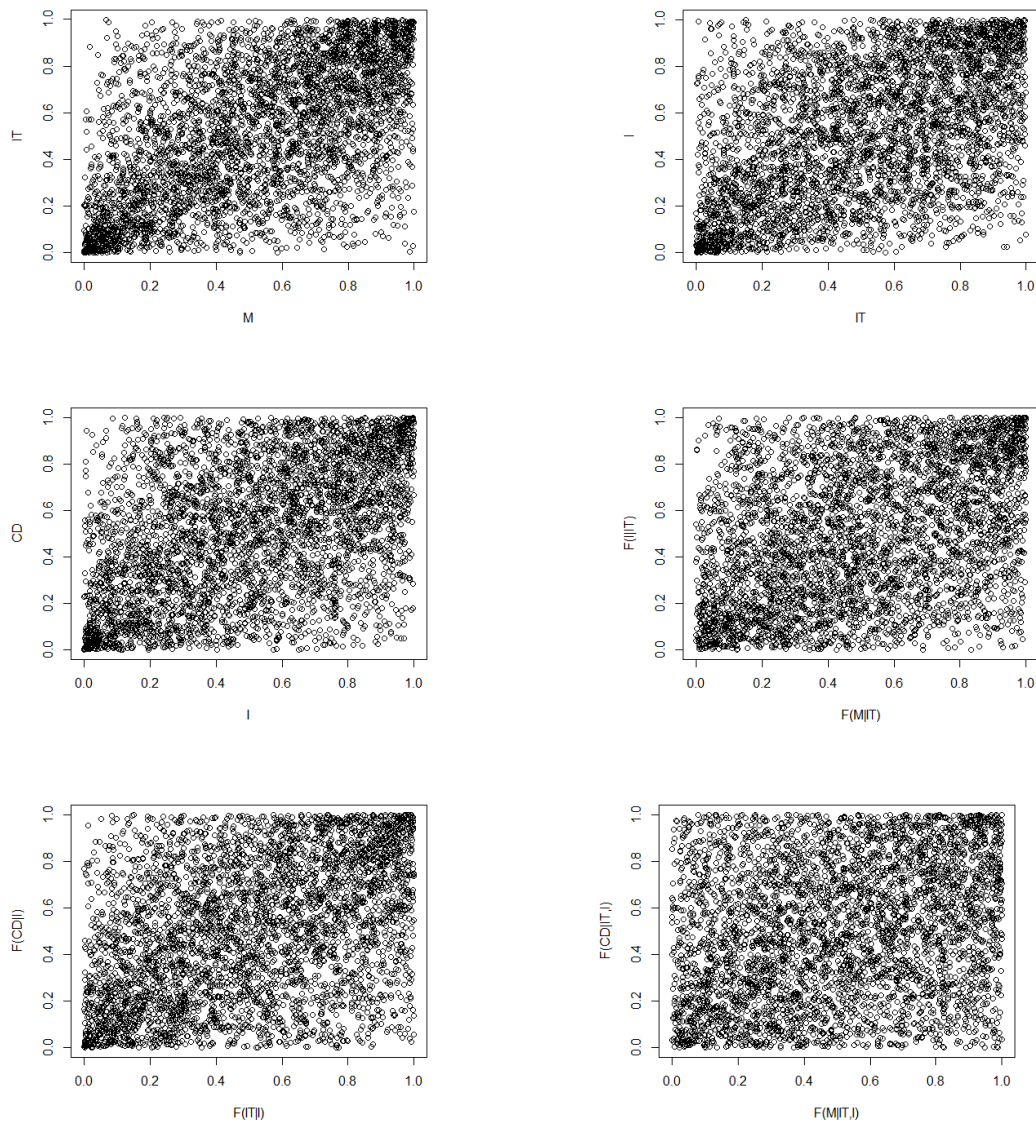
Tabell 9.1: Estimerte verdier for haleavhengigheten i bivariate Student's t-copulaer mellom par av variable.

De estimerte haleavhengighetskoeffisientene er relativt forskjellige og rettferdiggjør bruk av en par-copula-konstruksjon i stedet for en multivariat copula. Det ser derimot ikke ut som en av variablene har spesielt sterk avhengighet til alle de andre, så jeg velger å tilpasse dataene til en D-vine par-copula-konstruksjon. Jeg ordner dataene slik at de tre copulaene som skal tilpasses på nivå 1 i PCC-DCC-modellen korresponderer til de parene med sterkest haleavhengighet. Den valgte D-vine-strukturen er vist i figur 9.9.



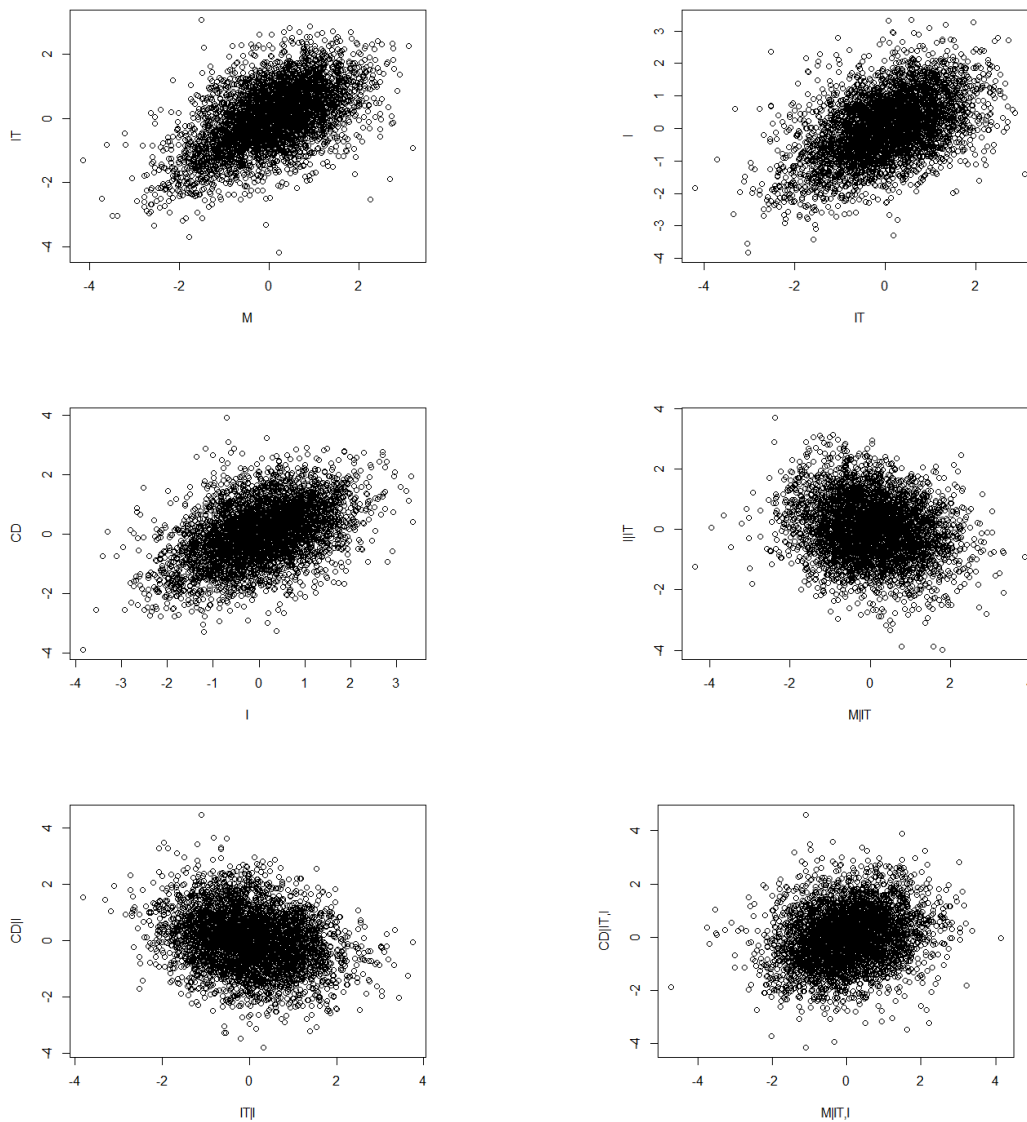
Figur 9.9: D-vine-struktur for det aktuelle datasettet.

Neste steg er å gjøre en vurdering av hvilke copulaer som er naturlig å bruke for hvilke par i oppsettet. Dette kan gjøres med en visuell vurdering av scatterplott av de to variablene som inngår i hver par-copula. For å få de betingede dataene i nivå 2 og 3 bruker jeg ligning (5.7). Scatterplottene er vist i figur 8.10.



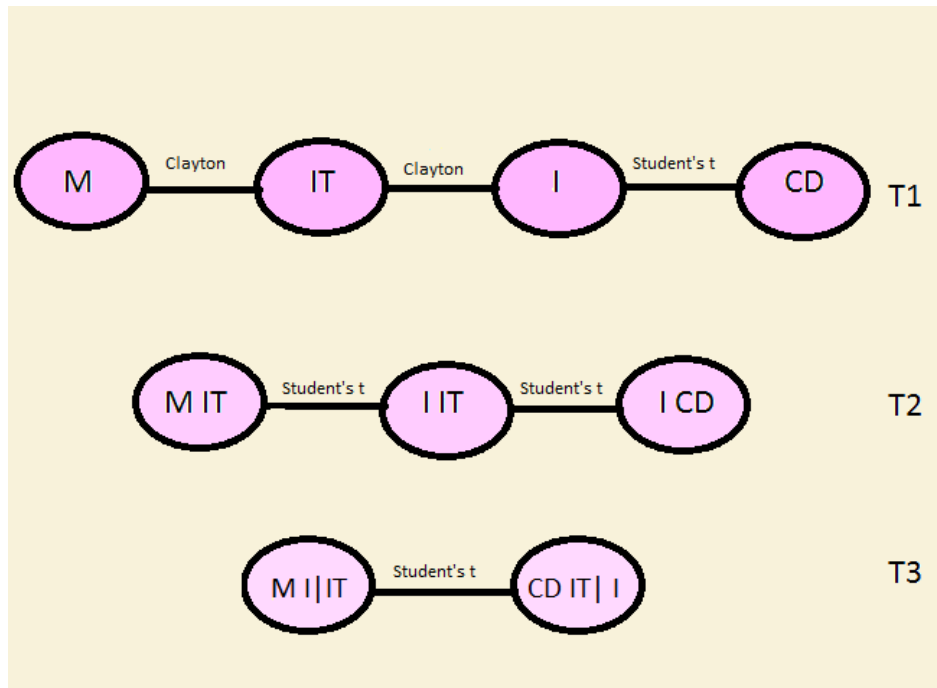
Figur 9.10: Scatter-plot av alle parene.

I spesielt de 4 øverste plottene ser det ut til å være sterkere avhengighet i halene enn ellers, men det er vanskelig å se ut fra disse plottene om denne er sterkere i nedre enn i øvre hale eller om den er symmetrisk. Et triks for å lettere se slike assymetrier kan da være å transformere dataene om til standard normalfordelte data. Dette er gjort i figur 8.11.



Figur 9.11: De samme dataene transformert til standard normalfordelte variable.

I denne figuren kan det se ut som om haleavhengigheten i de to øverste plottene er litt sterkere i den nedre halen enn den øvre, siden “ellipsen” spisser seg mot det nedre hjørnet. Jeg konkluderer derfor med at D-vine-oppsettet i figur 9.12 er passende for dette datasettet.



Figur 9.12: D-vine-oppsett med spesifisering av hvilke bivariate copulaer som antas å passe best for hvilke par.

9.3 Parameterestimering

Til tross for at jeg i forrige avsnitt konkluderte med at en D-vine-struktur som vist i figur 8.12 vil være den som beskriver dette datasettet best, velger jeg likevel å tilpasse datasettet til alle de tre tidligere nevnte modellene; 4-dimensjonal Copula-DCC-GARCH, PCC-DCC-GARCH med bivariat Student's t copula for alle par og PCC-DCC-GARCH med både bivariante Student t copulaer og bivariante Clayton copulaer. Jeg vil senere se på hvordan disse modellene gjør det i forhold til hverandre på dette datasettet.

Før jeg gjør estimeringene, sorteres tidsrekkene slik at jeg får de i riktig rekkefølge i forhold til D-vine-oppsettet i figur 9.9. Verdiene for parameterene i de univariate rekkene vil da være de samme for alle modellene siden denne estimeringen gjøres uavhengig av hvilken copula man velger å tilpasse. Optimering av likelihooden i ligning (3.5) gir følgende estimater:

| | $\widehat{\alpha}_0$ | $\widehat{\alpha}_1$ | $\widehat{\beta}$ | $\widehat{\nu}$ |
|------------------------|----------------------|----------------------|-------------------|-----------------|
| Materials | $4.98e - 06$ | 0.1161 | 0.8635 | 6.47 |
| Industrials | $4.83e - 06$ | 0.0922 | 0.9029 | 5.00 |
| Costumer Discretionary | $4.16e - 06$ | 0.0753 | 0.9141 | 8.47 |
| Information Technology | $2.04e - 06$ | 0.0789 | 0.9176 | 6.24 |

Tabell 9.2: Estimerte parametere i de univariate GARCH(1,1)-rekkene.

Når jeg har estimert disse verdiene, kan jeg lett finne estimater for ϵ ut fra ligningene (2.4) og (2.10). Videre finner jeg \bar{Q} , som blir brukt som startverdi for Q -matrisen, med ligning (2.11). Den estimerte R_1 -matrisen blir da som følger:

$$R_1 = Q_1^{*-1} Q_1 Q_1^{*-1} = \begin{pmatrix} 1 & 0.54 & 0.55 & 0.49 \\ 0.54 & 1 & 0.48 & 0.52 \\ 0.55 & 0.48 & 1 & 0.45 \\ 0.49 & 0.52 & 0.45 & 1 \end{pmatrix}.$$

9.3.1 4-dimensjonal Student t-copula

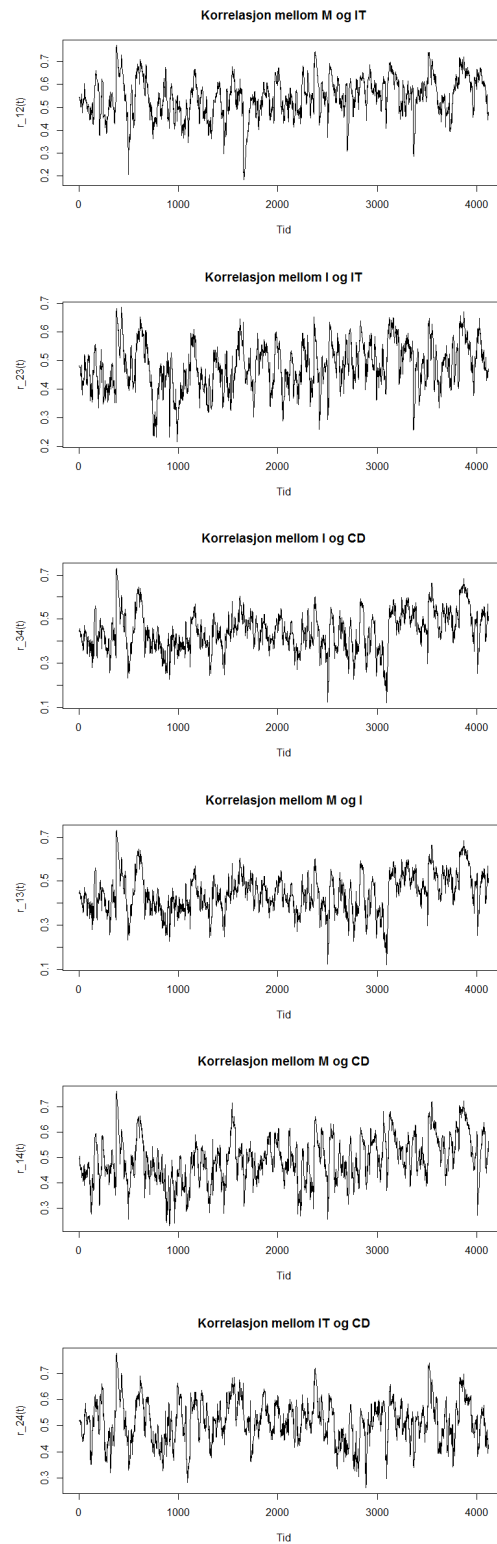
For å finne utviklingen i korrelasjonen (DCC-parameterene a og b) og frihetsgradene i den 4-dimensjonale Student's t-copulaen, optimerer jeg likelihooden i ligning (4.6). Optimeringen gir følgende verdier:

| Par | \hat{a} | \hat{b} | $\hat{\nu}$ |
|-----|-----------|-----------|-------------|
| Est | 0.02534 | 0.94621 | 14.08 |

Tabell 9.3: Estimerte DCC og copula-parametere.

Siden de univariate parameterene er identiske for alle modellene vil også startverdiene for korrelasjonen være det, og så snart man har estimert DCC-parameterene er man i stand til å danne et bilde av utviklingen i korrelasjonen. Utviklingen er for denne modellen vist i figur 9.13.

Ut fra plottene ser vi at det er ganske store variasjoner i korrelasjonen. I øverste plott ser vi blant annet at korrelasjonen går fra å være ca 0.8 og helt ned til rundt 0.2 på bare noen få dager. Dette underbygger antagelsen om DCC-dynamikk i dataene.



Figur 9.13: Utviklingen av korrelasjonen mellom de ulike parene over tid.

9.3.2 Par-copula-konstruksjon med Student t copula for alle par av variable

Estimeringen av parameterene i de univariate rekkene blir, som nevnt tidligere, helt identisk for alle modellene, og jeg får verdiene vist i tabell 9.2. Korrelasjonsmatrisen ved starttidspunktet vil også bli helt lik.

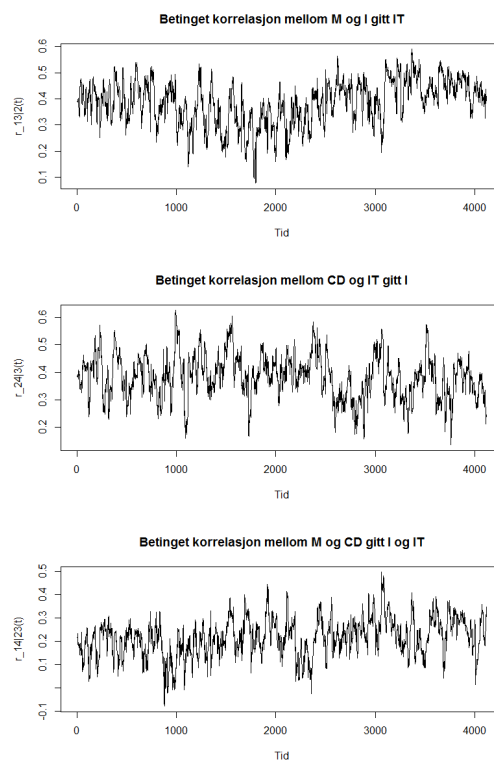
Når jeg videre skal finne parameterene i de bivariante copulaene i Par-Copula-konstruksjonen bruker jeg, som beskrevet i avsnitt 5.4, først en sekvensiell algoritme for å finne gode startverdier til den fulle likelihooden. Jeg lar korrelasjonsmatrisen være tidsvarierende i nivå 1 slik at jeg får estimater for a og b . I de andre nivåene lar jeg korrelasjonen være konstant siden dette sannsynligvis gir gode nok startverdier. Til slutt brukes de estimerte startverdiene i optimeringen av den fulle likelihooden. Tabell 9.4 viser de estimerte parameterene, både for den sekvensielle optimeringen og for den fulle.

| Par | Sekv | Full |
|-------------------|---------|---------|
| a | 0.03512 | 0.02502 |
| b | 0.9301 | 0.94608 |
| $\nu_{c_{12}}$ | 8.537 | 9.193 |
| $\nu_{c_{23}}$ | 11.255 | 13.276 |
| $\nu_{c_{34}}$ | 13.091 | 14.070 |
| $\nu_{c_{13 2}}$ | 21.665 | 21.715 |
| $\nu_{c_{24 3}}$ | 17.816 | 16.467 |
| $\nu_{c_{14 23}}$ | 24.100 | 22.430 |

Tabell 9.4: Tabell over estimerte parametere i modell 2.

Legg merke til at frihetsgradene varierer fra 9 og helt til 22. Stor variasjon i frihetsgradene er et tegn på at en PCC beskriver dataene bedre enn en multivariat Student's t copula. De estimerte DCC-parameterene er også slik at estimatene antas å være gode for dette datasettet.

Siden DCC-estimatene er (nesten) like som for C-DCC-modellen, og derav vil gi en lignende utvikling av korrelasjonen, velger jeg å kun plote utviklingen av de betingede korrelasjonene i nivå 2 og 3. Dette er gjort i figur 9.14. Også her ser vi ganske store variasjoner.



Figur 9.14: Utviklingen av korrelasjonen mellom de ulike parene i nivå 2 og 3 over tid.

9.3.3 Par-copula-konstruksjon med både Clayton- og Student t- copulaer

Fremgangsmåten for å finne parameterene i de bivariate copulaene i den andre Par-Copula-konstruksjonen er den samme som for modellen med Student t-copulaer for alle par. Jeg bruker først en sekvensiell algoritme for å finne gode startverdier til den fulle likelihooden.

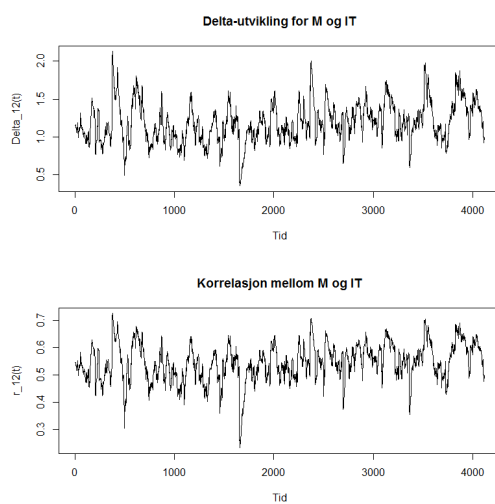
Forskjellen i denne estimeringen er at to av likelihoodene i nivå 1 er erstattet med Clayton-likelihood i stedet for Student's t, og tilsvarende for h-funksjoner. Tabell 9.5 viser de estimerte parameterene, både for den sekvensielle optimeringen og for den fulle.

| par | sekv | full |
|-------------------|---------|---------|
| a | 0.02152 | 0.01773 |
| b | 0.94756 | 0.95441 |
| $\nu_{c_{34}}$ | 13.023 | 13.835 |
| $\nu_{c_{13 2}}$ | 20.965 | 23.748 |
| $\nu_{c_{24 3}}$ | 32.339 | 29.03 |
| $\nu_{c_{14 23}}$ | 24.936 | 26.679 |

Tabell 9.5: Tabell over estimerte parametere i modell 3.

Som tabellen viser mister jeg to parametere i denne modellen i forhold til den forige. Dette skyldes at Clayton-copulaen kun har en parameter (i motsetning til Student's t-copulaen som har to), og denne er definert ved startverdien for korrelasjonen og utviklingsparameterene a og b . Startverdien for korrelasjonene er i dette tilfellet 0.54 og 0.55, som tilsvarer en startverdi for copulaparameterene på 1.16 og 0.94. Legg merke til at korrelasjonen er godt over 0, som er grensa for gyldige δ -verdier. Den estimerte a-verdien er også så lav at det ikke er fare for negative parametere for noen av tidspunktene.

Utviklingen av parameterene i de to Clayton-copulaene er vist i figur 9.15.



Figur 9.15: Utviklingen av Clayton-parameterene i nivå 1 over tid.

9.4 Modellvalidering

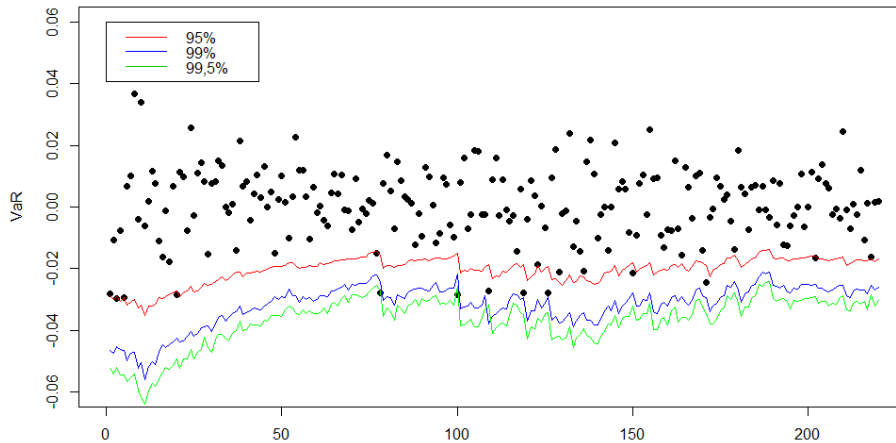
I dette avsnittet gjøres tester av modellene for å studere hvor godt de beskriver det aktuelle datasettet. Jeg starter med å sjekke hvor godt modellene estimerer Value-at-Risk. Deretter brukes Akaikes informasjonskriterie for å sammenligne modellene og se hvilken som er den beste tilpassingen.

9.4.1 Backtesting Value at Risk; Kuipец test og Christoffersen's Markov test

Ved estimering av Value at Risk antas likevekter for alle aktiva i porteføljen, og $p_t = \omega^T \mathbf{a}_t = \frac{1}{4}(a_{1,t} + a_{2,t} + a_{3,t} + a_{4,t})$. For hver av modellene estimeres Value at Risk for de 220 siste datapunktene, 12.desember 2011 til 28.september 2012, basert på de 1000 foregående dagene. Verdiene sammenlignes så med de observerte porteføljevkastningene.

Siden forskjellene mellom modellene i denne oppgaven vil være mest fremtredende i halene, velger jeg å finne $VaR_{99,5}$ i tillegg til VaR_{99} og VaR_{95} . Dette gjøres med følgende algoritme for alle de tre modellene[5]:

1. For hver variabel, $j = 1, \dots, 4$, beregn prediksjonen av $h_{t,j}$ en dag frem, gitt de 1000 foregående observasjonene.
2. Beregn prediksjonen av \mathbf{R}_t en dag frem.
3. For hver simulering $n = 1, \dots, 10000$:
 - Generer en trekning, a_1, \dots, a_4 , fra modellen.
 - Beregn porteføljevkastningen $p_t = \frac{1}{4}(a_{1,t} + a_{2,t} + a_{3,t} + a_{4,t})$
3. For signifikansnivå $q \in \{0.005, 0.01, 0.05\}$
 - Beregn 1-day VaR_t^q som q-kvantilen i fordelingen til p_t .
 - Hvis VaR_t^q er større enn den observerte verdien til har vi en overskridelse.

Modell 1: Copula-DCC med multivariat Student's t copula

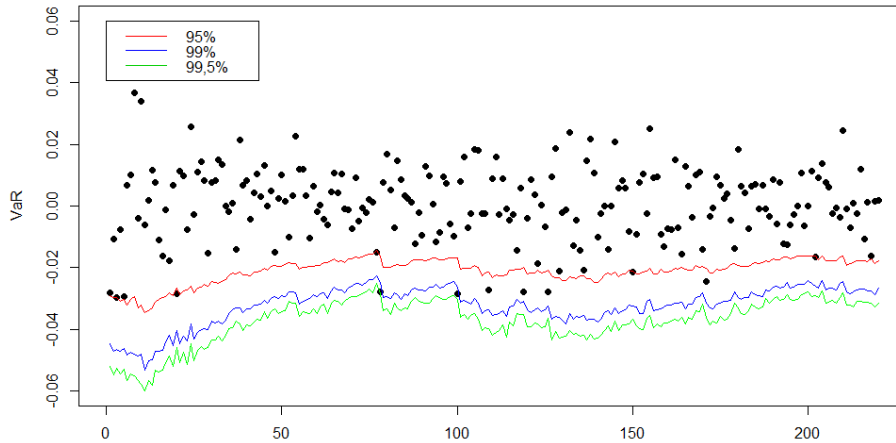
Figur 9.16: Estimerte Value-at-Risk verdier for modell 1

Figur 9.16 viser porteføljeavkastningen for de 220 siste dagene plottet mot Value-at-Risk verdier estimert med modell 1. Ved bare å se på figuren ser VaR-kurvene ut til å følge ytterpunktene til dataene bra. Der dataene viser høy volatilitet estimeres lave Value-at-Risk-verdier.

| Signifikansnivå | 0.005 | 0.1 | 0.5 |
|-----------------|-------|------|-----|
| Observert | 2 | 2 | 11 |
| Forventet | 1.1 | 2.2 | 11 |
| P-verdi | 0.44 | 0.89 | 1 |

Tabell 9.6: Kuiupec test for Modell 1.

I tabell 9.6 vises forventede overskridelser, observerte overskridelser og p-verdi i Kuiupec testen for de ulike signifikansnivåene. Det er høye p-verdier for alle de tre nivåene, og nullhypotesen i Kuiupec testen forkastes ikke i noen av tilfellene. Modell 1 gir gode estimater for Value-at-Risk.

Modell 2: PCC-DCC med bivariat Student's t copula for alle par.

Figur 9.17: Estimerte Value-at-Risk verdier for modell 2

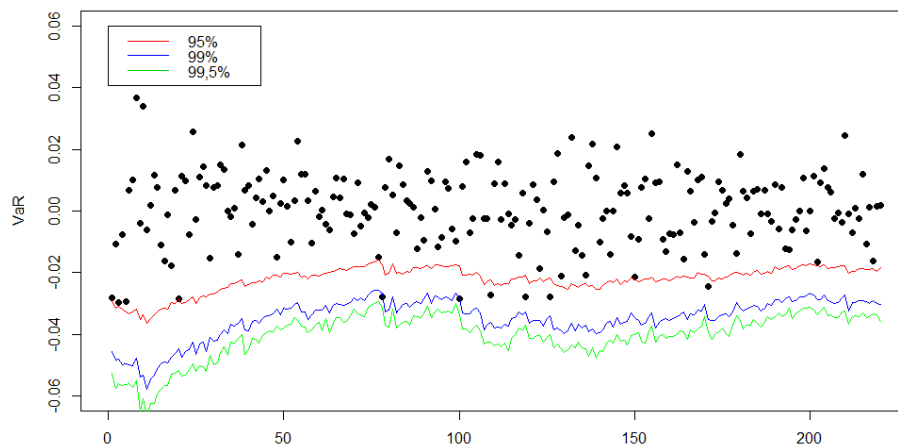
Som for modell 1 i figur 9.16, viser figur 9.17 porteføljeavkastningene for de 220 siste dagene plottet mot Value-at-Risk verdier estimert med modell 2. Det ser ut fra denne figuren ut som også modell 2 er god til å estimere Value at Risk.

Dette bekreftes i tabell 9.7. Antall overskridelser er veldig nær det forventede og p-verdiene er langt unna grensen for forkastning uansett hvilket signifikansnivå som velges for Kuipetesten. Sammenligner man med modell 1 har vi litt høyere p-verdi for $VaR_{99,5}$, men man må være forsiktig med å legge for mye i det i det ut fra så få observerte overskridelser.

| Signifikansnivå | 0.005 | 0.1 | 0.5 |
|-----------------|-------|------|------|
| Observert | 1 | 2 | 10 |
| Forventet | 1.1 | 2.2 | 11 |
| P-verdi | 0.92 | 0.89 | 0.75 |

Tabell 9.7: Kuipetest for Modell 2.

Modell 3: PCC-DCC med både bivariate Clayton- og Student's t copulaer i par-copula-konstruksjonen.



Figur 9.18: Estimerte Value-at-Risk verdier for modell 3

På samme måte som for de to foregående modellene har jeg plottet porteføljevkastningene for de 220 siste dagene mot estimerte Value at Risk-verdier også for modell 3. Dette er vist i figur 9.18.

| Signifikansnivå | 0.005 | 0.1 | 0.5 |
|-----------------|-------|------|------|
| Observert | 0 | 2 | 7 |
| Forventet | 1.1 | 2.2 | 11 |
| P-verdi | 0.13 | 0.89 | 0.19 |

Tabell 9.8: Kuippec test for modell 3.

Kuippec-testen forkaster heller ikke i denne modellen nullhypotesen, men hvis man sammenligner med de to foregående modellene er p-verdiene en del lavere. Selv om Modell 3 ikke kan forkastes ser det altså, basert på p-verdier i Kuippec-testen, ut til at Modell 1 og Modell 2 er

en litt bedre tilpassing til dette datasettet enn Modell 3.

Dette kan virke overraskende med tanke på at copula-strukturen i Modell 3 var den antatt beste ut fra de empiriske studiene tidligere i dette kapitlet. En forklaring kan være at det bare ut fra scatterplott er vanskelig å velge den beste copulaen. I figur 9.10 og 9.11 er det relativt enkelt å se at det er haleavhengighet mellom parene, men det er ikke like lett å se om det er assymetri i avhengighetsstrukturen eller ikke.

Christoffersen's Markov test

Kuipiec-testen sikrer den første egenskapen som må være oppfylt for Value at Risk-estimer; at antall observasjoner er tilfredsstillende nær det forventede. Den andre egenskapen, at overskridelsene er tilfeldig fordelt over tid, testes med Christoffersen's Markov test.

Siden det for VaR_{99} og $VaR_{99,5}$ bare er 1 og 2 overskridelser blir det meningsløst å sjekke om disse overskridelsene er tilfeldig fordelt. Christoffersen's Markov test utføres derfor bare på overskridelsene av VaR_{95} . Signifikansnivåene for de tre modellene er vist i tabell 9.9. Nullhypotesen forkastes ikke i noen av tilfellene.

| Modell | Modell 1 | Modell 2 | Modell 3 |
|---------|----------|----------|----------|
| P-verdi | 0.51 | 0.39 | 0.13 |

Tabell 9.9: Christoffersen's Markov test for de tre modellene.

9.4.2 AIC

I tillegg til Value at Risk, bruker jeg også Akaike's informasjonskriterer for å teste modellene opp mot hverandre. Når jeg skal finne AIC-verdier, velger jeg å "fjerne" marginalene og bare se på hvilken av modellene som modellerer avhengighetsstrukturen best. Jeg bruker ligning (7.1), hvor jeg setter inn de estimerte parameterene i copulalikelihooden for den gitte modellen.

AIC-verdiene for de ulike modellene er vist i tabell 9.10. Som tidligere refererer modell 1 til Copula-DCC modellen med multivariat Student's t copula, modell 2 er PCC-DCC-modellen med Student's t for alle par og modell 3 er PCC-DCC-modellen med både bivariate Clayton-copulaer og bivariate Student's t copulaer i par-copula-konstruksjonen.

| Modell | $\log(L_i(\hat{\theta}))$ | AIC_i |
|----------|---------------------------|----------|
| Modell 1 | 2794.84 | -5583.68 |
| Modell 2 | 2808.63 | -5601.26 |
| Modell 3 | 2629.23 | -5246.47 |

Tabell 9.10: AIC-verdier for de ulike DCC-modellene.

AIC-verdiene for de aktuelle modellene er relativt like. Spesielt er de to første modellene vanskelig å skille. Det ser likevel ut til at PCC-DCC-GARC modellen med Student's t copula for alle par er litt bedre enn de to andre modellene, mens Copula-DCC-GARCH modellen gjør det bedre enn PCC- modellen med Clayton for to av parene. Dette samsvarer med resultatene fra Value at Risk-eksperimentene.

Ut fra AIC-verdier og backtesting av Value at Risk konkluderer jeg altså med at, blant de modellene som nevnes i denne oppgaven, er en par-copula-konstruksjon med Student t copula for alle par den beste til å beskrive avhengighetsstrukturen for dataene presentert i dette kapittelet. Value at Risk-eksperimentene viser i tillegg at DCC GARCH- modellene beskrevet i oppgaven beskriver egenskapene til de finansielle tidsrekkene godt.

Det er viktig å påpeke at konklusjonene gjelder kun for akkurat dette datasettet.

Kapittel 10

Oppsummering og Konklusjon

I denne oppgaven har jeg presentert tre ulike DCC-GARCH-modeller for modellering av volatilitet i finansielle tidsrekker. Ved å bruke copulaer splittes den multivariate fordelingsfunksjonen til avkastningene i to deler. En for marginalene og en som fanger opp avhengigheten med en dynamisk multivariat copula. For å modellere avhengighetsstrukturen bruker jeg tre ulike copulaer; multivariat Student's t copula, en D-vine par-copula-konstruksjon med Student t copula for alle par av variable og en D-vine par-copula-konstruksjon med både Clayton copula og Student t copula.

Par-copula-konstruksjonene kan også defineres på andre måter enn det som er presentert i oppgaven. D-vine-konstruksjonen kan enkelt byttes ut med et C-vine-oppsett, og hvilke copulaer som tilpasses hvert enkelt par av variable kan velges fritt blant alle mulige bivariate copulaer.

Mens DCC-GARCH-modeller med kjente multivariate fordelinger krever at alle marginalfordelingene er de samme og at avhengighetsstrukturen er lik mellom alle aktiva, åpner bruk av copula opp for ulike marginalfordelinger for hver tidsrekke, mens par-copula-konstruksjonene gjør modellen enda mer fleksibel ved å tillate ulik avhengighetsstruktur mellom ulike par av aktiva. Å kombinere DCC-GARCH-teorien med teorien om copulaer og par-copula-konstruksjoner resulterer altså i mer fleksible modeller som fortsatt er relativt enkle å implementere i praksis.

Den største utfordringen med disse modellene er å velge den copulaen som best beskriver dataene. For multivariate copulaer er det i flere studier vist at innen finans er Student's t copulaen er overlegen i forhold til andre multivariate copulaer. Når det kommer til par-copula-konstruksjonen er det ikke like mange studier å støtte seg til og valg av bivariate copulaer er en utfordring. Scatterplott av dataene er en god hjelp, men som vi har sett i kapittel 9 er det ofte vanskelig å konkludere sikkert ut fra slike.

I kapittel 9 tilpassess modellene avkastningsdata fra Oslo Børs. Resultatene her viser at

alle de tre DCC-GARCH-modellene gir svært gode estimater for Value at Risk. Når man sammenligner modellene er det PCC-DCC-GARCH modellen med Student's t copula for alle par som gjør det aller best for akkurat dette datasettet. Dette trenger derimot ikke å være tilfelle for andre datasett.

En svakhet med PCC-DCC-GARCH-modellene er at de, som nevnt i kapittel 8, i noen tilfeller gir unøyaktige estimater for copulaparameterene. Men siden de estimerte parameterene i kapittel 9 holder seg innenfor området der parameterene antas å være godt estimert, antas resultatene i dette kapitlet å være tilfredsstillende. I andre DCC-GARCH-studier har jeg heller ikke sett noen estimerte α -verdier høyere enn 0.1. Det ser altså ut til at høye α -verdier sjelden forekommer i praksis.

Kapittel 11

Videre arbeid

Det er flere mulige retninger for videre arbeid. Hovedfokuset i denne oppgaven har vært på den delen av modellene som går på modellering av avhengighetsstrukturen i en portefølje. Å kombinere dette med mer nøyaktige univariate modeller vil kunne gi enda bedre resultater for modellen. En mulighet er å prøve ut modellen med andre marginalfordelinger enn Student t-fordelingen for de standardiserte feilene. Skjev Student t-fordeling åpner for eksempel opp for modellering av assymetri i avkastningene. Andre typer GARCH-modeller, som for eksempel EGARCH og QGARCH, er alternative marginale modeller som fanger opp assymetri i de betingede variansene.

Et annet forslag til videre arbeid er å jobbe mer med testing av robustheten til PCC-DCC-GARCH-modellene. Det kunne være interessant å studere hva som gjør at estimatene i noen tilfeller blir mer unøyaktige enn i andre.

En siste faktor som kunne forbedret modellene er å studere metoder for valg av copula i par-copula-konstruksjoner. Som nevnt i konklusjonen er den største utfordringen med PCC-DCC-GARCH modellene mangelen på gode metoder for å velge den copulaen som beskriver avhengighetsstrukturen best.

Bibliografi

- [1] Aas, Czado, Frigessi, and Bakken. Pair-copula constructions of multiple dependence. *Sonderforschungsbereich 386*, 2006.
- [2] Kjersti Aas. Modelling the Dependence Structure of Financial Assets: A Survey of Four Copulas. *SAMBA/22/04*, 2004.
- [3] Kjersti Aas and Xenia K. Diamakos. Statistical Modelling of Financial Time Series: An Introduction. *SAMBA/08/04*, 2004.
- [4] Tim Bedford and Roger M. Cooke. Vines—a new graphical model for dependent random variables. *Annals of Statistics*, 2002.
- [5] Daniel Berg and Kjersti Aas. Models for construction of multivariate dependence. *European Journal of Finance*, (15(7/8)):639–659, 2009.
- [6] C. Blanco and M Oks. Backtesting VaR models: Quantitative and Qualitative Tests. *The risk desk*, IV(1), 2004.
- [7] Tim Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31:307–1986, 1986.
- [8] Eike Christian Brechmann. *Truncated and simplified regular vines and their applications*. PhD thesis, Technische Universität München, 2010.
- [9] P. Christoffersen. Evaluating Interval Forecasts. *International Economic Review*, 39:841–862, 1998.
- [10] Stefano Demarta and Alexander J. McNeil. The t Copula and Related Copulas. *International Statistical Review*, 73(1):111–129, April 2004.
- [11] Robert F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance. *Econometrica*, 50(4):987–1007, 1982.

- [12] Robert F. Engle. Dynamic Conditional Correlation - A Simple Class of Multivariate GARCH Models. *Working paper series with number*, 2000.
- [13] Robert F. Engle and Kevin Sheppard. Theoretical and Empirical Properties of Dynamic Conditional Correlation Multivariate GARCH. *NBER Working Papers*, (8554), 2001.
- [14] Ingrid Hobæk Haff. Comparison of estimators for pair-copula constructions. *Journal of Multivariate Analysis*, 110:1–188, 2012.
- [15] Andreas Heinen and Alfonso Valdesogo. Asymmetric CAPM dependence for large dimensions: the Canonical Cine Autoregressive Model. 2008.
- [16] John C. Hull. *Options, Futures, and Other Derivatives*. Pearson education, 5 edition, 2003.
- [17] Alexander J. McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management - Concepts, Techniques and Tools*. Princeton series in finance. Princeton university press, 1 edition, 2005.
- [18] Harry Joe. Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. *Distributions with Ficed Marginals and Related Topics*, 1996.
- [19] Harry Joe and James J. Xu. The estimation method of inference functions for margins for multivariate models. Technical report, Department of statistics, University of British Columbia, 1996.
- [20] Dorota Kurowicka and Robert M. Cooke. Conditional, Partial and Rank Correlation for the Elliptical Copula: Dependence Modelling in Uncertainty Analysis. 2001. Delft University of Technology.
- [21] R. Mashal and A. Zeevi. Beyond Correlation: Extreme co-Movements Between Financial Assets. 2002. Technical report, Columbia University.
- [22] Elisabeth Orskaug. Multivariate DCC-GARCH Model- With Various Error Distributions. *SAMBA/19/09*, 2009.
- [23] The European Parliament and the Council of the European Union. Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvensy II). *Official Journal of the European Union*, 2009.
- [24] Andrew Patton. Estimation of copula models for time series of possibly different lengths. *Journal of Applied Econometrics*, 21(2):147–174, 2006.

- [25] P. Embrechts, A. McNeil, and D. Straumann. Correlation: Pitfalls and Alternatives.
- [26] Therèse Peters. Forecasting the Covariance Matrix With the DCC GARCH Model. *Examensarbete*, 4, Mai 2008.
- [27] P. Kuiper. Techniques for Verifying the Accuracy of Risk Measurement Models. *Journal of Derivatives*, 3:73–84, 1995.
- [28] David Posada and Keith A. Crandall. Selecting the Best-Fit Model of Nucleotide Substitution. *Syst. biol.*, 50(4):580–601, 2001.
- [29] Doris Schirmacher and Ernesto Schirmacher. Multivariate Dependence Modeling using Pair-Copulas. 2008.
- [30] Annastiina Silvennoinen. Multivariate GARCH models. *Working Paper Series in Economics and Finance*, (669), Januar 2008.
- [31] Tim Bollerslev. Modelling the Coherence in Short-run Nominal Exchange Rates: A Multivariate Generalized ARCH model. *The review of Economics and Statistics*, 72:490–505, 1990.
- [32] G. Udny Yule and Maurice Kendall. *An introduction to the theory of statistics*, chapter 12, pages 229–253. Charles Griffin and company, 5 edition, 1919.

Tillegg A

Testdata

For å teste at estimeringsalgoritmene mine er riktige før de anvendes på ekte data har jeg for de tre modellene som er brukt i oppgaven først generert et datasett som følger den gitte modellen. Deretter prøver jeg å estimere meg tilbake til disse parametrene med de metodene jeg har beskrevet tidligere i oppgaven. Parameterene som anvendes er selvvalgte, og alle algoritmer og koder som er benyttet finnes i appendiks D.

A.1 Copula- DCC- GARCH

A.1.1 Simulering

Jeg starter med den første modellen og simulerer et tredimensjonalt datasett fra Copula DCC GARCH- modellen der den multivariate fordelingen er satt sammen av en multivariat Student t copula og standardiserte Student t-fordelte marginaler.

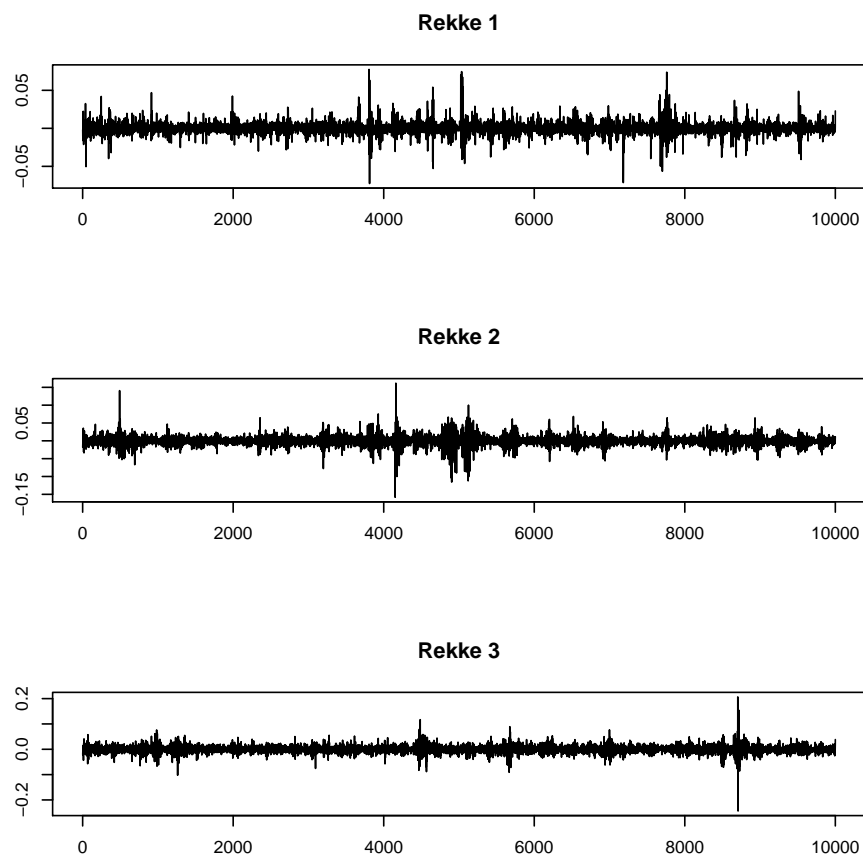
For de univariate GARCH-rekkene har jeg valgt følgende parametre, hentet fra [22] s.49:

| | α_0 | α_1 | β | ν |
|---------|------------------------|------------|---------|-------|
| rekke 1 | 2.28×10^{-06} | 0.116 | 0.854 | 3 |
| rekke 2 | 1.39×10^{-06} | 0.091 | 0.903 | 6 |
| rekke 3 | 3.99×10^{-06} | 0.117 | 0.863 | 9 |

For de resterende parameterene har jeg benyttet tallene fra eksperiment 2, s.71; $\nu_C = 7$, $a = 0.0066$, og $b = 0.947$. I tillegg har jeg valgt startmatrisen, \mathbf{Q}_1 til å være følgende:

$$Q_1 = \begin{pmatrix} 1 & 0.41 & 0.24 \\ 0.41 & 1 & 0.11 \\ 0.24 & 0.11 & 1 \end{pmatrix}$$

Jeg simulerer verdier for 10000 tidsenheter og får følgende tidsrekker:



Figur A.1: 1000 simulerte datapunkter fra Copula-DCC-GARCH- modellen.

A.1.2 Estimering av parametre

Som beskrevet i avsnitt 4.1.1 kan jeg, når jeg skal estimere meg tilbake til de opprinnelige parameterene i DCC-modellen, først optimere likelihooden for de univariate GARCH-rekkene og deretter anta disse som kjent når jeg skal optimere copula-likelihooden. Jeg bruker funksjonen `optim()` i R for å finne maximum likelihood estimatene. Estimeringen gir følgende parametre for alfaene, betaene og frihetsgradene:

| | $\widehat{\alpha}_0$ | $\widehat{\alpha}_1$ | $\widehat{\beta}$ | $\widehat{\nu}$ |
|---------|----------------------|----------------------|-------------------|-----------------|
| rekke 1 | $2.02e - 06$ | 0.129 | 0.862 | 2.86 |
| rekke 2 | $1.52e - 06$ | 0.0938 | 0.8986 | 5.95 |
| rekke 3 | $4.3e - 06$ | 0.112 | 0.8665 | 8.11 |

Mens jeg for a , b og ν_C får følgende parametre:

$$\begin{aligned}\nu_C &= 7.14 \\ a &= 0.00556 \\ b &= 0.96117\end{aligned}$$

Estimatene ligger nær nok simuleringsparameterene til at jeg konkluderer med at optimeringskoden er riktig implementert.

A.2 PCC- DCC GARCH

A.2.1 Simulering fra modellen med Student t copula for alle par

På samme måte som for modellen med multivariat Student t-copula og student-t fordelte marginaler, har jeg også for modellen med en D-vine par-copula-konstruksjon der alle de bivariate copulaene er Student-t-fordelte gjort en simulering. Jeg gjør simuleringen i 4 dimensjoner og bruker følgende parametre for de univariate rekkene:

| | α_0 | α_1 | β | ν |
|---------|------------------------|------------|---------|-------|
| rekke 1 | 2.28×10^{-06} | 0.116 | 0.854 | 4 |
| rekke 2 | 1.39×10^{-06} | 0.091 | 0.903 | 6 |
| rekke 3 | 3.99×10^{-06} | 0.117 | 0.863 | 8 |
| rekke 4 | 2.57×10^{-06} | 0.095 | 0.899 | 10 |

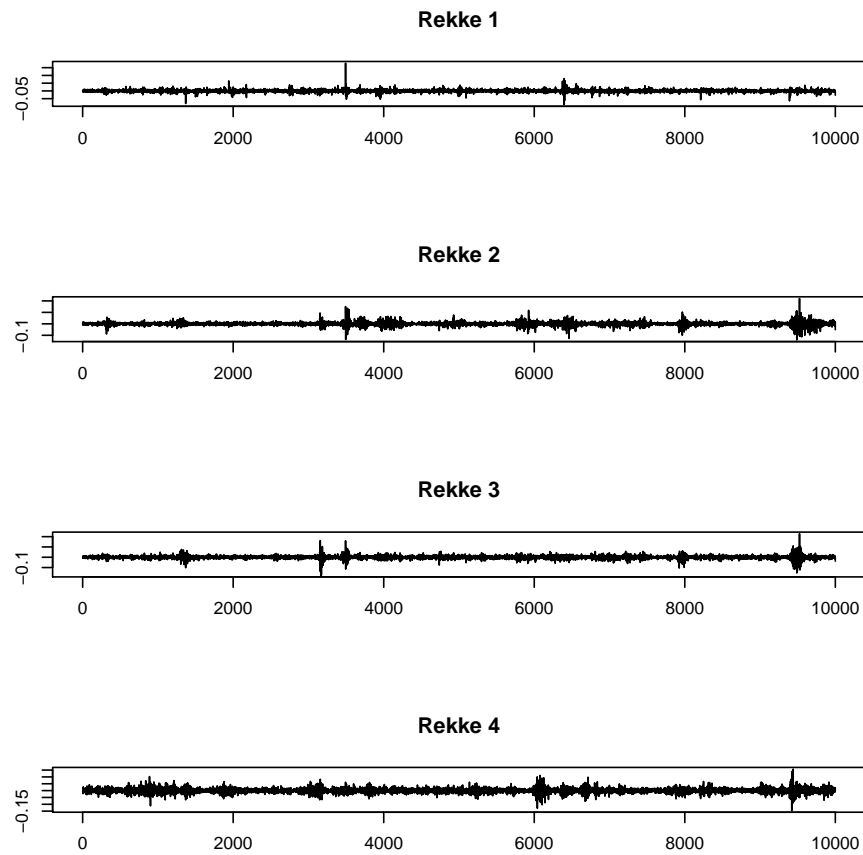
De tre første rekkene er de samme som jeg brukte for den forige modellen, mens den siste rekken er oppdiktet. For DCC-parametere har jeg valgt $a = 0.0066$ og $b = 0.947$, og par-copula-konstruksjonen er simulert med følgende copula-parametere:

| Copula | ν_C |
|-------------|---------|
| c_{12} | 3 |
| c_{23} | 5 |
| c_{34} | 7 |
| $c_{13 2}$ | 9 |
| $c_{24 3}$ | 11 |
| $c_{14 23}$ | 13 |

Startverdiene for Q_t er

$$Q_1 = \begin{pmatrix} 1 & -0.02 & 0.41 & 0.3 \\ -0.02 & 1 & 0.8 & 0.18 \\ 0.41 & 0.8 & 1 & 0.3 \\ 0.3 & 0.18 & 0.3 & 1 \end{pmatrix}.$$

Verdier for 10000 tidsenheter genereres, og de resulterende tidsrekkene er vist i figur A.2.



Figur A.2: 10000 simulerte datapunkter fra den første PCC-DCC-GARCH-modellen.

A.2.2 Estimering av parametre

Som for modellen med multivariat Student t- copula, optimeres først likelihooden for de univariate GARCH-rekkene og deretter antas disse som kjent når jeg skal optimere copula-likelihooden. Jeg bruker de samme algoritmene som tidligere for å finne parameterene i de univariate rekkene. Dette gir følgende parameterverdier:

| | α_0 | α_1 | β | ν |
|---------|--------------|------------|---------|-------|
| rekke 1 | $2.45e - 06$ | 0.1121 | 0.8505 | 4.13 |
| rekke 2 | $1.03e - 06$ | 0.0895 | 0.9092 | 5.79 |
| rekke 3 | $3.89e - 06$ | 0.1174 | 0.8644 | 8.07 |
| rekke 4 | $2.48e - 06$ | 0.0891 | 0.9038 | 10.2 |

Når jeg videre skal finne parameterene i de bivariate copulaene brukes, som beskrevet i avsnitt 5.4, først en sekvensiell algoritme for å finne gode startverdier til den fulle likelihooden. Jeg lar da korrelasjonsmatrisen være tidsvarierende i nivå 1 slik at jeg får estimater for a og b, mens i de andre nivåene lar jeg korrelasjonen være konstant siden det mest sannsynlig vil gi gode nok startverdier. Til slutt bruker jeg de estimerte startverdiene i optimeringen av den fulle likelihooden. Tabell A.1 viser de estimerte parameterene, både for den sekvensielle optimeringen og for den fulle.

| par | sekv | full | sann |
|-------------------|---------|---------|--------|
| a | 0.00326 | 0.00616 | 0.0066 |
| b | 0.98091 | 0.95477 | 0.947 |
| $\nu_{c_{12}}$ | 2.921 | 2.807 | 3 |
| $\nu_{c_{23}}$ | 4.919 | 4.889 | 5 |
| $\nu_{c_{34}}$ | 7.945 | 8.003 | 7 |
| $\nu_{c_{13 2}}$ | 8.727 | 7.616 | 9 |
| $\nu_{c_{24 3}}$ | 9.973 | 10.310 | 11 |
| $\nu_{c_{14 23}}$ | 11.675 | 12.944 | 13 |

Tabell A.1: Estimerte parametre fra begge optimeringsstegene i modell 2.

Også for denne modellen med dette parametersettet ser det ut til at jeg har implementert algoritmene riktig. Når jeg derimot varierer parameterene i simuleringen, viser det seg at jeg for noen parametersett får mer unøyaktige estimater enn andre. Dette har jeg skrevet mer om i kapittel 8.

A.3 Simulering fra modellen med Clayton og Student t par-copulaer

I den siste modellen jeg simulerer fra er D-vine-konstruksjonen bygd opp slik at jeg har Clayton copula for de to første parene i første nivå i D-vine-konstruksjonen og Student's t copula for de resterende parene (se figur 3.8).

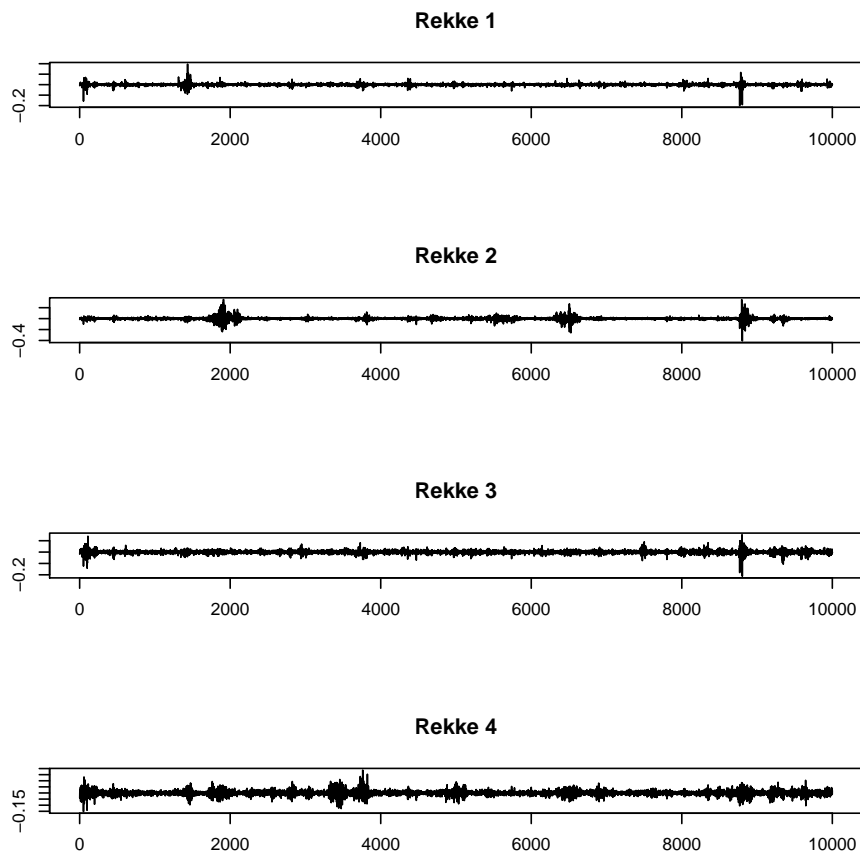
Parameterene i Clayton-copulaene blir da tidsvarierende og δ_t bestemmes ut fra \mathbf{R}_t ved ligningene (6.8) og (6.9). Som nevnt i avsnitt 6.1.2, må man ha positiv korrelasjon for å få gyldig parameter i Clayton-copulaen. Siden korrelasjonen varierer med tiden velger jeg å starte med relativt høy korrelasjon mellom de parene som skal beskrives med Clayton-copula for å være sikker på at variasjonen ikke gjør at δ_t blir negativ for noen t . Jeg starter med følgende \mathbf{Q} -matrise i simuleringen:

$$\mathbf{Q} = \begin{pmatrix} 1 & 0.7 & 0.76 & 0.3 \\ 0.7 & 1 & 0.8 & 0.4 \\ 0.76 & 0.8 & 1 & 0.3 \\ 0.3 & 0.4 & 0.3 & 1 \end{pmatrix}$$

Dette tilsvarer $\delta_{12,1} = 1.95$ og $\delta_{23,1} = 2.88$. For DCC-parameterene simulerer jeg med $a = 0.02$ og $b = 0.947$, og de resterende copula-parameterene velger jeg som følger:

| Copula | ν_C |
|-------------|---------|
| c_{34} | 3 |
| $c_{13 2}$ | 6 |
| $c_{24 3}$ | 9 |
| $c_{14 23}$ | 12 |

For de univariate GARCH-parameterene, brukes de samme som i den forrige modellen. Tidsrekkene som kommer ut er vist i figur A.3.



Figur A.3: 10000 datapunkter fra PCC-DCC-GARCH-modell nr.2

A.3.1 Estimering av parametre

Estimeringen av de univariate GARCH-parameterene er identisk med foregående modeller, så jeg velger å ikke ta den med her. Fremgangsmåten for å estimere de resterende parameterene blir også ganske lik. Den eneste forskjellen er at man må erstatte to av likelihoodene i første nivå med Clayton-likelihooden i stedet for Student's t. Estimertene jeg får ut er vist i tabell A.2.

| par | sekv | full | sann |
|------------------|---------|---------|-------|
| a | 0.0254 | 0.02358 | 0.02 |
| b | 0.93475 | 0.94079 | 0.947 |
| $\nu_{c_{12}}$ | 2.988 | 3.018 | 3 |
| $\nu_{c_{23}}$ | 4.793 | 5.448 | 6 |
| $\nu_{c_{34}}$ | 7.385 | 7.048 | 9 |
| $\nu_{c_{13 2}}$ | 13.174 | 13.393 | 12 |

Tabell A.2: Estimerte parametre i modell 3

Ved bruk av modellen på ekte data, bør man også finne den estimerte \bar{Q} -matrisen for å kunne si noe om Clayton-parameterene. Her er målet å sjekke om algoritmene ser ut til å være riktige, som jeg konkluderer med at de gjør, så her er ikke det nødvendig.

Tillegg B

h-funksjoner

B.1 Student t-copula

Denne utregningen finnes også i [1], men jeg har valgt å gjøre den litt mer utfyllende.

Om man setter $\mathbf{u} = (u_1, u_2)$ inn i (3.6) og integrerer, får man følgende uttrykk for den bivariate Student-t-copulaen:

$$\begin{aligned} C_{12}(u_1, u_2) &= \int_{-\infty}^{t_{\nu_{12}}^{-1}(u_1)} \int_{-\infty}^{t_{\nu_{12}}^{-1}(u_2)} \frac{\Gamma(\frac{\nu_{12}+2}{2})}{\Gamma(\frac{\nu_{12}}{2})\sqrt{(\pi\nu_{12})^2(1-\rho^2)}} \left(1 + \frac{x^2 - 2\rho_{12}xy + y^2}{\nu_{12}(1-\rho_{12}^2)}\right)^{-(\nu_{12}+2)/2} dy dx \\ &= \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} g(x, y) dy dx \end{aligned}$$

$t_{\nu_{12}}(u)$ er den kumulative fordelingsfunksjonen til Student t- fordelingen, og den korresponderende tettheten, $f_{\nu_{12}}(x)$, er som i (3.4).

Bruker så (5.4) for å finne uttrykket for $h_{12}(u_1, u_2)$:

$$\begin{aligned} h_{12}(u_1, u_2) &= F_{1|2}(u_1|u_2) \\ &= \frac{\partial}{\partial u_2} C_{12}(u_1, u_2) \\ &= \frac{\partial}{\partial u_2} \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} g(x, y) dy dx \\ &= \frac{\partial b_2}{\partial u_2} \frac{\partial}{\partial b_2} \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} g(x, y) dy dx \end{aligned} \tag{B.1}$$

Generelt har vi at

$$\begin{aligned} F(F^{-1}) &= x \\ f(F^{-1}(x)) \frac{d}{dx} (F^{-1}(x)) &= 1 \\ \frac{d}{dx} (F^{-1}(x)) &= \frac{1}{f(F^{-1}(x))} \end{aligned}$$

Setter vi dette inn for første faktor i (B.1), får vi

$$\begin{aligned} h_{12}(u_1, u_2) &= \frac{1}{f_{\nu_{12}}(b_2)} \frac{\partial}{\partial b_2} \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} g(x, y) dy dx \\ &= \frac{1}{f_{\nu_{12}}(b_2)} \int_{-\infty}^{b_1} \left[\frac{\partial}{\partial b_2} \int_{-\infty}^{b_2} g(x, y) dy \right] dx \\ &= \frac{1}{f_{\nu_{12}}(b_2)} \int_{-\infty}^{b_1} g(x, b_2) dx \\ &= \frac{1}{f_{\nu_{12}}(b_2)} \int_{-\infty}^{b_1} \frac{\Gamma(\frac{\nu_{12}+2}{2})}{\Gamma(\frac{\nu_{12}}{2}) \sqrt{(\pi \nu_{12})^2 (1-\rho^2)}} \\ &\quad \times \left(1 + \frac{x^2 - 2\rho_{12}xb_2 + b_2^2}{\nu_{12}(1-\rho_{12}^2)} \right)^{-(\nu_{12}+2)/2} dx \\ &= \frac{1}{f_{\nu_{12}}(b_2)} \int_{-\infty}^{b_1} \frac{\Gamma(\frac{\nu_{12}+2}{2})}{\Gamma(\frac{\nu_{12}}{2}) \sqrt{(\pi \nu_{12})^2 (1-\rho^2)}} \\ &\quad \times \left(1 + \frac{(x - \rho_{12}b_2)^2}{\nu_{12}(1-\rho_{12}^2)} \right)^{-(\nu_{12}+2)/2} \left(1 + \frac{b_2^2}{\nu_{12}} \right)^{-(\nu_{12}+2)/2} dx. \end{aligned}$$

Ønsker nå å få uttrykket på en slik form at vi kjenner igjen student-t fordelingen utenfor integralet og student t-copulaen inni integralet. Multipliserer og dividerer derfor med $\Gamma(\frac{\nu_{12}+1}{2}) \sqrt{\pi(\nu_{12} + b_2^2)(1 - \rho_{12}^2)}$, splitter siste faktor opp i to faktorer og trekker ut av integralet det som er beleilig. Vi står altså igjen med

$$\begin{aligned}
h_{12}(u_1, u_2) &= \frac{1}{f_{\nu_{12}}(b_2)} \frac{\Gamma(\frac{\nu_{12}+1}{2})\Gamma(\frac{\nu_{12}+1}{2})\sqrt{\pi(\nu_{12}+b_2^2)(1-\rho_{12}^2)}}{\Gamma(\frac{\nu_{12}}{2})\sqrt{(\pi\nu_{12})^2(1-\rho_{12}^2)}} \left(1 + \frac{b_2^2}{\nu_{12}}\right)^{-(\nu_{12}+1)/2} \left(1 + \frac{b_2^2}{\nu_{12}}\right)^{-1/2} \\
&\times \int_{-\infty}^{b_1} \frac{\Gamma(\frac{\nu_{12}+2}{2})}{\Gamma(\frac{\nu_{12}+1}{2})\sqrt{\pi(\nu_{12}+b_2^2)(1-\rho_{12}^2)}} \left(1 + \frac{(x-\rho_{12}b_2)2}{(\nu_{12}+b_2)^2(1-\rho_{12}^2)}\right)^{-(\nu_{12}+2)/2} dx \\
&= \frac{f_{\nu_{12}}(b_2)}{f_{\nu_{12}}(b_2)} \frac{\sqrt{\pi(\nu_{12}+b_2^2)(1-\rho_{12}^2)}}{\sqrt{(\pi\nu_{12})(1-\rho_{12}^2)}} \left(1 + \frac{b_2^2}{\nu_{12}}\right)^{-1/2} \\
&\times \int_{-\infty}^{b_1} \frac{\Gamma(\frac{\nu_{12}+2}{2})}{\Gamma(\frac{\nu_{12}+1}{2})\sqrt{\pi(\nu_{12}+b_2^2)(1-\rho_{12}^2)}} \left(1 + \frac{(x-\rho_{12}b_2)2}{(\nu_{12}+b_2)^2(1-\rho_{12}^2)}\right)^{-(\nu_{12}+2)/2} dx \\
&= \int_{-\infty}^{b_1} \frac{\Gamma(\frac{\nu_{12}+2}{2})}{\Gamma(\frac{\nu_{12}+1}{2})\sqrt{\pi(\nu_{12}+b_2^2)(1-\rho_{12}^2)}} \left(1 + \frac{(x-\rho_{12}b_2)2}{(\nu_{12}+b_2)^2(1-\rho_{12}^2)}\right)^{-(\nu_{12}+2)/2} dx.
\end{aligned}$$

Sett så

$$\begin{aligned}
\nu &= \nu_{12} + 1 \\
\mu &= \rho_{12}b_2 \\
\sigma^2 &= \frac{\nu_{12} + b_2}{\nu_{12} + 1} (1 - \rho_{12}^2).
\end{aligned}$$

Vi har da

$$\begin{aligned}
h_{12}(u_1, u_2) &= \int_{-\infty}^{b_1} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu\sigma}} \left(1 + \frac{1}{\nu} \left(\frac{x-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2} dx \\
&= \int_{-\infty}^{b_1} \frac{1}{\sigma} f_{\nu} \left(\frac{x-\mu}{\sigma}\right) dx \\
&= \int_{-\infty}^{(b_1-\mu)/\sigma} f_{\nu}(z) dz \\
&= t_{\nu} \left(\frac{b_1-\mu}{\sigma}\right), \tag{B.2}
\end{aligned}$$

hvor $t_{\nu}(x) = \int_{-\infty}^x f_{\nu}(y) dy$. Hvis vi nå setter inn uttrykkene for b_1 , ν , σ og μ i uttrykket, står vi igjen med

$$h_{12}(u_1, u_2) = t_{\nu_{12}+1} \left(\frac{t_{\nu_{12}}^{-1}(u_1) - \rho_{12} t_{\nu_{12}}^{-1}(u_2)}{\sqrt{\frac{(\nu_{12} + (t_{\nu_{12}}^{-1}(u_2))^2)(1 - \rho_{12}^2)}{\nu_{12} + 1}}} \right) \quad (\text{B.3})$$

B.2 Clayton copula

Siden fordelingsfunksjonen for den bivariate Clayton copulaen er gitt på eksplisitt form ved ligning (3.8), er utregningen av h -funksjonen for denne copulaen mindre kompleks, og vi får

$$\begin{aligned} h_{12}(u_1, u_2) &= F_{1|2}(u_1|u_2) \\ &= \frac{\partial}{\partial u_2} C_{12}(u_1, u_2) \\ &= \frac{\partial}{\partial u_2} (u_1^{-\delta_{12}} + u_2^{-\delta_{12}} - 1)^{-\frac{1}{\delta_{12}}} \\ &= -\frac{1}{\delta_{12}} (u_1^{-\delta_{12}} + u_2^{-\delta_{12}} - 1)^{-\left(\frac{1}{\delta_{12}} - 1\right)} \times -\delta_{12} u_2^{-\delta_{12} - 1} \\ &= u_2^{-(\delta_{12} - 1)} (u_1^{-\delta_{12}} + u_2^{-\delta_{12}} - 1)^{-\left(\frac{1}{\delta_{12}} - 1\right)}. \end{aligned} \quad (\text{B.4})$$

Tillegg C

Algoritme 2 og 4

Algoritmene i dette appendixet er hentet fra [1]. Siden jeg kun har brukt D-vines i mine simuleringer og estimeringer tar jeg bare med algoritme 2 og 4, og utelater algoritme 1 og 3 som gjelder C-vines.

C.1 Algoritme 2

Simuleringsalgoritme for en D-vine. Genererer et tilfeldig utvalg x_1, \dots, x_n fra vinen.

Trekk ω_i ; $i = 1, \dots, n$ fra $U(0, 1)$.

$$x_1 = v_{1,1} = \omega_1$$

$$x_2 = v_{2,1} = h^{-1}(\omega_2, v_{1,1}, \Theta_{1,1})$$

$$v_{2,2} = h(v_{1,1}, v_{2,1}, \Theta_{1,1})$$

for $i \leftarrow 3, 4, \dots, n$

$$v_{i,1} = \omega_i$$

for $k \leftarrow i - 1, \dots, 2$

$$v_{i,1} = h^{-1}(v_{i,1}, v_{i-1,2k-2}, \Theta_{k,i-k})$$

end for

$$v_{i,1} = h^{-1}(v_{i,1}, v_{i-1,1}, \Theta_{1,i-1})$$

$$x_i = v_{i,1}$$

if $i == n$ **then**

Stop

end if

$$v_{i,2} = h(v_{i-1,1}, v_{i,1}, \Theta_{1,i-1})$$

$$v_{i,3} = h(v_{i,1}, v_{i-1,1}, \Theta_{1,i-1})$$

if $i > 3$ **then** **for** $j \leftarrow 2, 3, \dots, i - 2$

```

     $v_{i,2j} == h(v_{i-1,2j-2}, v_{i,2j-1}, \Theta_{j,i-j})$ 
     $v_{i,2j+1} == h(v_{i-1,2j-1}, v_{i-1,2j-2}, \Theta_{j,i-j})$ 
  end for
end if
 $v_{i,2i-2} = h(v_{i-1,2i-4}, v_{i,2i-3}, \Theta_{i-1,1})$ 
end for

```

For å beregne de betingede fordelingsfunksjonene brukes h-funksjonen, definert ved 5.4, med de foregående beregnede betingede fordelingsfunksjonene, $v_{i,j} = F(x_i | x_1, \dots, x_{j-1})$, som de to første argumentene. Det siste argumentet i h-funksjonen, $\Theta_{i,j}$ er parametersettet til copulattettheten $c_{i,i+j|i+1,\dots,i+j-1}$.

C.2 Algoritme 4

Likelihoodevaluering for en D-vine.

```

log-likelihood=0
for  $i \leftarrow 1, 2, \dots, n$ 
     $v_{0,i} = \mathbf{x}_i$ 
end for
for  $i \leftarrow 1, 2, \dots, n-1$ 
    log-likelihood=log-likelihood +  $L(\mathbf{v}_{0,i}, \mathbf{v}_{0,i+1}, \Theta_{1,i})$ 
end for
 $v_{1,1} = h(\mathbf{v}_{0,1}, \mathbf{v}_{0,2}, \Theta_{1,1})$ 
for  $k \leftarrow 1, 2, \dots, n-3$ 
     $v_{1,2k} = h(\mathbf{v}_{0,k+2}, \mathbf{v}_{0,k+1}, \Theta_{1,k+1})$ 
     $v_{1,2k+1} = h(\mathbf{v}_{0,k+1}, \mathbf{v}_{0,k+2}, \Theta_{1,k+1})$ 
end for
 $v_{1,2n-4} = h(\mathbf{v}_{0,n}, \mathbf{v}_{0,n-1}, \Theta_{1,n-1})$ 
for  $j \leftarrow 2, \dots, n-1$ 
    for  $i \leftarrow 1, 2, \dots, n-j$ 
        log-likelihood=log-likelihood+ $L(\mathbf{v}_{j-1,2i-1}, \mathbf{v}_{j-1,2i}, \Theta_{j,i})$ 
    end for
    if  $j == n-1$  then
        Stop    end if     $v_{j,1} = h(\mathbf{v}_{j-1,1}, \mathbf{v}_{j-1,2}, \Theta_{j,1})$ 

```

```

if  $n > 4$  then
  for  $i \leftarrow 1, 2, \dots, n - j - 2$ 
     $\mathbf{v}_{j,2i} = h(\mathbf{v}_{j-1,2i+2}, \mathbf{v}_{j-1,2i+1}, \Theta_{j,i+1})$ 
     $\mathbf{v}_{j,2i+1} = h(\mathbf{v}_{j-1,2i+1}, \mathbf{v}_{j-1,2i+2}, \Theta_{j,i+1})$ 
  end for
end if
 $\mathbf{v}_{j,2n-2j-2} = h(\mathbf{v}_{j-1,2n-2j}, \mathbf{v}_{j-1,2n-2j-1}, \Theta_{j,n-j})$ 
end for

```


Tillegg D

R-kode

D.1 Dataanalyse

Innlesing av data:

```
> indekser<-read.table("sektorindekser.csv",sep=";")
> data<-indekser
```

D.1.1 Diverse plott

Plott av log-avkastningene:

```
>par(mfrow=c(4,1))
>plot(diff(log(data[,1])),type="l",main="Materials",
      ylab="",xlab="Tid")
>plot(diff(log(data[,2])),type="l",main="Industrials",
      ylab="",xlab="Tid")
>plot(diff(log(data[,3])),type="l",main="Consumer Discretionary",
      ylab="",xlab="Tid")
>plot(diff(log(data[,4])),type="l",main="Information Technology",
      ylab="",xlab="Tid")
>par(mfrow=c(1,1))
```

Autokorrelasjons- og krosskorrelasjonsplottene:

```

>indeks<-cbind(diff(log(data[,1])),diff(log(data[,2])),
               diff(log(data[,3])),diff(log(data[,4])))
>aksjer<-indeks
>par(mfrow=c(4,1))
>acf((aksjer[,1]),main="Materials",ylim=c(-0.2,0.2))
>acf((aksjer[,2]),main="Industrials",ylim=c(-0.2,0.2))
>acf((aksjer[,3]),main="Consumer Discretionary",ylim=c(-0.2,0.2))
>acf((aksjer[,4]),main="Information technology",ylim=c(-0.2,0.2))
>par(mfrow=c(1,1))

>par(mfrow=c(4,1))
>acf((aksjer[,1])^2,main="Materials",ylim=c(-0.2,0.5))
>acf((aksjer[,2])^2,main="Industrials",ylim=c(-0.2,0.5))
>acf((aksjer[,3])^2,main="Consumer Discretionary",ylim=c(-0.2,0.5))
>acf((aksjer[,4])^2,main="Information technology",ylim=c(-0.2,0.5))
>par(mfrow=c(1,1))

>par(mfrow=c(4,1))
>acf(abs(aksjer[,1]),main="Materials",ylim=c(-0.2,0.5))
>acf(abs(aksjer[,2]),main="Industrials",ylim=c(-0.2,0.5))
>acf(abs(aksjer[,3]),main="Consumer Discretionary",ylim=c(-0.2,0.5))
>acf(abs(aksjer[,4]),main="Information technology",ylim=c(-0.2,0.5))
>par(mfrow=c(1,1))

>par(mfrow=c(3,1))
>ccf((aksjer[,1]),(aksjer[,2]),
      main="Materials og Industrials")
>ccf((aksjer[,1]),(aksjer[,3]),
      main="Materials og Consumer Discretionary")
>ccf((aksjer[,1]),(aksjer[,4]),
      main="Materials og Information technology")

```

```

>ccf((aksjer[,2]),(aksjer[,3]),
      main="Industrials og Consumer Discretionary")
>ccf((aksjer[,2]),(aksjer[,4]),
      main="Industrials og Information technology")
>ccf((aksjer[,3]),(aksjer[,4]),
      main="Consumer Discretionary og Information Technology")

>ccf(abs(aksjer[,1]),abs(aksjer[,2]),
      main="Materials og Industrials")
>ccf(abs(aksjer[,1]),abs(aksjer[,3]),
      main="Materials og Consumer Discretionary")
>ccf(abs(aksjer[,1]),abs(aksjer[,4]),
      main="Materials og Information Technology")

>ccf(abs(aksjer[,2]),abs(aksjer[,3]),
      main="Industrials og Consumer Discretionary")
>ccf(abs(aksjer[,2]),abs(aksjer[,4]),
      main="Industrials og Information Technology")
>ccf(abs(aksjer[,3]),abs(aksjer[,4]),
      main="Consumer Discretionary og Information Technology")
>par(mfrow=c(1,1))

```

Scatterplott av de seks parene i D-vine-strukturen:

```

>plot(Fx2[,1],Fx2[,2],xlab="M",ylab="IT")
>plot(Fx2[,2],Fx2[,3],xlab="IT",ylab="I")
>plot(Fx2[,3],Fx2[,4],xlab="I",ylab="CD")
>plot(engitttto,tregittto,xlab="F(M|IT)",ylab="F(I|IT)")
>plot(togitttre,firegitttre,xlab="F(IT|I)",ylab="F(CD|I)")
>plot(Fx1x2x3,Fx4x2x3,xlab="F(M|IT,I)",ylab="F(CD|IT,I)")

```

D.1.2 Valg av par-copula-konstruksjon

Kode for å tilpasse hvert par av risikofaktorer til en bivariat Student's t-copula og beregne haleavhengigheten:

```

>optimizer<-optim(par<-c(15,0.2),fn<-Lt,data=Fx[,2:3])
>ICD<-optimizer$par[1]
>ICD2<-optimizer$par[2]
>ICD3<-2*pt((-sqrt(ICD+1))*sqrt((1-ICD2)/(1+ICD2)),df=ICD)

>optimizer<-optim(par<-c(15,0.2),fn<-Lt,data=Fx[,3:4])
>CDIT<-optimizer$par[1]
>CDIT2<-optimizer$par[2]
>CDIT3<-2*pt((-sqrt(CDIT+1))*sqrt((1-CDIT2)/(1+CDIT2)),df=CDIT)

>optimizer<-optim(par<-c(15,0.2),fn<-Lt,data=cbind(Fx[,1],Fx[,3]))
>MCD<-optimizer$par[1]
>MCD2<-optimizer$par[2]
>MCD3<-2*pt((-sqrt(MCD+1))*sqrt((1-MCD2)/(1+MCD2)),df=MCD)

>optimizer<-optim(par<-c(15,0.2),fn<-Lt,data=cbind(Fx[,1],Fx[,4]))
>MIT<-optimizer$par[1]
>MIT2<-optimizer$par[2]
>MIT3<-2*pt((-sqrt(MIT+1))*sqrt((1-MIT2)/(1+MIT2)),df=MIT)

>optimizer<-optim(par<-c(15,0.2),fn<-Lt,data=cbind(Fx[,2],Fx[,4]))
>IIT<-optimizer$par[1]
>IIT2<-optimizer$par[2]
>IIT3<-2*pt((-sqrt(IIT+1))*sqrt((1-IIT2)/(1+IIT2)),df=IIT)

```

Her er `optim` en ferdig utviklet funksjon i R som minimerer funksjonen den blir brukt på. Det er derfor viktig at man i funksjonen for likelihooden returnerer den negative likelihoodverdien siden det er optimering som er interessant her.

Likelihoodfunksjon for bivariat Student's t-copula:

```

> Lt<-function(theta,data){
+   nu<-theta[1]
+   rho<-theta[2]
+   u<-data[,1]
+   v<-data[,2]
+   L<-log(gamma(0.5*(nu+2)))-log(gamma(0.5*nu))-log(nu*pi)

```

```

      -log(dt(qt(u,nu),nu))-log(dt(qt(v,nu),nu))
      -log(sqrt(1-rho^2))+(-0.5*(nu+2))*log(1+(qt(u,nu)^2
      +qt(v,nu)^2-2*rho*qt(u,nu)*qt(v,nu))/(nu*(1-rho^2)))
+   likli<-sum(L)

#Returnerer negativ verdi

+   return(-likli)
+ }

```

Organisering av data etter valgt struktur:

```

> aksjer2<-cbind(aksjer[,2],aksjer[,4],aksjer[,1],aksjer[,3])
> Fx2<-cbind(Fx[,2],Fx[,4],Fx[,1],Fx[,3])

```

H-funksjon og dens inverse for Student's t copula og Clayton copula:

```

#Student's t-copula

> H<-function(x,y,nu,r){
+   H<-pt((qt(x,nu)-r*qt(y,nu))/(sqrt(((nu+qt(y,nu)^2
      *(1-r^2))/(nu+1))),nu+1)
+ }

>hinvers<-function(x,y,nu,r){
+   hinvers<-pt(sqrt(((nu+qt(y,nu)^2)*(1-r^2))/(nu+1))*qt(x,nu+1)
+     +r*qt(y,nu),nu)
+}

#Clayton-copula

> Hclay<-function(u,v,delta){
+   h<-v^(-delta-1)*(u^(-delta)+v^(-delta)-1)^(-1-(1/delta))
+   return(h)
+ }

>Hclayinv<-function(u,v,delta){

```

```

+ h<-((u*v^(delta+1))^-delta/(delta+1)+1-v^(-delta))^-1/delta)
+
+ return(h)
>}

```

Beregning av betingede kumulative fordelinger (Parametrisk tilpassing ved hjelp av h-funksjon).

```

> ento<-optim(par<-c(15,0.2),fn<-Lt,data=Fx2[,1:2])
> engittto<-H(Fx2[,1],Fx2[,2],ento$par[1],ento$par[2])
> treto<-optim(par<-c(15,0.2),fn<-Lt,data=Fx2[,2:3])
> tregittto<-H(Fx2[,3],Fx2[,2],treto$par[1],treto$par[2])
> togitttre<-H(Fx2[,2],Fx2[,3],treto$par[1],treto$par[2])
> trefire<-optim(par<-c(15,0.2),fn<-Lt,data=Fx2[,3:4])
> firegitttre<-H(Fx2[,4],Fx2[,3],treto$par[1],treto$par[2])
> entotre<-optim(par<-c(15,0.2),fn<-Lt,
                 data=cbind(engittto,tregittto))
> Fx1x2x3<-H(engittto,tregittto,entotre$par[1],entotre$par[2])
> firetotre<-optim(par<-c(15,0.2),fn<-Lt,
                 data=cbind(firegitttre,togitttre))
> Fx4x2x3<-H(firegitttre,togitttre,firetotre$par[1],firetotre$par[2])

```

D.2 Simulering fra de ulike copulaene

D.2.1 Implisitte copulaer

Funksjon som gjør n d-dimensjonale trekninger fra en Gaussisk copula:

```

> rgauss<-function(n,R){
+   mu<-rep(0,dim(R)[1])
+   x<-rmvnorm(n,mu,R)
+   rgauss<-apply(x,1,pnorm)
+   return(t(rgauss))
+ }

```

Funksjon som gjør n d-dimensjonale trekninger fra en Student's t- copula:

```

> rstudt<-function(n,nuC,R){
+ x<-rmvt(n,R,df=nuC)
+ rstudt<-apply(x,1,pt,df=nuC)
+ return(t(rstudt))
+ }

```

D.2.2 Eksplisitte copulaer

Funksjon som gjør n to-dimensjonale trekninger fra en Clayton-copula:

```

> trekk<-function(gam,delta){
+ un<-runif(2)
+ rclay<-(1-log(un)/gam)^(-1/delta)
+ return(rclay)
+ }
> rclayton<-function(n,delta){
+ gam<-rgamma(n,1/delta,1)
+ rclayton<-apply(as.matrix(gam),1,trekk,delta=delta)
+ return(t(rclayton))
+ }

```

Når det gjelder Gumbel-copulaen har jeg valgt å ikke lage en egen funksjon for å gjøre trekninger fra denne. Når jeg lager plottet i kapittel 2 bruker jeg den ferdige funksjonen rcopula fra pakken copula i R.

D.3 Simulering fra 3-dimensjonal Copula DCC-GARCH

Funksjonen simCDCC simulerer n trekninger fra en Copula-DCC-GARCH- modell med DCC-parametere a og b, og resterende parametere som i kapittel A.1.1.

```

simCDCC<-function(n,a,b){
+ dim<-3
+ n<-n
+ r<-matrix(rep(0,n*dim),ncol=dim)
+ h<-matrix(rep(0,n*dim),ncol=dim)
+ epsilon<-matrix(rep(0,n*dim),ncol=dim)
+ a<-a

```

```

+ b<-b
+ qstrek<-matrix(c(1,0.41,0.24,0.41,1,0.11,0.24,0.11,1),nrow=3,byrow=T)

#GARCH-parametere:

+ uni<-matrix(c(2.28*10^-06,0.116,0.854,1.39*10^-06,0.091,
+               0.903,3.99*10^-06,0.117,0.863),byrow=T,nrow=3)

#Parametere i marginalfordelinger og copula:

+ ny1<-3
+ stand1<-sqrt((ny1-2)/ny1)
+ ny2<-6
+ stand2<-sqrt((ny2-2)/ny2)
+ ny3<-9
+ stand3<-sqrt((ny3-2)/ny3)
+ nuC<-7
+ stand<-c(stand1,stand2,stand3)
+ Qi=qstrek
+ R=qstrek

#Som startverdi på variansen, h, brukes den asymptotiske variansen:

+ asymp1<-uni[1,1]/(1-uni[1,2]-uni[1,3])
+ asymp2<-uni[2,1]/(1-uni[2,2]-uni[2,3])
+ asymp3<-uni[3,1]/(1-uni[3,2]-uni[3,3])

#Trekker en trekning fra Student's t-copulaen:

+ multit<-rmvt(1, R, df = nuC)
+ finn_u<-function(x){pt(x,nuC)}
+ u<-apply(multit,1,finn_u)
+ epsilon[1,]<-c(qst(u[1],location=0,scale=stand1,shape=0,df=ny1),
+               qst(u[2],location=0,scale=stand2,shape=0,df=ny2),
+               qst(u[3],location=0,scale=stand3,shape=0,df=ny3))

+ h[1,]<-c(asymp1,asymp2,asymp3)
+ r[1,]<-epsilon[1,]*sqrt(h[1,])

#For hvert tidspunkt t:
# Beregn volatiliteten i de univariate rekkene, og lag matrisen  $D_t$ .

```



```

# Beregn korrelasjonsmatrisen,  $\mathbf{R}_t$ .
# Generer en tilfeldig trekning fra den d-dimensjonale copulaen med korrelasjonsmatrise  $\mathbf{R}_t$ 
og  $\nu_C$  frihetsgrader.
# Beregn t-fordelte observasjoner ved å benytte den inverse fordelingsfunksjonen til den
standardiserte t-fordelingen.

+ for (i in 2:n){
+   h[i,]=c(uni[1,1]+uni[1,2]*r[i-1,1]^2+uni[1,3]*h[i-1,1],
+           uni[2,1]+uni[2,2]*r[i-1,2]^2+uni[2,3]*h[i-1,2],
+           uni[3,1]+uni[3,2]*r[i-1,3]^2+uni[3,3]*h[i-1,3])
+   Di<-sqrt(diag(h[i,]))
+   Qi=(1-a-b)*qstrek+a*epsilon[i-1,]%*%t(epsilon[i-1,])+b*Qi
+   Qdiag=diag(Qi)
+   Q_star=diag(sqrt(Qdiag))
+   R<-solve(Q_star)%*%Qi%*%solve(Q_star)
+
+   multit<-rmvt(1, R, df = nuC)
+   finn_u<-function(x){pt(x,nuC)}
+   u<-apply(multit,1,finn_u)
+
+   epsilon[i,]<-c(qst(u[1],location=0,scale=stand1,shape=0,df=ny1),
+                 qst(u[2],location=0,scale=stand2,shape=0,df=ny2),
+                 qst(u[3],location=0,scale=stand3,shape=0,df=ny3))

+   r[i,]<-Di%*%epsilon[i,]
+
+ }
return(r)
}

```

D.4 Estimering av parameter i Copula DCC-GARCH

For å få gyldige parametere må jeg gjøre noen transformasjoner når jeg estimerer. Jeg optimerer, som i [19], for eksempel over $\log(\nu)$ for å være sikker på å få ν positiv, mens optimering over

$$\frac{\log(\alpha_1)}{1 + \log(\alpha_1) + \log(\alpha_2)}$$

og

$$\frac{\log(\alpha_1)}{1 + \log(\alpha_1) + \log(\alpha_2)}$$

sikrer $\alpha > 0$, $\beta > 0$ og $\alpha + \beta < 1$.

D.4.1 Optimering

Jeg finner først parameterene i de univariate modellene ved å optimere 4.5:

```
#Definerer initialverdier
```

```
> paraminit<-rep(0,4)
> paraminit[2]<-log(0.1)/(1+log(0.1)+log(0.8))
> paraminit[3]<-log(0.8)/(1+log(0.1)+log(0.8))
> paraminit[1]<-log(var(r[1,])/(1-0.8/1.9-0.1/1.9))
> paraminit[4]<-log(6)
```

#For hver univariate rekke, brukes funksjonen optim() for å optimere den egendefinerte likelihoodfunksjonen, t.loglike():

```
> optimere<-optim(paraminit,t.loglike,data=r[,1])
> rekke1_est<-c(exp(optimere$par[1]),
                exp(optimere$par[2])/
                (1+exp(optimere$par[2])+exp(optimere$par[3])),
                exp(optimere$par[3])/
                (1+exp(optimere$par[2])+exp(optimere$par[3])),
                exp(optimere$par[4]))

> paraminit[1]<-log(var(r[2,])/(1-0.8/1.9-0.1/1.9))
> optimere<-optim(paraminit,t.loglike,data=r[,2])
> rekke2_est<-c(exp(optimere$par[1]),
                exp(optimere$par[2])/
                (1+exp(optimere$par[2])+exp(optimere$par[3])),
                exp(optimere$par[3])/
                (1+exp(optimere$par[2])+exp(optimere$par[3])),
                exp(optimere$par[4]))
```

```

> paraminit[1]<-log(var(r[3,])/(1-0.8/1.9-0.1/1.9))
> optimere<-optim(paraminit,t.loglike,data=r[,3])
>rekke3_est<-c(exp(optimere$par[1]),
               exp(optimere$par[2])/
               (1+exp(optimere$par[2])+exp(optimere$par[3])),
               exp(optimere$par[3])/
               (1+exp(optimere$par[2])+exp(optimere$par[3])),
               exp(optimere$par[4]))

> uniparam<-rbind(rekke1_est,rekke2_est,rekke3_est)

```

Når dette er gjort, sender jeg de parameterene jeg finner inn i likelihooden for Student t-copulaen, (4.6), og optimerer denne. Også her gjør jeg transformasjoner for å sikre gyldige parametere:

```

#Definerer initialverdier:

> paramin<-rep(0,3)
> paramin[1]<-log(0.1)/(1+log(0.1)+log(0.8))
> paramin[2]<-log(0.8)/(1+log(0.1)+log(0.8))
> paramin[3]<-log(6)

#Optimerer likelihoodrfunksjonen, copula.like, ved bruk av optim():

> optimizer<-optim(par=paramin,fn=copula.like,data=r,
                  uniparam=uniparam,d=3)
> trall<-optimizer$par
>estimater<-c(exp(trall[1])/(1+exp(trall[1])+exp(trall[2])),
              exp(trall[2])/(1+exp(trall[1])+exp(trall[2])),
              exp(trall[3]))

```

D.4.2 Likelihoodfunksjon for de univariate rekkene:

```

> t.loglike<-function(param,data){
+   n<-length(data)
+   sigma2<-array(0,n)

```

#Som startverdi for volatiliteten brukes empirisk varians til datasettet:

```
+ sigma2[1]<-var(data)
+ alfa<-exp(param[1])
+ alfa1<-exp(param[2])/(1+exp(param[2])+exp(param[3]))
+ beta<-exp(param[3])/(1+exp(param[2])+exp(param[3]))
+ nu<-exp(param[4])
+
```

#Volatiliteten oppdateres for hvert tidspunkt etter ligning (2.4):

```
+ for(i in 2:n){
+   sigma2[i]<-alfa+alfa1*data[i-1]^2+beta*sigma2[i-1]
+ }
+
```

#Likelihood (4.7)

```
+ logL<-n*log(gamma((nu+1)/2))-n*log(gamma(0.5*nu))
+   -n*0.5*log(pi*(nu-2))-0.5*sum(log(sigma2))
+   -0.5*(nu+1)*sum(log(1+(data^2)/((nu-2)*sigma2)))
```

#Returner negativ likelihoodverdi:

```
+ return(-logL)
+ }
```

D.4.3 Likelihood for Student t- copulaen i d dimensjoner med dynamisk korrelasjon

```
> copula.like<-function(est,data,uniparam,d){
+   T<-dim(data)[1]
+   a<-exp(est[1])/(1+exp(est[1])+exp(est[2]))
+   b<-exp(est[2])/(1+exp(est[1])+exp(est[2]))
+   nuC<-exp(est[3])
+   d<-d
+   std<-rep(0,d)
+   h<-matrix(rep(0,d*T),ncol=d)
+ }
```

#Utvikling av volatiliteten med estimerte parametere fra steg 1:

```
+ for(k in 1:d){
+   std[k]<-sqrt((uniparam[k,4]-2)/uniparam[k,4])
+   h[1,k]<-var(data[,k])
+   for(i in 2:T){
+     h[i,k]<-uniparam[k,1]+uniparam[k,2]*data[i-1,k]^2
+       +uniparam[k,3]*h[i-1,k]
+   }
+ }
```

```
#Estimering av  $\bar{Q}$  og  $\epsilon$ -verdier:
```

```
+ epsilon<-matrix(rep(0,d*T),ncol=d)
+ tQstrek<-0
+ for(i in 1:T){
+   epsilon[i,]<-data[i,]*%solve(diag(sqrt(h[i,])))
+   tQstrek<-tQstrek+epsilon[i,]*%t(epsilon[i,])
+ }
+ Qstrek<-tQstrek/T
```

```
#Transformerer dataene over på copula-skalaen:
```

```
+ u<-x<-matrix(rep(0,d*T),ncol=d)
+ for(j in 1:d){
+   u[,j]<-pst(epsilon[,j],location=0,scale=std[j],
+             shape=0,df=uniparam[j,4])
+   x[,j]<-qt(u[,j],nuC)
+ }
+
+ Q<-Qstrek
+ R<-Qstrek
```

```
#Likelihood (4.6):
```

```
+ logL<-(-0.5*log(det(R))+log(gamma((nuC+d)/2))
+        -((nuC+d)/2)*log(1+(1/nuC)*t(x[1,])%solve(R)%x[1,])
+        +(d-1)*log(gamma(0.5*nuC))-d*log(gamma((nuC+1)/2))
+        +0.5*(nuC+1)*sum(log(1+(x[1,]^2/nuC))))
```

```
#For hver t:
```

```
#Oppdater korrelasjonsmatrisen ved ligning (2.8) og ligning (2.9)
```

```
#Beregn likelihooden
```

```
+ for(i in 2:T){
+   Q<-(1-a-b)*Qstrek+a*(epsilon[i-1,]*%t(epsilon[i-1,]))+b*Q
+   Qdiag=diag(Q)
+   Q_star=diag(sqrt(Qdiag))
+   R<-solve(Q_star)%Q%solve(Q_star)
+ }
```

```
+   logL<- (logL+(-0.5*log(det(R))+log(gamma((nuC+d)/2))
           -(((nuC+d)/2)*log(1+(t(x[i,])%%solve(R))%*%x[i,])/nuC))
           +(d-1)*log(gamma(0.5*nuC))-d*log(gamma((nuC+1)/2))
           +0.5*(nuC+1)*sum(log(1+(x[i,]^2/nuC))))
+ }

#Returner negativ likelihoodverdi:

+   return(-logL)
+ }
>
```

D.5 Simulering fra PCC-DCC-GARCH

Simuleringsalgoritmen for de to PCC- modellene er identisk med den forige bortsett fra punktet der jeg genererer samples fra t-copula med korrelasjonsmatrise R_t og ν_C frihetsgrader. I stedet bruker jeg en kode tilsvarende Algoritme 2 for å generere en trekning fra par-copula-konstruksjonen. Denne er ganske lik, både for modellen med bivariate Student's t-copulaer for alle par og modellen med både Clayton- og Student's t-copulaer, men har noen små variasjoner, som for eksempel ulik H-funksjon, så velger å ta med koden for begge her.

D.5.1 Student t-copula for alle par

```
> rPCC<-function(n,nu,rho){
+ dim=4
+ x<-matrix(rep(0,n*dim),byrow=n,ncol=4)
+ for(i in 1:n){
+   v<-matrix(c(rep(0,dim*dim)),ncol=dim,byrow=T)
+   u<-runif(dim)
+   x[i,1]<-v[1,1]<-u[1]
+   x[i,2]<-v[2,1]<-hinvers(u[2],v[1,1],nu[1],rho[1])
+   v[2,2]<-H(v[1,1],v[2,1],nu[1],rho[1])
+   v[3,1]<-u[3]
+   v[3,1]<-hinvers(v[3,1],v[2,2],nu[4],rho[4])
+   v[3,1]<-hinvers(v[3,1],v[2,1],nu[2],rho[2])
+   x[i,3]<-v[3,1]
+   v[3,2]<-H(v[2,1],v[3,1],nu[2],rho[2])
+   v[3,3]<-H(v[3,1],v[2,1],nu[2],rho[2])
+   v[3,4]<-H(v[2,2],v[3,3],nu[4],rho[4])
+   v[4,1]<-u[4]
+   v[4,1]<-hinvers(v[4,1],v[3,4],nu[6],rho[6])
+   v[4,1]<-hinvers(v[4,1],v[3,2],nu[5],rho[5])
+   x[i,4]<-v[4,1]<-hinvers(v[4,1],v[3,1],nu[3],rho[3])
+ }
+ return(x)
+
+ }
```


Her er $n = 1$, $nu = (v_{12}, v_{23}, v_{34}, v_{13|2}, v_{24|3}, v_{14|23})$ og ρ er en vektor med korrelasjonene for det gitte tidspunktet, der ρ_{12}, ρ_{23} og ρ_{34} er hentet direkte ut fra korrelasjonsmatrisen R , og de betingede korrelasjonene er funnet ved hjelp av formelen for partiell korrelasjon, ligning (6.1).

D.5.2 Clayton- og Student t PCC

```
>rPCC2<-function(n,nu,rho,delta){
+dim=4
+ x<-matrix(rep(0,n*dim),byrow=n,ncol=4)
+ for(i in 1:n){
+   v<-matrix(c(rep(0,dim*dim)),ncol=dim,byrow=T)
+   u<-runif(dim)
+   x[i,1]<-v[1,1]<-u[1]
+   x[i,2]<-v[2,1]<-Hclayinv(u[2],v[1,1],delta[1])
+   v[2,2]<-Hclay(v[1,1],v[2,1],delta[1])
+   v[3,1]<-u[3]
+   v[3,1]<-hinvers(v[3,1],v[2,2],nu[4],rho[4])
+   v[3,1]<-Hclayinv(v[3,1],v[2,1],delta[2])
+   x[i,3]<-v[3,1]
+   v[3,2]<-Hclay(v[2,1],v[3,1],delta[2])
+   v[3,3]<-Hclay(v[3,1],v[2,1],delta[2])
+   v[3,4]<-H(v[2,2],v[3,3],nu[4],rho[4])
+   v[4,1]<-u[4]
+   v[4,1]<-hinvers(v[4,1],v[3,4],nu[6],rho[6])
+   v[4,1]<-hinvers(v[4,1],v[3,2],nu[5],rho[5])
+   x[i,4]<-v[4,1]<-hinvers(v[4,1],v[3,1],nu[3],rho[3])
+ }
+ return(x)
>}
```

D.6 Estimering av parametere i PCC-DCC-GARCH

Også for disse modellene gjør jeg noen transformeringer når jeg optimerer for å sikre at jeg bare optimerer over gyldige GARCH-parametere. Før jeg kan bruke den sekvensielle algoritmen må jeg først transformere dataene mine over på copula-skalaen ved å bruke de univariate parameterene (som jeg finner ved samme optimeringsalgoritme som før).

```

> std<-rep(0,4)
> h<-matrix(rep(0,4*n),ncol=4)
> Fx<-matrix(rep(0,n*4),ncol=4)
> n<-length(x[,1])

#Utvikling av volatilitet med estimerte parametere fra de univariate rekkene:

> for(k in 1:4){
+   std[k]<-sqrt((uniparam[k,4]-2)/uniparam[k,4])
+   h[1,k]<-var(x[,k])
+   for(i in 2:length(x[,1])){
+     h[i,k]<-uniparam[k,1]+uniparam[k,2]*x[i-1,k]^2
+           +uniparam[k,3]*h[i-1,k]
+   }
+ }

#Estimering av  $\bar{Q}$  og  $\epsilon$ -verdier:

> epsilon<-matrix(rep(0,4*length(x[,1])),ncol=4)
> tQstrek<-0
> for(i in 1:length(x[,1])){
+   epsilon[i,]<-x[i,]%*%solve(diag(sqrt(h[i,])))
+   tQstrek<-tQstrek+epsilon[i,]%*%t(epsilon[i,])
+ }
> Qstrek<-tQstrek/length(x[,1])
> R<-Q<-Qstrek

#Transformerer dataene over på copula-skalaen:

> for(j in 1:4){
+   Fx[,j]<-pst(epsilon[,j],location=0,scale=std[j]
+             ,shape=0,df=uniparam[j,4])
+ }

```

Som med simuleringen, er også optimeringsalgoritmene nesten identiske for de to PCC-modellene, bortsett fra noen små forskjeller. Siden forskjellene i selve optimeringsstegene er så små, velger jeg kun å ta med disse for den ene modellen.

D.6.1 Student t-copula for alle par

Sekvensiell algoritme:

#Definerer initialverdier for første nivå og DCC-parametere:

```
> init<-rep(0,5)
> init[1]<-log(0.1)/(1+log(0.1)+log(0.8))
> init[2]<-log(0.8)/(1+log(0.1)+log(0.8))
> init[3]<-init[4]<-init[5]<-4
```

#Optimerer likelihooden for nivå 1:

```
> optimerer<-optim(init,niva2.likeli, data=Fx,K=3,
                  epsilon=epsilon,Qstrek=Qstrek)
> trall<-optimerer$par
> niva2_estimater<-c(exp(trall[1])/(1+exp(trall[1])+exp(trall[2])),
                  exp(trall[2])/(1+exp(trall[1])+exp(trall[2])),
                  trall[3],trall[4],trall[5])
> sekv_nu<-c(rep(0,6))
> sekv_nu[1:3]<-niva2_estimater[3:5]
> a<-niva2_estimater[1]
> b<-niva2_estimater[2]
```

#Beregner observasjoner i niva2 (Bruker Qstrek for korrelasjonene):

```
>u<-Fx
>d<-dim(u)[2]
>n<-dim(u)[1]
>v<-array(rep(0,d*d*n),dim<-c(d,d,n))
>
>for(i in 1:n){
+   if(i>1){
+     Q<-(1-a-b)*Qstrek+a*(epsilon[i-1,]%*%t(epsilon[i-1,]))+b*Q
+     Qdiag=diag(Q)
+     Q_star=diag(sqrt(Qdiag))
+     R<-solve(Q_star)%*%Q%*%solve(Q_star)
+   }
>v[1,1,i]<-H(u[i,1],u[i,2],sekv_nu[1],R[1,2])
```

```
>v[1,2,i]<-H(u[i,3],u[i,2],sekv_nu[2],R[2,3])
>v[1,3,i]<-H(u[i,2],u[i,3],sekv_nu[2],R[2,3])
>v[1,4,i]<-H(u[i,4],u[i,3],sekv_nu[3],R[3,4])
>}

```

#Estimerer parametere i nivå 2:

```
>init<-c(15,0.3)
>data=cbind(t(t(v[1,1,])),t(t(v[1,2,])))
>par<-optim(init,fn=L,data=data)
>sekv_nu[4]<-par$par[1]

```

```
>data=cbind(t(t(v[1,3,])),t(t(v[1,4,])))
>par<-optim(init,fn=L,data=data)
>sekv_nu[5]<-par$par[1]
>rho2<-par$par[2]

```

#Beregner observasjoner for nivå3 :

```
>v[2,1,]<-H(v[1,1,],v[1,2,],sekv_nu[4],rho1)
>v[2,2,]<-H(v[1,4,],v[1,3,],sekv_nu[5],rho2)
>data=cbind(t(t(v[2,1,])),t(t(v[2,2,])))

```

#Estimerer parametere i nivå 3:

```
>par<-optim(init,fn=L,data=data)
>sekv_nu[6]<-par$par[1]

```

Sekvnu er nå min vektor med startverdier for frihetsgradene, mens a og b er startverdier for DCC-parameterene.

Likelihood, nivå 1:

```
> niva2.likeli<-function(init,data,K,epsilon,Qstrek){
+   likelihoood<-0
+   T<-length(data[,1])
+   K<-K
+   a<-exp(init[1])/(1+exp(init[1])+exp(init[2]))
+   b<-exp(init[2])/(1+exp(init[1])+exp(init[2]))
+   nu<-init[3:5]

```

```

+ Fx<-data
+ R<-Q<-Qstrek
+ Rho<-c(R[1,2],R[2,3],R[3,4])
+ u<-Fx
+
+ likelihood<-rep(0,K)
+ likelihoodsum<-0

#Har tidsvarierende korrelasjon i første nivå for å få startverdier også for DCC-parameterene:

+ for(i in 2:T){
+   if(i>1){
+     Q<-(1-a-b)*Qstrek+a*(epsilon[i-1,]%*%t(epsilon[i-1,]))+b*Q
+     Qdiag=diag(Q)
+     Q_star=diag(sqrt(Qdiag))
+     R<-solve(Q_star)%*%Q%*%solve(Q_star)
+     Rho<-c(R[1,2],R[2,3],R[3,4])
+   }
+ for(k in 1:K){
+   likelihood[k]<-log(gamma(0.5*(nu[k]+2)))-log(gamma(0.5*nu[k]))
+     -log(nu[k]*pi)-log(dt(qt(u[i,k],nu[k]),nu[k]))
+     -log(dt(qt(u[i,k+1],nu[k]),nu[k]))-0.5*log(1-R[k]^2)
+     -0.5*(nu[k]+2)*log(1+(qt(u[i,k],nu[k])^2+qt(u[i,k+1],nu[k])^2
+     +2*R[k]*qt(u[i,k],nu[k])*qt(u[i,k+1],nu[k]))/(nu[k]*(1-R[k]^2)))

+
+   likelihoodsum<-likelihoodsum+likelihood[k]
+ }
+ }

#Returnerer negativ likelihoodverdi:

+ return(-likelihoodsum)
+
+}

```

Likelihood, nivå 2 og 3:

```

>L<-function(theta,data){
+ nu<-theta[1]
+ rho<-theta[2]
+ u<-data[,1]
+ v<-data[,2]
+ L<-log(gamma(0.5*(nu+2)))-log(gamma(0.5*nu))-log(nu*pi)
+   -log(dt(qt(u,nu),nu))-log(dt(qt(v,nu),nu))-log(sqrt(1-rho^2))
+   +(-0.5*(nu+2))*log(1+(qt(u,nu)^2
+   +qt(v,nu)^2-2*rho*qt(u,nu)*qt(v,nu))/(nu*(1-rho^2)))
+ likli<-sum(L)
+ return(-likli)
+}

```

Full likelihood

Når jeg har funnet startverdiene for den fulle optimeringen går jeg videre til å optimere den fulle likelihooden som er tilsvarende algoritme 4, bortsett fra at man for hver t må oppdatere Θ .

```

> PCC.DCC.likeli<-function(init,data,Qstrek,epsilon){
+ loglikelihood<-0
+ n<-dim(data)[1]
+ d<-dim(data)[2]
+ a<-exp(init[1])/(1+exp(init[1])+exp(init[2]))
+ b<-exp(init[2])/(1+exp(init[1])+exp(init[2]))
+ nu<-init[3:8]
+ Q<-R<-Qstrek
+ u<-data
+

```

#Lagrer parameterverdiene i Θ -matrisen:

```

+ theta<-array(rep(0,d*d*2),dim<-c(4,4,2))
+ for(i in 1:d-1){
+   theta[1,i,1]<-nu[i]
+ }
+ for (i in 1:(d-2)){

```

```

+   theta[2,i,1]<-nu[i+3]
+ }
+   theta[3,1,1]<-nu[6]
+
#Oppdaterer  $\Theta$  for hvert tidspunkt:
+   for(t in 1:n){
+     if(t==1){
+       R<-Q<-Qstrek
+     }
+     if(t>=2){
+       Q<-(1-a-b)*Qstrek+a*(epsilon[t-1,]%*%t(epsilon[t-1,]))+b*Q
+       Qdiag=diag(Q)
+       Q_star=diag(sqrt(Qdiag))
+       R<-solve(Q_star)%*%Q%*%solve(Q_star)
+     }
+
#Beregner korrelasjonen i nivå 2 og 3 med formelen for delvis korrelasjon:
+   r12<-R[1,2]
+   r23<-R[2,3]
+   r34<-R[3,4]
+   r132<-(R[1,3]-R[1,2]*R[2,3])/(sqrt(1-R[1,2]^2)*sqrt(1-R[2,3]^2))
+   r243<-(R[2,4]-R[3,4]*R[2,3])/(sqrt(1-R[3,4]^2)*sqrt(1-R[2,3]^2))
+   r142<-(R[1,4]-R[1,2]*R[2,4])/(sqrt(1-R[1,2]^2)*sqrt(1-R[2,4]^2))
+   r342<-(R[3,4]-R[3,2]*R[4,2])/(sqrt(1-R[3,2]^2)*sqrt(1-R[4,2]^2))
+   r1423<-(r142-r132*r342)/(sqrt(1-r132^2)*sqrt(1-r342^2))
+
+   for(i in 1:d-1){
+     theta[1,i,2]<-R[i,i+1]
+   }
+   theta[2,1,2]<-r132
+   theta[2,2,2]<-r243
+   theta[3,1,2]<-r1423
+
#Kaller på funksjonen PCClick() som tilsvare algoritme 4 for ett gitt tidspunkt og beregner
likelihoodverdien:

```

```

+   loglikelihood<-loglikelihood+PCClick(theta,u[t,])
+ }
+
#Returnerer negativ likelihoodverdi:
+   return(-loglikelihood)
+ }

```

Denne likelihoodfunksjonen kaller på funksjonen PCClick(), som beregner likelihooden i følge algoritme 4 for hvert tidspunkt t, med den oppdaterte korrelasjonsmatrisen holdt konstant. PCClick() er gjengitt under:

#Likelihood for 4-dimensjonal PCC med Student's t-copula for alle par:

```

> PCClick<-function(theta,data){
+   loglikelihood<-0
+   n<-1
+   d<-length(data)
+   v<-array(rep(0,d*d),dim=c(d,d))
+
+   for(i in 1:d){
+     v[1,i]<-data[i]
+   }
+
+   #likelihoodene i nivå 1:
+   for( i in 1:(d-1)){
+     loglikelihood<-loglikelihood+L(v[1,i],v[1,i+1],theta[1,i,])
+   }
+
+   v[2,1]<-H(v[1,1],v[1,2],theta[1,1,1],theta[1,1,2])
+   v[2,2]<-H(v[1,3],v[1,2],theta[1,2,1],theta[1,2,2])
+   v[2,3]<-H(v[1,2],v[1,3],theta[1,2,1],theta[1,2,2])
+   v[2,4]<-H(v[1,4],v[1,3],theta[1,3,1],theta[1,3,2])
+
+   for(j in 2:3){
+     for(i in 1:(4-j)){

```



```

+     loglikelihood<-loglikelihood+L(v[j, (2*i)-1], v[j, 2*i], theta[j, i,])
+   }
+
+   # if(j==d-1)
+   # break
+   v[3,1]=H(v[2,1], v[2,2], theta[2,1,1], theta[2,1,2])
+   v[3,2]=H(v[2,4], v[2,3], theta[2,2,1], theta[2,2,2])
+ }
+ return(loglikelihood)
+ }

> L<-function(u, v, theta){
+   nu<-theta[1]
+   rho<-theta[2]
+   L<-log(gamma(0.5*(nu+2)))-log(gamma(0.5*nu))
+     -log(nu*pi)-log(dt(qt(u, nu), nu))
+     -log(dt(qt(v, nu), nu))
+     -log(sqrt(1-rho^2))+(-0.5*(nu+2))*log(1+(qt(u, nu)^2
+     +qt(v, nu)^2-2*rho*qt(u, nu)*qt(v, nu))/(nu*(1-rho^2)))
+   sum(L)
+   return(L)
+ }

```

D.6.2 Clayton- og Student t-PCC

Forskjellene i optimeringsalgoritmene ligger for det meste i selve likelihoodfunksjonene (i tillegg til ulike h-funksjoner), så jeg velger å bare ta med koden for disse for denne modellen:

Sekvensiell algoritme, Likelihood nivå 1

```

>niva2clay.likeli<-function(init, data, epsilon, Qstrek){
+ likelihoodsum<-0
+ N<-length(data[,1])
+ a<-exp(init[1])/(1+exp(init[1])+exp(init[2]))
+ b<-exp(init[2])/(1+exp(init[1])+exp(init[2]))
+ nu<-init[3]
+ R<-Q<-Qstrek
+ Rho<-c(R[1,2], R[2,3], R[3,4])

```

```

+ u<-data
+ for(i in 1:N){
+   if(i>1){
+     Q<-(1-a-b)*Qstrek+a*(epsilon[i-1,]%*%t(epsilon[i-1,]))+b*Q
+     Qdiag=diag(Q)
+     Q_star=diag(sqrt(Qdiag))
+     R<-solve(Q_star)%*%Q%*%solve(Q_star)
+     Rho<-c(R[1,2],R[2,3],R[3,4])
+   }
}

#Finner Kendalls tau for de to aktuelle parene:

+   tau1<-2*asin(R[1,2])/pi
+   tau2<-2*asin(R[2,3])/pi

#Finner copulaparameteren:

+   delta1<-2*tau1/(1-tau1)
+   delta2<-2*tau2/(1-tau2)
+   delta<-c(delta1,delta2)
+   delta<-abs(delta)
+   likelihoodsum<-likelihoodsum + Lclay(u[i,1],u[i,2],delta[1])
+                                     +Lclay(u[i,2],u[i,3],delta[2])
+                                     +Lt(u[i,3],u[i,4],c(nu,Rho[3]))
+ }
return(-likelihoodsum)
}

>Lt<-function(u,v,theta){
+   nu<-theta[1]
+   rho<-theta[2]
+   L<-log(gamma(0.5*(nu+2)))-log(gamma(0.5*nu))
+     -log(nu*pi)-log(dt(qt(u,nu),nu))
+     -log(dt(qt(v,nu),nu))-log(sqrt(1-rho^2))
+     +(-0.5*(nu+2))*log(1+(qt(u,nu)^2+qt(v,nu)^2
+     -2*rho*qt(u,nu)*qt(v,nu))/(nu*(1-rho^2)))
+   sum(L)
}

```

```

+ return(L)
>}

>Lclay<-function(u,v,delta){
+ L<-log(1+delta)
+   -(delta+1)*log(u*v)
+   -(2+1/delta)*log(u^(-delta)+v^(-delta)-1)
+ sum(L)
+ return(L)
>}

```

Full likelihood

```

>clayPCCDCCLikeli<-function(init,data,Qstrek,epsilon){
+ loglikelihood<-0
+ n<-dim(data)[1]
+ d<-dim(data)[2]
+ a<-exp(init[1])/(1+exp(init[1])+exp(init[2]))
+ b<-exp(init[2])/(1+exp(init[1])+exp(init[2]))
+ nu<-rep(0,6)
+ nu[3:6]<-init[3:6]
+ Q<-R<-Qstrek
+ u<-data

#Lagrer parameterverdier i  $\Theta$ -matrisen:

+ theta<-array(rep(0,d*d*2),dim<-c(4,4,2))
+ for(i in 1:d-1){
+   theta[1,i,1]<-nu[i]
+ }
+ for (i in 1:(d-2)){
+   theta[2,i,1]<-nu[i+3]
+ }
+ theta[3,1,1]<-nu[6]
+ tau1<-2*asin(R[1,2])/pi

```

```

+ tau2<-2*asin(R[2,3])/pi
+
+ delta1<-2*tau1/(1-tau1)
+ delta2<-2*tau2/(1-tau2)
+ delta<-c(delta1,delta2)

#Oppdaterer  $\Theta$  for hvert tidspunkt og beregner likelihooden:

+ for(t in 1:n){
+   if(t==1){
+     R<-Q<-Qstrek
+   }
+   if(t>=2){
+     Q<-(1-a-b)*Qstrek+a*(epsilon[t-1,]**t(epsilon[t-1,]))+b*Q
+     Qdiag=diag(Q)
+     Q_star=diag(sqrt(Qdiag))
+     R<-solve(Q_star)**Q**solve(Q_star)
+   }
+   r12<-R[1,2]
+   r23<-R[2,3]
+   r34<-R[3,4]
+   r132<-(R[1,3]-R[1,2]*R[2,3])/(sqrt(1-R[1,2]^2)*sqrt(1-R[2,3]^2))
+   r243<-(R[2,4]-R[3,4]*R[2,3])/(sqrt(1-R[3,4]^2)*sqrt(1-R[2,3]^2))
+   r142<-(R[1,4]-R[1,2]*R[2,4])/(sqrt(1-R[1,2]^2)*sqrt(1-R[2,4]^2))
+   r342<-(R[3,4]-R[3,2]*R[4,2])/(sqrt(1-R[3,2]^2)*sqrt(1-R[4,2]^2))
+   r1423<-(r142-r132*r342)/(sqrt(1-r132^2)*sqrt(1-r342^2))
+
+   for(i in 1:d-1){
+     theta[1,i,2]<-R[i,i+1]
+   }
+   theta[2,1,2]<-r132
+   theta[2,2,2]<-r243
+   theta[3,1,2]<-r1423
+
#Finner Kendalls tau for de to aktuelle parene:

+ tau1<-2*asin(R[1,2])/pi
+ tau2<-2*asin(R[2,3])/pi

```

#Finner copulaparameteren, δ :

```
+ delta1<-2*tau1/(1-tau1)
+ delta2<-2*tau2/(1-tau2)
+ delta<-c(delta1,delta2)
+
+ loglikelihood<-loglikelihood+clayPCClick(theta=theta,data=u[t,],delta=delta)
+ }
+
+ return(-loglikelihood)
>}
```

Denne likelihoodfunksjonen kaller på funksjonen `clayPCClick()`, som beregner likelihooden i følge algoritme 4 for hvert tidspunkt t , med den oppdaterte korrelasjonsmatrisen holdt konstant. `clayPCClick()` er gjengitt under:

#Likelihood for 4-dimensjonal PCC med Claytons- og Student's t-copula:

```
>clayPCClick<-function(theta,data,delta){
+ loglikelihood<-0
+ n<-1
+ d<-length(data)
+ v<-array(rep(0,d*d),dim<-c(d,d))
+
+ for(i in 1:d){
+   v[1,i]<-data[i]
+ }
+
+ #likelihoodene i nivå 1:
+
+ loglikelihood<-loglikelihood+
+           Lclay(v[1,1],v[1,2],delta[1])+
+           Lclay(v[1,2],v[1,3],delta[2])+
+           L(v[1,3],v[1,4],theta[1,3,])
+
+ }
```

```

+ v[2,1]<-Hclay(v[1,1],v[1,2],delta[1])
+ v[2,2]<-Hclay(v[1,3],v[1,2],delta[2])
+ v[2,3]<-Hclay(v[1,2],v[1,3],delta[2])
+ v[2,4]<-H(v[1,4],v[1,3],theta[1,3,1],theta[1,3,2])
+
+ for(j in 2:3){
+   for(i in 1:(4-j)){
+     loglikelihood<-loglikelihood+L(v[j,(2*i)-1],v[j,2*i],theta[j,i,])
+   }
+
+   # if(j==d-1)
+   # break
+   v[3,1]=H(v[2,1],v[2,2],theta[2,1,1],theta[2,1,2])
+   v[3,2]=H(v[2,4],v[2,3],theta[2,2,1],theta[2,2,2])
+ }
+ return(loglikelihood)
>}

```

D.7 VaR-evalueringer

Koden for Value at Risk evalueringene er implementert etter algoritmen i avsnitt 8.4.2. Algoritmen er den samme for alle de tre modellene, så jeg velger å bare ta med en av de:

```

> data<-read.table("sektorindekser.csv",sep=";")
> indekser<-data
> mat<-indekser[,3]
> indu<-indekser[,4]
> consdis<-indekser[,5]
> inftec<-indekser[,9]
> data<-cbind(mat,indu,consdis,inftec)
> indeks<-cbind(diff(log(data[,1])),
                diff(log(data[,2])),
                diff(log(data[,3])),
                diff(log(data[,4])))
> aksjer<-indeks
> aksjer2<-cbind(aksjer[,2],aksjer[,4],aksjer[,1],aksjer[,3])

```

#Deler opp datasettet i et testsett og et estimeringssett:

```
> estimeringssett<-aksjer2[2981:(length(aksjer2[,1])-121),]
> testsett<-aksjer2[3981:length(aksjer2[,1]),]
> portefoljesum<-(testsett[,1]+testsett[,2]+testsett[,3]+testsett[,4])/4
```

#Bruker estimeringssettet til å estimere parametere for prediksjon av volatilitet og korrelasjon ved neste tidspunkt:

```
> varhel<-varhel2<-varhel3<-rep(0,220)
```

```
>for(i in 1:220){
+ bit<-varPCCDCOpt(estimeringssett)
+ uniparam<-bit$uni
+ copest<-bit$copest
+ a<-copest[1]
+ b<-copest[2]
+ nuvec<-copest[3:8]
+ ht<-bit$ht
+ qstrek<-bit$qstrek
+ Qstrek<-qstrek
+ epsilon<-bit$epsilon
+
+ }
```

#Beregner volatilitet og korrelasjon ved neste tidspunkt:

```
+ for(t in 1:(length(epsilon[,1])+1)){
+   if(t==1){
+     R<-Q<-Qstrek
+   }
+   if(t>=2){
+     Q<-(1-a-b)*Qstrek+a*(epsilon[t-1,]%*%t(epsilon[t-1,]))+b*Q
+     Qdiag=diag(Q)
+     Q_star=diag(sqrt(Qdiag))
+     R<-solve(Q_star)%*%Q)%*%solve(Q_star)
+   }
+ }
+ }
```

```

+ for(k in 1:4){
+   h[k]<-uniparam[k,1]
+     +uniparam[k,2]*estimeringssett[length(estimeringssett[,1]),k]^2
+     +uniparam[k,3]*ht[k]
+ }
+
+ vektor<-rep(0,10000)
+
#Beregner Value at Risk: Simulerer 10000 trekninger for neste dag og finner de ønskede
kvantilene:

+ n=10000
+ for (k in 1:n){
+   vec<-varPCCDCCsim(1,a,b,nuvec,uniparam,R,h)
+   vektor[k]<-(vec[1]+vec[2]+vec[3]+vec[4])/4
+ }
+ varhel1[i]<-quantile(vektor,0.05)
+ varhel2[i]<-quantile(vektor,0.01)
+ varhel3[i]<-quantile(vektor,0.005)
+
#Flytter estimeringssettet en dag frem og gjentar prosessen til jeg er kommet til enden av
testsettet:

+ estimeringssett<-aksjer2[(3000+i):(length(aksjer2[,1])-121+i),]
+}
>

```