

Portfolio-VaR-Schätzung mittels bedingter Copula-GARCH-Methode



Quelle: http://images.finanzen.net/mediacenter/unsortiert/1_aktien_charts_ramcreations_shutterstock_660.jpg

Bachelorarbeit zur Erlangung des
akademischen Grades eines Bachelor of Science

Magdalena Leitner

Betreuer/in: Univ.-Prof. Mag. Dr. Werner G. Müller

LVA-Leiter/in: Assoz. Univ.-Prof.ⁱⁿ Mag.^a Dr.ⁱⁿ Helga Wagner

Juli 2016

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, Juli 2016

Magdalena Leitner

Abstract

Value-at-Risk (VaR) is an important instrument for risk assessment and is commonly used in finance. The estimation can be carried out in several ways. One possibility to model VaR is to use GARCH models and copula functions which have gained in importance in the last decades and can be used for modeling dependence structures of random variables. The conditional Copula-GARCH-method which is applied in this bachelor thesis is a quite flexible approach and can be used for a broad range of applications. Estimating VaR for a portfolio, composed of ATX and DAX, shows that the Survival-Clayton-Copula-GARCH-t-method seems to be an adequate model in this context. In case of inaccuracies concerning the VaR estimation tendencies to underestimate the number of violations do not involve such grave consequences for portfolio holders as overestimation would. Therefore it is recommended to use GARCH-t-margins as they rather underestimate the violation number in contrast to GARCH-n-margins.

Kurzzusammenfassung

Value-at-Risk (VaR) ist ein bedeutendes Instrument zur Risikoabschätzung bei Finanzzeitreihen. Seine Schätzung kann auf verschiedene Arten erfolgen. Eine Möglichkeit ist die Modellierung mit Hilfe von GARCH-Modellen und Copula-Funktionen, die in den letzten Jahrzehnten massiv an Bedeutung gewonnen haben und ein Tool sind, mit denen Abhängigkeitsstrukturen zwischen Zufallsvariablen modelliert werden können. Bei der bedingten Copula-GARCH-Methode, die in dieser Bachelorarbeit verwendet wird, handelt es sich um einen äußerst flexiblen Ansatz, der sehr breit eingesetzt werden kann. In Anwendung auf ein Portfolio, bestehend aus den beiden Aktienindizes ATX und DAX, scheint die Survival-Clayton-GARCH-t-Methode ein durchaus angemessenes Modell zu sein. In Anbetracht dessen, dass bei Ungenauigkeiten in der VaR-Schätzung eine tendenzielle Unterschätzung der Verletzungsanzahl nicht so gravierende Folgen nach sich zieht wie eine Überschätzung, kann zudem eine Empfehlung für GARCH-t-Randverteilungen ausgesprochen werden, da diese im Gegensatz zu GARCH-n-Randverteilungen eben diese Eigenschaft mit sich bringen.

Inhaltsverzeichnis

Abstract	2
1 Einleitung	7
1.1 Problemstellung und Forschungsstand	7
1.2 Gliederung der Arbeit	7
2 Theoretische Herangehensweise	9
2.1 Definition Rendite	9
2.2 Definition VaR	9
2.3 Modellierung der Randverteilungen	10
2.4 Copula-Theorie	10
2.5 VaR-Schätzung	13
3 Empirische Ergebnisse	15
3.1 Beschreibung der Daten	15
3.2 Schätzung der Randverteilungen	17
3.3 Copula-Modellierung	18
3.4 VaR-Schätzung	19
4 Vergleich mit herkömmlichen VaR-Schätzmethoden	23
5 Fazit	26
Anhang	27
R-Code	27
Verwendete R-packages	49
Literatur	51

Tabellenverzeichnis

3.1	Deskriptive Kennzahlen und Engle-Tests	16
3.2	Ergebnisse der univariaten GARCH(1,1)-n-Modelle	17
3.3	Ergebnisse der univariaten GARCH(1,1)-t-Modelle	18
3.4	Copula-Selektion - Kendall's τ und Selektionskriterien	19
3.5	Anzahl der Verletzungen bei der VaR-Schätzung (erwartet: 61; $\alpha=0.05$) . .	20
3.6	Anzahl der Verletzungen bei der VaR-Schätzung (erwartet: 12; $\alpha=0.01$) . .	21
3.7	Anzahl der Verletzungen bei der VaR-Schätzung	22
4.1	Vergleich der VaR-Schätzmethoden	25

Abbildungsverzeichnis

3.1	Entwicklung des ATX und des DAX	15
3.2	Renditen des ATX und des DAX (feiertagsbereinigt)	16
3.3	VaR-Schätzung mittels Survival-Clayton-Copula-GARCH-t-Modell	20
4.1	Vergleich der VaR-Schätzmethoden	25

1. Einleitung

1.1 Problemstellung und Forschungsstand

Value-at-Risk (VaR) ist ein wichtiges Risikomaß für Finanzinstitute, die dieses Analyseinstrument beispielsweise zur Risikoabschätzung von Portfolios verwenden, und kann auf verschiedene Arten modelliert werden (Jorion, 2001). Die meisten dieser herkömmlichen Schätzverfahren weisen allerdings gewisse Nachteile bzw. Restriktionen (z.B. Normalverteilungsannahme, Annahme konstanter Volatilität) auf (Fricke, 2006). Copula-Funktionen bieten in Verbindung mit GARCH-Modellen die Möglichkeit einer flexibleren Modellierung, wodurch die Qualität der Schätzung des VaR verbessert werden soll, was für die Auseinandersetzung mit diesem Thema spricht.

Zu den ersten, die GARCH-Modelle und bedingte Copula-Funktionen gemeinsam zur Modellierung von Finanzzeitreihen eingesetzt haben, dürften Rockinger und Jondeau (2001) gehören. Arbeiten, die sich mit dem Einsatz dieser Methode in der VaR-Schätzung befassen, wurden zum Beispiel von Palaro und Hotta (2006); Hotta, Lucas und Palaro (2008); Jondeau und Rockinger (2006); Ozun, Cifter et al. (2007) und Lu, Lai und Liang (2014) verfasst. Auch Huang, Lee, Liang und Lin (2009) haben sich mit dieser Thematik auseinandergesetzt, wobei in ihrer Arbeit mehrere Copula-Funktionen mit unterschiedlichen Randverteilungen (GARCH- und GJR-Modelle) angewendet wurden, um die Robustheit der Methode hervorzuheben.

Diese Bachelorarbeit beschäftigt sich mit der VaR-Modellierung für ein Aktienportfolio mittels der bedingten Copula-GARCH-Methode, wobei jeweils 1-Schritt-Vorhersagen gemacht werden. Dabei wird sich die Autorin der Arbeit vor allem am Paper von Huang et al. (2009) orientieren. Zusätzlich werden weitere Copula-Funktionen verwendet, die in der Arbeit von Huang et al. (2009) nicht eingesetzt wurden. Angewendet wird der Ansatz schließlich auf die beiden Aktienindizes DAX und ATX.

1.2 Gliederung der Arbeit

In Kapitel 2 werden die theoretischen Grundlagen des Ansatzes verdeutlicht, wobei zuerst einige wichtige Begriffe definiert (Kapitel 2.1 und 2.2), weiters auf die Randverteilungsmodellierung (Kapitel 2.3) eingegangen, dann die grundlegenden Bausteine der Copula-Theorie erläutert (Kapitel 2.4) und schließlich der Schätzprozess des VaR erklärt wird (Kapitel 2.5). Kapitel 3 beschäftigt sich mit der Präsentation der empirischen Ergebnisse. Dazu werden als erstes die verwendeten Daten charakterisiert (Kapitel 3.1) sowie im weiteren Verlauf die Ergebnisse der Schätzungen der univariaten Verteilungen (Kapitel 3.2) und die Copula-Selektion (Kapitel 3.3) dargestellt. Weiters werden die Ergebnisse der Copula-GARCH-VaR-Schätzung diskutiert (Kapitel 3.4), welche im Anschluss mit herkömmlichen VaR-Schätzmethoden, wie der historischen Simulation oder der Varianz-Kovarianz-Methode, verglichen (Kapitel 4) werden. In Kapitel 5 wird Resümee über die Arbeit gezogen.

2. Theoretische Herangehensweise

2.1 Definition Rendite

Aufgrund der besseren statistischen Eigenschaften werden grundsätzlich nicht die Aktienindizes selbst zur Modellierung herangezogen, sondern deren Renditen x_t , die wie folgt definiert sind (siehe Huang et al., 2009, S. 319):

$$x_t = 100 \cdot \ln \left(\frac{P_t}{P_{t-1}} \right), \quad (2.1)$$

wobei P_t der Wert des Aktienindizes zum Zeitpunkt t ist.

2.2 Definition VaR

Für BesitzerInnen von Aktienportfolios dürfte unter anderem die Frage interessant sein, wie viel sie mit ihrem Portfolio innerhalb einer bestimmten Zeitspanne mit einer gewissen Wahrscheinlichkeit maximal verlieren können. Dies führt zum Konzept des VaR, welcher angibt, welche Verlusthöhe in einer gewissen Periode mit einer festgelegten Sicherheit nicht überschritten wird (Wolke, 2008, S.31). Weist der VaR für den nächsten Tag bei $\alpha = 0.05$ beispielsweise einen Wert von -2 auf, so bedeutet das, dass man mit besagtem Portfolio mit einer Sicherheit von 95 Prozent am nächsten Tag nicht mehr als 2 Prozent verlieren wird (Wolke, 2008, S. 33). Mathematisch kann dies folgendermaßen angeschrieben werden (siehe Huang et al., 2009, S. 318):

$$\text{VaR}_t(\alpha) = \inf\{s : F_t(s) \geq \alpha\}, \quad (2.2)$$

wobei F_t die Verteilungsfunktion der Portfoliorendite $X_{p,t}$ zum Zeitpunkt t ist.

Zudem gilt:

$$P(X_{p,t} \leq \text{VaR}_t(\alpha) | \Omega_{t-1}) = \alpha, \quad (2.3)$$

wobei Ω_{t-1} die Information zum Zeitpunkt $t - 1$ darstellt.

Wie sich also erkennen lässt, ist für die VaR-Berechnung die Verteilungsfunktion der

Portfoliorendite von Bedeutung. Um die gemeinsame Verteilung der Renditen schätzen zu können, wird daher im Verlauf der Arbeit auf den Copula-GARCH-Ansatz zurückgegriffen.

2.3 Modellierung der Randverteilungen

Bei der Copula-GARCH-Methode handelt es sich um ein äußerst flexibles Tool, da hier die Randverteilungen und Abhängigkeiten getrennt voneinander modelliert werden. Der Schätzvorgang der gemeinsamen Verteilungsfunktion wird also in zwei Phasen aufgeteilt (siehe z.B. Huang et al., 2009). Für die Schätzung der Randverteilungen werden zunächst GARCH-Modelle (Bollerslev, 1986) herangezogen. Diese werden grundsätzlich bei Finanzdaten sehr häufig eingesetzt, da sie sich gut zur Volatilitätsmodellierung eignen. In diesem Fall scheint ein GARCH(1,1)-Modell angemessen. Die Modellgleichung kann wie folgt angeschrieben werden:

$$x_t = \mu + \sigma_t \epsilon_t, \quad (2.4)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 (\sigma_{t-1} \epsilon_{t-1})^2 + \gamma \sigma_{t-1}^2, \quad (2.5)$$

wobei $\epsilon_t \sim N(0, 1)$ oder $\epsilon_t \sim t_d$.

Durch den mit dem GARCH(1,1)-Modell geschätzten Mittelwert bzw. die bedingten Varianzen können die Renditen mit Hilfe der entsprechenden Verteilungsfunktion transformiert werden, um sie so für den Copula-Schätzprozess heranziehen zu können (Huang et al., 2009, S. 317).

2.4 Copula-Theorie

Nun wird der Begriff der Copula definiert. Zur Vereinfachung beschränkt sich die Definition (Nelsen, 2006, S. 10) auf den bivariaten Fall. Sei $\mathbb{I} = [0, 1]$. Eine zweidimensionale Copula ist eine bivariate Funktion $C : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$ mit den folgenden Eigenschaften:

- für jedes $u, v \in \mathbb{I}$

$$C(u, 0) = 0, C(u, 1) = u, C(0, v) = 0, C(1, v) = v; \quad (2.6)$$

- für jedes $u_1, v_1, u_2, v_2 \in \mathbb{I}$, so dass $u_1 \leq u_2$ und $v_1 \leq v_2$

$$C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0. \quad (2.7)$$

Copulas dienen zur Modellierung von Interdependenzen zwischen Zufallsvariablen (Joe, 1997, S. 12), in diesem Fall wird also mit Hilfe der Copula die Abhängigkeit zwischen den

beiden mit den Randverteilungen transformierten Aktienrenditen angepasst. Die Verbindung zwischen der Copulafunktion und der gemeinsamen Verteilungsfunktion von Zufallsvariablen wird über Sklar's Theorem (Sklar, 1959) hergestellt: Sei F_{X_1, X_2} eine gemeinsame Verteilungsfunktion mit stetigen Randverteilungen F_{X_1} und F_{X_2} , dann existiert eine eindeutige zweidimensionale Copula, so dass

$$F_{X_1, X_2} = C(F_{X_1}(x_1), F_{X_2}(x_2)). \quad (2.8)$$

Da multivariate Verteilungsfunktionen in der Regel schwierig zu schätzen sind, kann durch die Verwendung der Copulafunktion und somit durch die Trennung des Schätzvorgangs der Modellierungsprozess erleichtert werden.

In dieser Arbeit werden die Gaussian Copula, die Student t-Copula, die Clayton Copula, die Gumbel Copula, die Frank Copula, die BB1 Copula, die BB6 Copula, die BB7 Copula, die Survival Clayton Copula und die Survival Gumbel Copula herangezogen. Nachfolgend werden diese Copulafamilien genauer beschrieben bzw. ihre Verteilungsfunktionen vorgestellt:

(a) Gaussian Copula (Joe, 1997, S. 140-141)

Die Verteilungsfunktion der Gaussian Copula wird wie folgt definiert:

$$C_{Gaussian}(u, v; \delta) = \Phi_{\delta}(\Phi^{-1}(u), \Phi^{-1}(v)), \quad (2.9)$$

wobei $\delta \in [0, 1]$, Φ die Verteilungsfunktion der Standardnormalverteilung darstellt und Φ_{δ} die bivariate Standardnormalverteilungsfunktion mit Korrelationsparameter δ . Die Gaussian Copula ist spiegelsymmetrisch.

(b) Student t-Copula (Demarta & McNeil, 2005, S. 2)

Die Verteilungsfunktion der Student t-Copula ähnelt derer der Gaussian Copula, nur dass anstelle der Standardnormalverteilungen t-Verteilungen mit d Freiheitsgraden verwendet werden:

$$C_T(u, v; \delta, d) = t_{d, \delta}(t_d^{-1}(u), t_d^{-1}(v)). \quad (2.10)$$

Bei der Verwendung von großen Werten für d kann die Abhängigkeit in Randbereichen verringert, bei kleinen erhöht werden. Große Werte für d bedeuten daher eine Annäherung an die Gaussian Copula.

(c) Clayton Copula (Joe, 1997, S. 141)

Die Verteilungsfunktion der Clayton Copula wird folgendermaßen angeschrieben:

$$C_{Clayton}(u, v; \delta) = (u^{-\delta} + v^{-\delta} - 1)^{-\frac{1}{\delta}}, \quad (2.11)$$

wobei $\delta \in [0, \infty)$. Mit dieser Copula können Randabhängigkeiten im unteren Bereich modelliert werden, weshalb sie in Hinblick auf das Konzept des VaR interessant sein könnte.

(d) Gumbel Copula (Joe, 1997, S. 142)

Im Gegensatz zur Clayton Copula ist die Gumbel Copula gut zur Modellierung von Abhängigkeiten in oberen Randbereichen geeignet. Die Verteilungsfunktion wird so definiert:

$$C_{Gumbel}(u, v; \delta) = \exp\{ -((-\log u)^\delta + (-\log v)^\delta)^{\frac{1}{\delta}} \}, \quad (2.12)$$

wobei $\delta \in [1, \infty)$.

(e) Frank Copula (Joe, 1997, S. 141)

Die Definition der Frank Copula lautet wie folgt:

$$C_{Frank}(u, v; \delta) = -\delta^{-1} \log \left(\frac{\eta - (1 - e^{-\delta u})(1 - e^{-\delta v})}{\eta} \right), \quad (2.13)$$

wobei $\delta \in [0, \infty)$ und $\eta = 1 - e^{-\delta}$. Genau wie die Gaussian Copula ist die Frank Copula spiegelsymmetrisch.

(f) BB1 Copula (Joe, 1997, S. 150)

Die BB1 Copula stellt eine Kombination aus Clayton Copula und Gumbel Copula dar. Die Verteilungsfunktion wird so definiert, wobei $\theta \in (0, \infty)$ und $\delta \in [1, \infty)$:

$$C_{BB1}(u, v; \theta, \delta) = \{1 + [(u^{-\theta} - 1)^\delta + (v^{-\theta} - 1)^\delta]^{\frac{1}{\delta}}\}^{-\frac{1}{\theta}}. \quad (2.14)$$

(g) BB6 Copula (Joe, 1997, S. 142 u. 152)

Die BB6 Copula ist eine Verbindung aus Joe Copula und Gumbel Copula. Die Joe Copula weist wie die Gumbel Copula verstärkte Abhängigkeiten in oberen Randbereichen auf. Definiert wird die BB6 Copula folgendermaßen:

$$C_{BB6}(u, v; \theta, \delta) = 1 - (1 - \exp\{ -[(-\log(1 - (1 - u)^\theta))^\delta + (-\log(1 - (1 - v)^\theta))^\delta]^{\frac{1}{\delta}} \})^{\frac{1}{\theta}}, \quad (2.15)$$

wobei $\theta \in [1, \infty)$ und $\delta \in [1, \infty)$.

(h) BB7 Copula (Joe, 1997, S. 153)

Die BB7 Copula verbindet die Joe Copula mit der Clayton Copula. Die Verteilungsfunktion wird wie folgt angeschrieben:

$$C_{BB7}(u, v; \theta, \delta) = 1 - (1 - [(1 - (1 - u)^\theta)^{-\delta} + (1 - (1 - v)^\theta)^{-\delta} - 1]^{-\frac{1}{\delta}})^{\frac{1}{\theta}}, \quad (2.16)$$

wobei $\theta \in [1, \infty)$ und $\delta \in (0, \infty)$.

(i) Survival Clayton Copula (Joe, 1997, S. 140 u. 141)

Für die Survival Clayton Copula wird die Clayton Copula um 180° gedreht. Somit wird die Verteilungsfunktion folgendermaßen definiert:

$$C_{Rotated-Clayton}(u, v; \delta) = u + v - 1 + C_{Clayton}(1 - u, 1 - v; \delta), \quad (2.17)$$

wobei $\delta \in [0, \infty)$.

(j) Survival Gumbel Copula (Joe, 1997, S. 140 u. 142)

Dasselbe gilt für die Survival Gumbel Copula. Auch hier wird die Gumbel Copula entsprechend gedreht. Die Verteilungsfunktion sieht daher wie folgt aus:

$$C_{Rotated-Gumbel}(u, v; \delta) = u + v - 1 + C_{Gumbel}(1 - u, 1 - v; \delta), \quad (2.18)$$

wobei $\delta \in [1, \infty)$.

2.5 VaR-Schätzung

Die Portfoliorendite, also die gemeinsame Rendite der beiden Aktienindizes kann approximativ folgendermaßen angeschrieben werden (Huang et al., 2009, S. 319):

$$X_{p,t} = \omega X_{1,t} + (1 - \omega) X_{2,t}, \quad (2.19)$$

wobei ω und $(1 - \omega)$ die Portfoliogewichte der beiden Indizes sind. Der Einfachheit halber wird davon ausgegangen, dass die beiden Aktien im Portfolio gleich gewichtet sind, wobei dies natürlich beliebig verallgemeinert werden kann. Somit können die Gewichte jeweils mit $\frac{1}{2}$ ersetzt werden:

$$X_{p,t} = \frac{1}{2} X_{1,t} + \frac{1}{2} X_{2,t}. \quad (2.20)$$

Durch Einsetzen der approximierten Portfoliorendite aus (2.20) in (2.3) erhält man - abweichend von Huang et al. (2009), die hier vermutlich einen Tippfehler gemacht haben:

$$P\left(\frac{1}{2}X_{1,t} + \frac{1}{2}X_{2,t} \leq VaR_t(\alpha)|\Omega_{t-1}\right) \quad (2.21)$$

$$= P(X_{1,t} \leq 2VaR_t(\alpha) - X_{2,t}|\Omega_{t-1}) = \alpha. \quad (2.22)$$

Um nun das entsprechende Quantil der gemeinsamen Verteilung bei $\alpha = 0.05$ zu erhalten, wird die gemeinsame Dichte der beiden Renditen wie folgt integriert (siehe Huang et al., 2009, S. 319):

$$P(X_{p,t} \leq VaR_t(0.05)|\Omega_{t-1}) \quad (2.23)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{2VaR_t(0.05)-x_{2,t}} f(x_{1,t}, x_{2,t}|\Omega_{t-1}) dx_{1,t} dx_{2,t} \quad (2.24)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{2VaR_t(0.05)-x_{2,t}} c(F(x_{1,t}), F(x_{2,t})|\Omega_{t-1}) f(x_{1,t}|\Omega_{t-1}) f(x_{2,t}|\Omega_{t-1}) dx_{1,t} dx_{2,t} = 0.05. \quad (2.25)$$

Dieses Integral kann nun numerisch gelöst und durch das Nullsetzen der umgeformten Gleichung eine Schätzung für VaR bestimmt werden.

Da der Ansatz nun in der Theorie vorgestellt wurde, wird er jetzt in Kapitel 3 zur Anwendung gebracht.

3. Empirische Ergebnisse

3.1 Beschreibung der Daten

Um die oben vorgestellte Methode empirisch zu testen, wurde der Ansatz auf Daten der Aktienindizes ATX und DAX im Zeitraum vom 01.01.2007 bis 31.12.2015 mit insgesamt 2228 täglichen Beobachtungen angewendet. Die Daten stammen aus der Datastream-Datenbank der Abteilung Asset Management der JKU Linz. Um künstliche Autokorrelation zu vermeiden, wurden alle Tage, an denen in mindestens einem der beiden Länder, also Österreich bzw. Deutschland, nicht an der Börse gehandelt wurde, eliminiert. In Abbildung 3.1 ist die Entwicklung des ATX und des DAX im oben genannten Zeitraum ersichtlich.

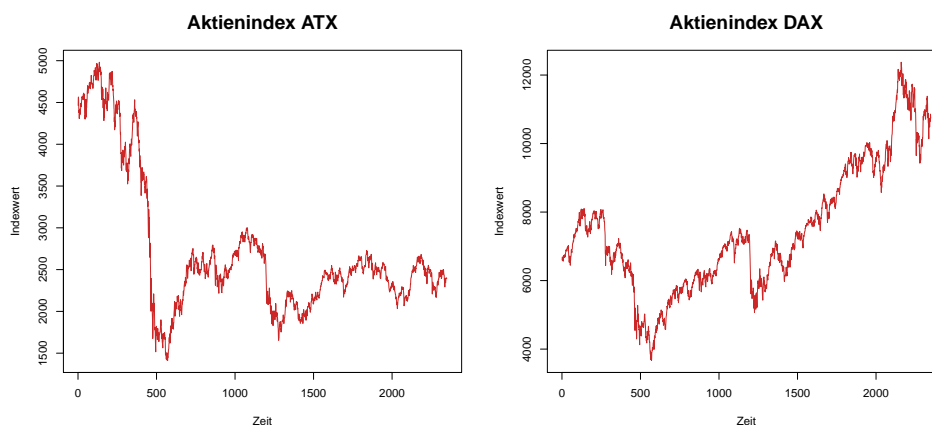


Abbildung 3.1: Entwicklung des ATX und des DAX

Abbildung 3.2 zeigt die Renditen der beiden Aktienindizes, wobei hier Volatilitätscluster gut erkennbar sind, was auf die Angemessenheit eines GARCH-Modells zur Randverteilungsmodellierung hindeutet. Tabelle 3.1 liefert einen Überblick über einige deskriptive Kennzahlen der Aktienrenditen sowie Teststatistiken zu ARCH-Effekten. Wie sich gut erkennen lässt, weisen beide Renditen einen Mittelwert in der Nähe von 0 auf. Zudem sind die Renditen des ATX linksschief und die Verteilungen beider Aktienrenditen schmaler gewölbt als eine Normalverteilung. Die Engle-Teststatistiken lassen darauf schließen, dass die Renditen ARCH-Effekte (Engle, 1982) aufweisen.

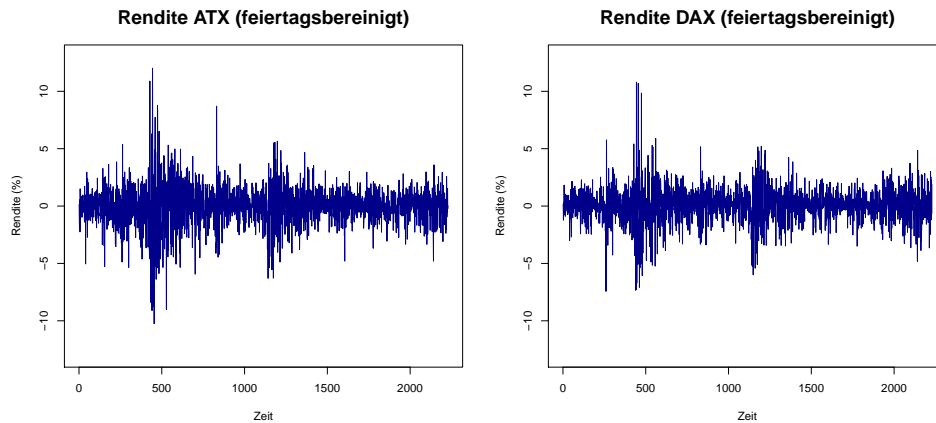


Abbildung 3.2: Renditen des ATX und des DAX (feiertagsbereinigt)

Statistik	ATX		DAX	
N	2228		2228	
Mittelwert	-0.029		0.023	
Standardabweichung	1.753		1.486	
Schiefe	-0.168		0.01	
Wölbung	5.001		5.552	
Engle-Test	Q-Statistik	p-Wert	Q-Statistik	p-Wert
LM(4)	227.23	0.000	45.77	0.000
LM(6)	262.31	0.000	75.27	0.000
LM(8)	282.14	0.000	87.86	0.000
LM(10)	461.98	0.000	103.96	0.000

Tabelle 3.1: Deskriptive Kennzahlen und Engle-Tests

Um das Abschneiden der bedingten Copula-GARCH-Methode bewerten und sie mit anderen Methoden vergleichen zu können, werden die Daten in Test- und Trainingsdaten aufgeteilt. Von den 2228 täglichen Beobachtungen werden die ersten 1000 Beobachtungen verwendet, um den VaR für den 1001. Zeitpunkt zu schätzen. Dann werden die Beobachtungen 2 bis 1001 zur Schätzung des VaR für den 1002. Zeitpunkt herangezogen. Dieser Vorgang wird entsprechend oft wiederholt. Am Ende ergeben sich somit 1228 VaR-Schätzungen, die schließlich mit den tatsächlichen Portfoliorenditen zu diesen Zeitpunkten abgeglichen werden können. Aufgrund dieser Einteilung in Test- und Trainingsdaten beziehen sich die Ergebnisse der univariaten Randverteilungen sowie die der Copula-Schätzungen auf die ersten 1000 Beobachtungen und stellen nur einen Ausschnitt dar.

3.2 Schätzung der Randverteilungen

Tabelle 3.2 zeigt die Ergebnisse der Randverteilungsmodellierung für die ersten 1000 Beobachtungen mittels GARCH-n-Modell sowie Teststatistiken von Ljung-Box- und Engle-Tests (Ljung & Box, 1978; Engle, 1982). Wie sich hier erkennen lässt, kann die Nullhypothese, die besagt, dass keine Autokorrelation vorhanden ist, bei einem Signifikanzniveau von 5 Prozent nicht abgelehnt werden. Für die quadrierten Residuen des ATX können zudem keine ARCH-Effekte mehr nachgewiesen werden. Beim DAX lassen sich die ARCH-Effekte nicht ganz eliminieren. Da durch Veränderung der GARCH-Ordnungen jedoch keine Verbesserungen erzielt werden konnten, wurde trotz verbleibenden ARCH-Effekten eine Entscheidung für das GARCH(1,1)-Modell getroffen. Da - abgesehen von μ - auch alle Parameter der Modelle signifikant sind, kann davon ausgegangen werden, dass es sich hier um eine adäquate Anpassung handelt.

GARCH-n						
Parameter	ATX			DAX		
	Wert	Std	p-Wert	Wert	Std	p-Wert
μ	0.0687	0.0467	0.1414	0.0271	0.0382	0.4781
ω	0.0893	0.0306	0.0035	0.0518	0.0167	0.0020
α_1	0.1514	0.0271	0.0000	0.1148	0.0209	0.0000
γ	0.8316	0.0271	0.0000	0.8672	0.0216	0.0000
d						
LLF	-1.9829			-1.7491		
AIC	3.9738			3.5062		
BIC	3.9934			3.5258		
Lags	Q-Statistik	p-Wert		Q-Statistik	p-Wert	
Ljung-Box-Test						
QW(1)	0.5724	0.4493		0.1315	0.7168	
QW(3)	1.8410	0.6061		0.8997	0.8255	
QW(5)	2.3676	0.7963		1.9760	0.8525	
QW(7)	2.3678	0.9367		2.9425	0.8903	
Engle-Test						
LM(4)	5.8769	0.2085		23.4199	0.0001	
LM(6)	8.5699	0.1993		24.7062	0.0004	
LM(8)	9.7155	0.2856		24.9469	0.0016	
LM(10)	11.5674	0.3150		25.1672	0.0050	

Tabelle 3.2: Ergebnisse der univariaten GARCH(1,1)-n-Modelle

Dasselbe gilt für die Ergebnisse der GARCH-t-Modelle in Tabelle 3.3. Die hier geschätzten Parameter unterscheiden sich zudem nur geringfügig von denen der GARCH-n-Modelle.

GARCH-t						
Parameter	ATX			DAX		
	Wert	Std	p-Wert	Wert	Std	p-Wert
μ	0.0809	0.0458	0.0772	0.0271	0.0376	0.4717
ω	0.0650	0.0287	0.0236	0.0423	0.0180	0.0185
α_1	0.1374	0.0288	0.0000	0.1008	0.0217	0.0000
γ	0.8542	0.0280	0.0000	0.8849	0.0221	0.0000
d	10.0000	2.4066	0.0000	7.2251	1.6839	0.0000
LLF	-1.9766			-1.7313		
AIC	3.9633			3.4726		
BIC	3.9878			3.4972		
Lags	Q-Statistik	p-Wert		Q-Statistik	p-Wert	
Ljung-Box-Test						
QW(1)	0.4459	0.5043		0.1458	0.7026	
QW(3)	1.8156	0.6115		0.8586	0.8354	
QW(5)	2.2945	0.8071		1.9079	0.8617	
QW(7)	2.2948	0.9417		2.7979	0.9031	
Engle-Test						
LM(4)	6.3105	0.1771		22.8921	0.0001	
LM(6)	8.8637	0.1814		24.1476	0.0005	
LM(8)	10.2527	0.2477		24.4484	0.0019	
LM(10)	11.9917	0.2856		24.6336	0.0061	

Tabelle 3.3: Ergebnisse der univariaten GARCH(1,1)-t-Modelle

3.3 Copula-Modellierung

Anhand von Tabelle 3.4 ist erkennbar, dass laut den Selektionskriterien AIC, BIC und LLF für die ersten 1000 Beobachtungen die BB1 Copula mit GARCH-t-Randverteilungen am geeignetsten erscheint. Ebenfalls niedrige AIC- und BIC-Werte sowie eine hohe LLF weist die Student t-Copula mit GARCH-t-Randverteilungen auf. Bei Verwendung von GARCH-n-Randverteilungen schneidet ebenfalls die Student t-Copula am besten ab. Die Parameter der Copulas wurden mittels MLE-Schätzung bestimmt. Einzig bei der Gaussian Copula wurde in Anlehnung an Huang et al. (2009) die Kendall's τ Transformation eingesetzt. Kendall's τ stellt ein Maß für die Stärke des Zusammenhangs zwischen zwei Zufallsvariablen dar (Joe, 1997, S. 31-32) und wird hier anstelle der Parameter, die sich je nach Copula in ihrer Anzahl unterscheiden - zur besseren Vergleichbarkeit herangezogen. Wie man ablesen kann, liegt Kendall's τ bei allen Copulas zwischen 0.4 und 0.65, was auf einen mittelstarken Zusammenhang hinweist. Bei den Modellen, die laut den Selektionskriterien am besten abschneiden, hat Kendall's τ bei allen mindestens den Wert

0.6.

Copula		GARCH-n	GARCH-t	Copula		GARCH-n	GARCH-t
Gaussian	τ	0.592	0.592	Clayton-Gumbel (BB1)	τ	0.467	0.623
	AIC	-964.354	-993.407		AIC	-858.208	-1012.405
	BIC	-959.446	-988.499		BIC	-848.392	-1002.589
	LLF	483.177	497.704		LLF	431.104	508.202
Student t	τ	0.582	0.629	Joe-Gumbel (BB6)	τ	0.480	0.619
	AIC	-967.919	-1011.251		AIC	-853.504	-995.757
	BIC	-958.103	-1001.436		BIC	-843.689	-985.941
	LLF	485.959	501.143		LLF	428.752	499.878
Clayton	τ	0.401	0.489	Joe-Clayton (BB7)	τ	0.524	0.593
	AIC	-603.818	-729.649		AIC	-876.693	-968.482
	BIC	-598.910	-724.741		BIC	-866.878	-958.667
	LLF	302.909	365.824		LLF	440.347	486.241
Gumbel	τ	0.574	0.619	Sur. Clayton	τ	0.500	0.552
	AIC	-952.105	-997.823		AIC	-768.927	-855.022
	BIC	-947.197	-992.915		BIC	-764.019	-850.115
	LLF	477.052	499.911		LLF	385.463	428.511
Frank	τ	0.604	0.622	Sur. Gumbel	τ	0.520	0.588
	AIC	-950.802	-963.907		AIC	-798.561	-901.659
	BIC	-945.894	-958.999		BIC	-793.653	-896.752
	LLF	476.401	482.953		LLF	400.280	451.830

Tabelle 3.4: Copula-Selektion - Kendall's τ und Selektionskriterien

3.4 VaR-Schätzung

Abbildung 3.3 zeigt exemplarisch die VaR-Schätzung mittels Survival Clayton Copula und GARCH-t-Randverteilungen. Die blauen Punkte stellen die Portfoliorenditen der Testdaten dar. Die orange und die grüne Linie zeigen die VaR-Schätzungen mit einer Sicherheit von 95 bzw. 99 Prozent. Für diese Schätzungen wurden - wie bereits oben beschrieben - immer jeweils die letzten 1000 Beobachtungen herangezogen. Laut Definition des VaR (siehe Kapitel 2.2) sollten im Idealfall - also wenn die Methode gut zur Schätzung geeignet ist - 5 Prozent aller Punkte unterhalb der orangefarbenen und 1 Prozent aller Punkte unterhalb der grünen Linie liegen. Da dies anhand der Abbildung nicht sonderlich gut zu erkennen ist, werden im Folgenden die Detailergebnisse in Tabellen präsentiert. Aus Tabelle 3.5 lässt sich ablesen, dass bei einer VaR-Schätzung, die mit einer Sicherheit von 95 Prozent angegeben werden kann, 61 Verletzungen erwartet werden. Wenn beispielsweise die Gaussian Copula mit GARCH-n-Randverteilungen zur Schätzung herangezogen wird,

dann liegen 75 Verletzungen vor anstatt 61. Wenn man für jede Copula - unabhängig von der Randverteilung - den MAE berechnet, dann liegen die Gaussian Copula und die Student t-Copula durchschnittlich im Vergleich zu den anderen Copulas am nächsten bei der erwarteten Verletzungsanzahl.

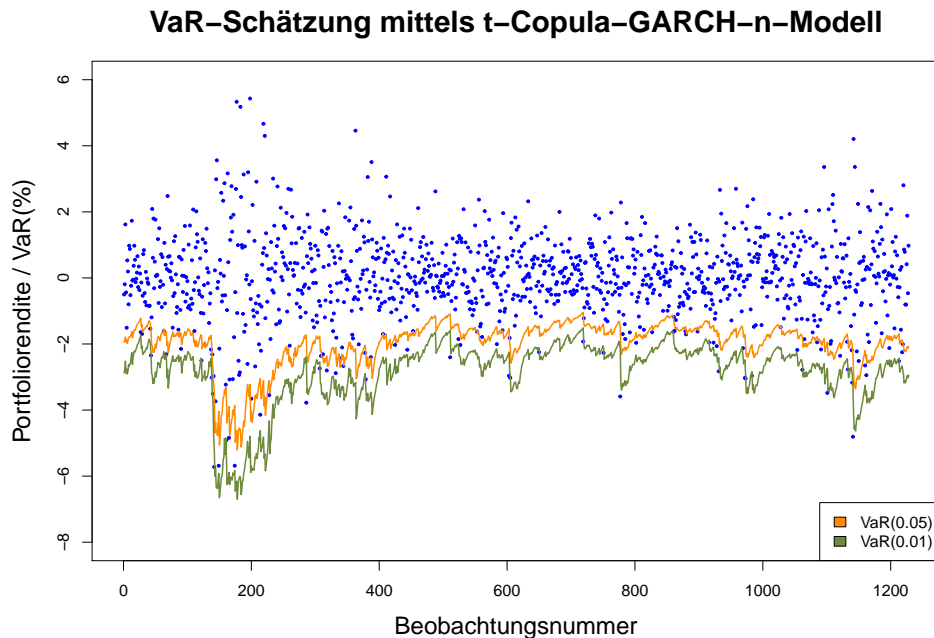


Abbildung 3.3: VaR-Schätzung mittels Survival-Clayton-Copula-GARCH-t-Modell

Copula	GARCH-n	GARCH-t	MAE
Gaussian	75	41	17
Student t	75	41	17
Clayton	74	38	18
Gumbel	80	42	19
Frank	77	42	17.5
BB1	78	41	18.5
BB6	80	47	16.5
BB7	80	41	19.5
BB8	82	46	18
Survival Clayton	93	51	21
Survival Gumbel	74	39	17.5

Tabelle 3.5: Anzahl der Verletzungen bei der VaR-Schätzung (erwartet: 61; $\alpha=0.05$)

Bei einer VaR-Schätzung mit einer Sicherheit von 99 Prozent sollen 12 Verletzungen begangen werden wie in Tabelle 3.6 erkennbar ist. In diesem Fall liegen die Clayton Copula und die Survival Gumbel Copula im Durchschnitt am nächsten bei der erwarteten Zahl an Verletzungen. Allerdings sagt diese Darstellungsweise nichts darüber aus, wie ein einzelnes

Modell abschneidet, sondern es kann nur die Robustheit gegenüber den Randverteilungen beurteilt werden, daher wird noch auf eine weitere Darstellungsform zurückgegriffen (Tabelle 3.7).

Copula	GARCH-n	GARCH-t	MAE
Gaussian	25	5	10
Student t	24	5	9.5
Clayton	23	5	9
Gumbel	29	5	12
Frank	37	5	16
BB1	26	5	10.5
BB6	29	7	11
BB7	25	5	10
BB8	43	12	15.5
Survival Clayton	48	13	18.5
Survival Gumbel	23	5	9

Tabelle 3.6: Anzahl der Verletzungen bei der VaR-Schätzung (erwartet: 12; $\alpha=0.01$)

Wenn die Anzahl der Verletzungen für jedes Modell, sowohl bei einem Signifikanzniveau von 5 als auch von 1 Prozent in Tabelle 3.7 betrachtet wird, nähert sich die Verletzungsanzahl der Survival Clayton Copula mit GARCH-t-Randverteilungen am ehesten an die erwartete Anzahl an. So werden bei $\alpha = 0.05$ 51 statt 61 Verletzungen geschätzt und bei $\alpha = 0.01$ 13 anstelle von 12. Fraglich ist allerdings, ob Abweichungen von der erwarteten Verletzungsanzahl nach oben genauso beurteilt werden sollen wie solche nach unten. Schließlich ist es in der Praxis für PortfoliobesitzerInnen nicht so gravierend, wenn der VaR zum Beispiel nur 40 Mal anstelle von 60 Mal unterschritten wird. Im umgekehrten Fall, also beispielsweise bei 80 Unterschreitungen sind die Auswirkungen schwerwiegender, was bedeutet, dass eine Unterschätzung der erwarteten Verletzungsanzahl weniger negativ als eine Überschätzung derer beurteilt werden sollte.

In diesem Sinn kann tendenziell eine Empfehlung für die Verwendung von GARCH-t-Randverteilungen gegeben werden, da diese die erwartete Verletzungsanzahl fast immer unterschätzen, während es bei GARCH-n-Randverteilungen eher zu einer Überschätzung kommt. Da beim Survival-Clayton-GARCH-t-Modell die erwartete Verletzungsanzahl bei $\alpha = 0.05$ unterschätzt und bei $\alpha = 0.01$ nur um eins überschätzt wird, scheint dies auch in dieser Hinsicht ein durchaus adäquates Modell zu sein. Im Vergleich dazu kann die Survival-Clayton-Copula-GARCH-n-Methode hier als die ungeeignetste bezeichnet werden, da sie nicht nur den höchsten MAE aufweist, sondern auch die Anzahl an Verletzungen weit überschätzt.

Erwartete Verletzungszahl	61	12		Erwartete Verletzungszahl	61	12	
GARCH-n	0.05	0.01	MAE	GARCH-t	0.05	0.01	MAE
Gaussian	75	25	27	Gaussian	41	5	27
Student t	75	24	26	Student t	41	5	27
Clayton	74	23	24	Clayton	38	5	30
Gumbel	80	29	36	Gumbel	42	5	26
Frank	77	37	41	Frank	42	5	26
BB1	78	26	31	BB1	41	5	27
BB6	80	29	36	BB6	47	7	19
BB7	80	25	32	BB7	41	5	27
BB8	82	43	52	BB8	46	12	15
Survival Clayton	93	48	68	Survival Clayton	51	13	11
Survival Gumbel	74	23	24	Survival Gumbel	39	5	29

Tabelle 3.7: Anzahl der Verletzungen bei der VaR-Schätzung

4. Vergleich mit herkömmlichen VaR-Schätzmethoden

Wie bereits in Kapitel 1.1 angesprochen wurde, gibt es viele Herangehensweisen um VaR zu schätzen. Im Folgenden werden einige dieser Methoden kurz vorgestellt und anschließend mit der bedingten Copula-GARCH-Methode verglichen:

(a) **Historische Simulation** (Fricke, 2006, S. 30-33)

Die historische Simulation trifft bezüglich Verteilung der Portfoliorendite keine parametrische Verteilungsannahme. Es wird also die empirische Verteilungsfunktion der historischen Daten verwendet und ein Quantil geschätzt. Positiv ist, dass es ein recht einfacher Ansatz ist. Probleme kann allerdings die Auswahl der Datenlänge machen, da hier alle Beobachtungen gleich gewichtet eingehen und somit auch weiter zurückliegende Beobachtungen großen Einfluss auf die Schätzung haben. Zudem wird die Volatilität der Renditenzeitreihe als konstant betrachtet, was in der Praxis nicht gegeben ist.

(b) **Varianz-Kovarianz-Methode** (Fricke, 2006, S. 28-30)

Die Varianz-Kovarianz-Methode ist ebenfalls ein einfacher Ansatz. Hier wird die Verteilungsfunktion der Portfoliorendite als normalverteilt angenommen, was oft als nicht haltbar kritisiert wird. Auch hier wird die Volatilität als konstant betrachtet. Die Berechnung des VaR kann wie folgt definiert werden (Huang et al., 2009, S. 319):

$$\sigma_{p,t}^2 = \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \times \begin{bmatrix} \sigma_{1,t}^2 & \sigma_{12,t} \\ \sigma_{21,t} & \sigma_{2,t}^2 \end{bmatrix} \times \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \omega \Sigma_t \omega, \quad (4.1)$$

$$VaR_{p,t}(\alpha) = \sigma_{p,t} \cdot Z_\alpha + \mu_{p,t}. \quad (4.2)$$

Dabei stellen $\mu_{p,t}$ und $\sigma_{p,t}^2$ den Mittelwert und die Varianz des Portfolios zum Zeitpunkt t dar. ω_1 und ω_2 sind die Gewichte der beiden Aktienindizes und Z_α ist das Standardnormalverteilungsquantil zur Wahrscheinlichkeit α .

- (c) **EWMA-Methode** (exponentially weighted moving average-Methode) (Fricke, 2006, S. 33-35 u. 42)

Das EWMA-Modell ist ein Spezialfall des GARCH(1,1)-Modells (siehe Kapitel 2.3), wobei die Stationaritätsbedingung nicht gilt. Bei diesem Ansatz handelt es sich - genau wie bei der Varianz-Kovarianz-Methode um einen parametrischen Ansatz. In der Regel wird eine Normalverteilung mit $\mu = 0$ angenommen. Ein Vorteil dieser Methode ist, dass die zeitveränderliche Volatilität explizit modelliert wird. Aktuellere Beobachtungen gehen mit einem stärkeren Gewicht ein, wodurch das Modell sehr schnell auf ungewöhnlich hohe bzw. niedrige Renditen reagieren kann. Berechnet wird die bedingte Volatilität indem beim GARCH(1,1)-Modell $\mu = 0, \alpha_0 = 0, \alpha_1 = 1 - \lambda$ und $\gamma = \lambda$ gesetzt werden.

- (d) **Univariate GARCH-VaR-Methode** (Fricke, 2006, S. 38-42)

Die univariate GARCH-Methode ist nicht auf die Normalverteilung beschränkt und modelliert ebenfalls die zeitveränderliche Volatilität. Zudem wird im Gegensatz zum EWMA-Modell ein langfristiger Mittelwert modelliert. Die Berechnung erfolgt durch Anpassung eines GARCH(1,1)-Modells an die Portfoliorenditenzeitreihe, wobei dann $\sigma_{p,t}^2 | \Omega_{t-1}$ berechnet wird. Dadurch ergibt sich (Huang et al., 2009, S. 319):

$$VaR_{p,t}(\alpha) | \Omega_{t-1} = (\sigma_{p,t} \cdot Z_\alpha + \mu_{p,t}) | \Omega_{t-1}. \quad (4.3)$$

- (e) **Multivariate GARCH-VaR-Methode** (Fricke, 2006, S. 48)

Bei der multivariaten GARCH-Methode wird ebenfalls die zeitveränderliche Komponente berücksichtigt. Zudem werden - wie bei der bedingten Copula-GARCH-Methode - auch die Abhängigkeiten zwischen den Renditen der einzelnen Aktienindizes explizit modelliert, was die VaR-Schätzungen verbessern soll. Ein Nachteil ist die hohe Komplexität dieser Methode.

Nachfolgend wird der Vergleich zwischen den herkömmlichen Schätzverfahren und der geeignetsten bzw. ungeeignetsten Copula-GARCH-Methode diskutiert. Dabei wird allerdings auf die Anwendung der multivariaten GARCH-VaR-Methode verzichtet, da dies den Umfang dieser Bachelorarbeit übersteigen würde. Wenn man nun in Abbildung 4.1 die Ergebnisse der verschiedenen Methoden betrachtet, dann lässt sich erkennen, dass die historische Simulation und die Varianz-Kovarianz-Methode die zeitveränderliche Volatilität nicht abbilden. Die anderen Methoden liefern recht ähnliche Ergebnisse, mit Ausnahme der Copula-GARCH-Modelle, wobei das Survival Clayton-GARCH-t-Modell stärker und das Survival-Clayton-GARCH-n-Modell weniger stark streut als die übrigen Methoden.

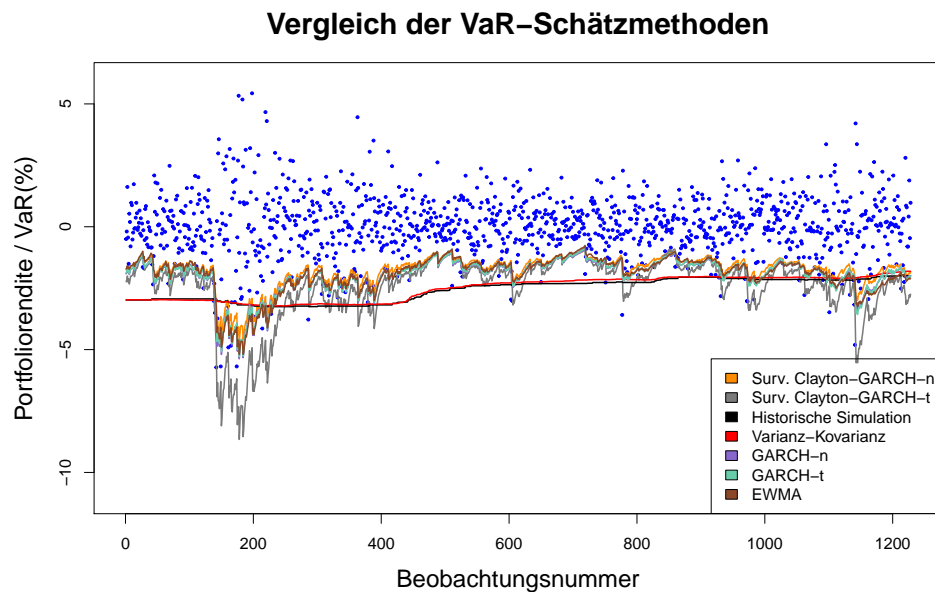


Abbildung 4.1: Vergleich der VaR-Schätzmethoden

Betrachtet man die Verletzungsanzahlen in Tabelle 4.1, kann man erkennen, dass laut MAE das Survival-Clayton-Copula-GARCH-t-Modell am besten abschneidet, gefolgt von der Varianz-Kovarianz-Methode. Dass die Varianz-Kovarianz-Methode so gut abschneidet, ist durchaus überraschend, da sie ja die Zeitveränderlichkeit nicht berücksichtigt, lässt sich aber wahrscheinlich dadurch erklären, dass die Volatilität in diesem Fall sehr stark schwankt und die Varianz-Kovarianz-Methode somit im Durchschnitt mit ihrer Konstanz dennoch gute Ergebnisse erzielt. Nicht geeignet ist das Survival-Clayton-Copula-GARCH-n-Modell.

α	0.05	0.01	
Erwartete Verletzungsanzahl	61	12	MAE
Surv. Clayton-GARCH-n	93	48	68
Surv. Clayton-GARCH-t	51	13	11
Historische Simulation	39	3	31
Varianz-Kovarianz	41	13	21
GARCH-n	76	30	33
GARCH-t	77	29	33
EWMA	82	32	41

Tabelle 4.1: Vergleich der VaR-Schätzmethoden

5. Fazit

Zusammenfassend lässt sich sagen, dass die Survival Clayton Copula mit GARCH-t-Randverteilungen von allen Modellen am besten abschneidet - sowohl in Hinblick auf den MAE, als auch dahingehend, dass die Verletzungsanzahl eher unter- als überschritten wird. Bei der Copula-Selektion ergeben sich laut Kriterien wie AIC und BIC allerdings keine Hinweise darauf, dass dieses Modell geeignet ist, was für AnwenderInnen eventuell problematisch sein könnte. Ähnlich wie bei Huang et al. (2009) wird die Anzahl an Verletzungen bei der Verwendung von GARCH-n-Randverteilungen eher über- und von GARCH-t-Randverteilungen unterschätzt, wodurch sich die Verwendung von GARCH-Modellen mit t-verteilten Fehlern empfiehlt.

Beim bedingten Copula-GARCH-Modell handelt es sich somit grundsätzlich um eine durchaus geeignete Methode zur Schätzung von VaR. Bedacht werden muss allerdings, dass stark ändernde Volatilitäten zu Ungenauigkeiten führen können. Dadurch kann es passieren, dass - wie in diesem Fall - weniger komplexe Modelle wie die Varianz-Kovarianz-Methode geeigneter sind, um VaR zu schätzen, als die meisten bedingten Copula-GARCH-Modelle.

Anhang

R-Code

```
#-----#
#                               Einlesen der Daten
#-----#

D1<-read.table(file="ATX_PI.txt",header=T,dec=",")
D2<-read.table(file="DAX_PI.txt",header=T,dec=",")

D<-cbind(D1,D2)
D<-D[,-(seq(3,ncol(D),2))]

#-----#
#                               Berechnung der Renditen
#-----#

Returns<-function(x){
  ret<-vector("numeric",length=length(x)-1)
  for(i in 2:length(x)){
    ret[i-1]<-100*log(x[i]/x[i-1])
  }
  return(ret)
}

D[-1,2:dim(D)[2]]<-apply(D[,2:(dim(D)[2])],2,Returns)
D<-D[-1,]
colnames(D)<-c("Date","ATX","DAX")
D<-as.data.frame(D)
```

```
#-----#  
#           Feiertagsbereinigung  
#-----#  
  
D[D==0]<-NA  
data<-na.omit(D)  
data<-data[-c(1,dim(data)[1]),]  
  
#-----#  
#           Grafische Darstellung der Indizes  
#-----#  
  
plot.ts(D1[,2],col="firebrick3",main="Aktienindex ATX"  
        ,xlab="Zeit",ylab="Indexwert",cex.main=1.7)  
plot.ts(D2[,2],col="firebrick3",main="Aktienindex DAX"  
        ,xlab="Zeit",ylab="Indexwert",cex.main=1.7)  
  
#-----#  
#           Grafische Darstellung der Renditen  
#-----#  
  
plot.ts(data[,2],col="blue4",main="Rendite ATX (feiertags  
        bereinigt)",xlab="Zeit",ylab="Rendite (%)"  
        ,cex.main=1.7,ylim=c(-13,13))  
plot.ts(data[,3],col="blue4",main="Rendite DAX (feiertags  
        bereinigt)",xlab="Zeit",ylab="Rendite (%)"  
        ,cex.main=1.7,ylim=c(-13,13))
```

```
#-----#  
#           Deskriptive Kennzahlen  
#-----#
```

```
mean(data$DAX)  
mean(data$ATX)  
sd(data$DAX)  
sd(data$ATX)
```

```
library(e1071)  
skewness(data$DAX)  
skewness(data$ATX)  
kurtosis(data$DAX)  
kurtosis(data$ATX)
```

```
#-----#  
#           Engle-Tests  
#-----#
```

```
library(FinTS)  
ArchTest(data$DAX^2, lags=4)  
ArchTest(data$DAX^2, lags=6)  
ArchTest(data$DAX^2, lags=8)  
ArchTest(data$DAX^2, lags=10)
```

```
ArchTest(data$ATX^2, lags=4)  
ArchTest(data$ATX^2, lags=6)  
ArchTest(data$ATX^2, lags=8)  
ArchTest(data$ATX^2, lags=10)
```

```
#-----#  
#           Schätzung der GARCH(1,1)-Modelle  
#-----#
```

```
library(fGarch)
```

```

D1<-data$DAX[1:1000]
D2<-data$ATX[1:1000]

G1N<-garchFit(formula=~garch(1,1),D1,include.mean = T)
G1T<-garchFit(formula=~garch(1,1),D1,include.mean = T
               ,cond.dist = "std")
G2N<-garchFit(formula=~garch(1,1),D2,include.mean = T)
G2T<-garchFit(formula=~garch(1,1),D2,include.mean = T
               ,cond.dist = "std")

#-----#
#           LLF, AIC und BIC der GARCH-Modelle
#-----#

summary(G1N)
summary(G1T)
summary(G2N)
summary(G2T)

#-----#
#           Transformation der Renditen
#-----#

m1<-G1N@fit$par[1]
m2<-G2N@fit$par[1]
m1T<-G1T@fit$par[1]
m2T<-G2T@fit$par[1]
uN<-cbind(pnorm(D1,m1,G1N@sigma.t)
           ,pnorm(D2,m2,G2N@sigma.t))
df1<-G1T@fit$par[5]
df2<-G2T@fit$par[5]
uT<-cbind(pt((D1-m1T)/G1T@sigma.t,df1)
           ,pt((D2-m2T)/G2T@sigma.t,df2))

```

```
#-----#  
#           Ljung-Box-Test  
#-----#
```

```
Box.test (uN[,1], lag = 1, type = "Ljung")  
Box.test (uN[,1], lag = 3, type = "Ljung")  
Box.test (uN[,1], lag = 5, type = "Ljung")  
Box.test (uN[,1], lag = 7, type = "Ljung")
```

```
Box.test (uN[,2], lag = 1, type = "Ljung")  
Box.test (uN[,2], lag = 3, type = "Ljung")  
Box.test (uN[,2], lag = 5, type = "Ljung")  
Box.test (uN[,2], lag = 7, type = "Ljung")
```

```
Box.test (uT[,1], lag = 1, type = "Ljung")  
Box.test (uT[,1], lag = 3, type = "Ljung")  
Box.test (uT[,1], lag = 5, type = "Ljung")  
Box.test (uT[,1], lag = 7, type = "Ljung")
```

```
Box.test (uT[,2], lag = 1, type = "Ljung")  
Box.test (uT[,2], lag = 3, type = "Ljung")  
Box.test (uT[,2], lag = 5, type = "Ljung")  
Box.test (uT[,2], lag = 7, type = "Ljung")
```

```
#-----#  
#           Engle-Tests  
#-----#
```

```
ArchTest(uN[,1]^2, lags=4)  
ArchTest(uN[,1]^2, lags=6)  
ArchTest(uN[,1]^2, lags=8)  
ArchTest(uN[,1]^2, lags=10)
```

```
ArchTest(uN[,2]^2, lags=4)  
ArchTest(uN[,2]^2, lags=6)  
ArchTest(uN[,2]^2, lags=8)  
ArchTest(uN[,2]^2, lags=10)
```



```

ArchTest(uT[,1]^2, lags=4)
ArchTest(uT[,1]^2, lags=6)
ArchTest(uT[,1]^2, lags=8)
ArchTest(uT[,1]^2, lags=10)

```

```

ArchTest(uT[,2]^2, lags=4)
ArchTest(uT[,2]^2, lags=6)
ArchTest(uT[,2]^2, lags=8)
ArchTest(uT[,2]^2, lags=10)

```

```

#-----#
#           Copula-Selektion und Parameterschätzung
#-----#

```

```

library(copula)
library(VineCopula)
library(CDVine)

```

```

BiCopSelect(uN[,1], uN[,2]
            , familyset=c(1,2,3,4,5,7,8,9,10,13,14))
BiCopSelect(uN[,1], uN[,2]
            , familyset=c(1,2,3,4,5,7,8,9,10,13,14)
            , selectioncrit = "BIC")
BiCopSelect(uT[,1], uT[,2]
            , familyset=c(1,2,3,4,5,7,8,9,10,13,14))
BiCopSelect(uT[,1], uT[,2]
            , familyset=c(1,2,3,4,5,7,8,9,10,13,14)
            , selectioncrit = "BIC")

```

```

fNn<-BiCopEst(uN[,1], uN[,2], 1, method="itau")
BiCopPar2Tau(1, fNn$par, fNn$par2)
CDVineAIC(uN, 1, fNn$par, type=1)$AIC
CDVineBIC(uN, 1, fNn$par, type=1)$BIC
CDVineLogLik(uN, 1, fNn$par, type=1)$loglik

```

```

fTn<-BiCopEst(uT[,1],uT[,2],1,method="itau")
BiCopPar2Tau(1,fTn$par,fTn$par2)
CDVineAIC(uT,1,fTn$par,type=1)$AIC
CDVineBIC(uT,1,fTn$par,type=1)$BIC
CDVineLogLik(uT,1,fTn$par,type=1)$loglik

fNt<-BiCopEst(uN[,1],uN[,2],2,method="mle")
BiCopPar2Tau(2,fNt$par,fNt$par2)
CDVineAIC(uN,2,fNt$par,fNt$par2,type=1)$AIC
CDVineBIC(uN,2,fNt$par,fNt$par2,type=1)$BIC
CDVineLogLik(uN,2,fNt$par,fNt$par2,type=1)$loglik
fTt<-BiCopEst(uT[,1],uT[,2],2,method="mle")
BiCopPar2Tau(2,fTt$par,fTt$par2)
CDVineAIC(uT,2,fTt$par,fTt$par2,type=1)$AIC
CDVineBIC(uT,2,fTt$par,fTt$par2,type=1)$BIC
CDVineLogLik(uT,2,fTn$par,fTt$par2,type=1)$loglik

fNc<-BiCopEst(uN[,1],uN[,2],3,method="mle")
BiCopPar2Tau(3,fNc$par,fNc$par2)
CDVineAIC(uN,3,fNc$par,type=1)$AIC
CDVineBIC(uN,3,fNc$par,type=1)$BIC
CDVineLogLik(uN,3,fNc$par,type=1)$loglik
fTc<-BiCopEst(uT[,1],uT[,2],3,method="mle")
BiCopPar2Tau(3,fTc$par,fTc$par2)
CDVineAIC(uT,3,fTc$par,type=1)$AIC
CDVineBIC(uT,3,fTc$par,type=1)$BIC
CDVineLogLik(uT,3,fTc$par,type=1)$loglik

fNg<-BiCopEst(uN[,1],uN[,2],4,method="mle")
BiCopPar2Tau(4,fNg$par,fNg$par2)
CDVineAIC(uN,4,fNg$par,type=1)$AIC
CDVineBIC(uN,4,fNg$par,type=1)$BIC
CDVineLogLik(uN,4,fNg$par,type=1)$loglik
fTg<-BiCopEst(uT[,1],uT[,2],4,method="mle")
BiCopPar2Tau(4,fTg$par,fTg$par2)
CDVineAIC(uT,4,fTg$par,type=1)$AIC
CDVineBIC(uT,4,fTg$par,type=1)$BIC
CDVineLogLik(uT,4,fTg$par,type=1)$loglik

```

```

fNf<-BiCopEst(uN[,1],uN[,2],5,method="mle")
BiCopPar2Tau(5,fNf$par,fNf$par2)
CDVineAIC(uN,5,fNf$par,type=1)$AIC
CDVineBIC(uN,5,fNf$par,type=1)$BIC
CDVineLogLik(uN,5,fNf$par,type=1)$loglik
fTf<-BiCopEst(uT[,1],uT[,2],5,method="mle")
BiCopPar2Tau(5,fTf$par,fTf$par2)
CDVineAIC(uT,5,fTf$par,type=1)$AIC
CDVineBIC(uT,5,fTf$par,type=1)$BIC
CDVineLogLik(uT,5,fTf$par,type=1)$loglik

fNB1<-BiCopEst(uN[,1],uN[,2],7,method="mle")
BiCopPar2Tau(7,fNB1$par,fNB1$par2)
CDVineAIC(uN,7,fNB1$par,fNB1$par2,type=1)$AIC
CDVineBIC(uN,7,fNB1$par,fNB1$par2,type=1)$BIC
CDVineLogLik(uN,7,fNB1$par,fNB1$par2,type=1)$loglik
fTB1<-BiCopEst(uT[,1],uT[,2],7,method="mle")
BiCopPar2Tau(7,fTB1$par,fTB1$par2)
CDVineAIC(uT,7,fTB1$par,fTB1$par2,type=1)$AIC
CDVineBIC(uT,7,fTB1$par,fTB1$par2,type=1)$BIC
CDVineLogLik(uT,7,fTB1$par,fTB1$par2,type=1)$loglik

fNB6<-BiCopEst(uN[,1],uN[,2],8,method="mle")
BiCopPar2Tau(8,fNB6$par,fNB6$par2)
CDVineAIC(uN,8,fNB6$par,fNB6$par2,type=1)$AIC
CDVineBIC(uN,8,fNB6$par,fNB6$par2,type=1)$BIC
CDVineLogLik(uN,8,fNB6$par,fNB6$par2,type=1)$loglik
fTB6<-BiCopEst(uT[,1],uT[,2],8,method="mle")
BiCopPar2Tau(8,fTB6$par,fTB6$par2)
CDVineAIC(uT,8,fTB6$par,fTB6$par2,type=1)$AIC
CDVineBIC(uT,8,fTB6$par,fTB6$par2,type=1)$BIC
CDVineLogLik(uT,8,fTB6$par,fTB6$par2,type=1)$loglik

fNB7<-BiCopEst(uN[,1],uN[,2],9,method="mle")
BiCopPar2Tau(9,fNB7$par,fNB7$par2)
CDVineAIC(uN,9,fNB7$par,fNB7$par2,type=1)$AIC
CDVineBIC(uN,9,fNB7$par,fNB7$par2,type=1)$BIC
CDVineLogLik(uN,9,fNB7$par,fNB7$par2,type=1)$loglik

```

```

fTB7<-BiCopEst(uT[,1],uT[,2],9,method="mle")
BiCopPar2Tau(9,fTB7$par,fTB7$par2)
CDVineAIC(uT,9,fTB7$par,fTB7$par2,type=1)$AIC
CDVineBIC(uT,9,fTB7$par,fTB7$par2,type=1)$BIC
CDVineLogLik(uT,9,fTB7$par,fTB7$par2,type=1)$loglik

fNB8<-BiCopEst(uN[,1],uN[,2],10,method="mle")
BiCopPar2Tau(10,fNB8$par,fNB8$par2)
CDVineAIC(uN,10,fNB8$par,fNB8$par2,type=1)$AIC
CDVineBIC(uN,10,fNB8$par,fNB8$par2,type=1)$BIC
CDVineLogLik(uN,10,fNB8$par,fNB8$par2,type=1)$loglik
fTB8<-BiCopEst(uT[,1],uT[,2],10,method="mle")
BiCopPar2Tau(10,fTB8$par,fTB8$par2)
CDVineAIC(uT,10,fTB8$par,fTB8$par2,type=1)$AIC
CDVineBIC(uT,10,fTB8$par,fTB8$par2,type=1)$BIC
CDVineLogLik(uT,10,fTB8$par,fTB8$par2,type=1)$loglik

fNcr<-BiCopEst(uN[,1],uN[,2],13,method="mle")
BiCopPar2Tau(13,fNcr$par,fNcr$par2)
CDVineAIC(uN,13,fNcr$par,type=1)$AIC
CDVineBIC(uN,13,fNcr$par,type=1)$BIC
CDVineLogLik(uN,13,fNcr$par,type=1)$loglik
fTcr<-BiCopEst(uT[,1],uT[,2],13,method="mle")
BiCopPar2Tau(13,fTcr$par,fTcr$par2)
CDVineAIC(uT,13,fTcr$par,type=1)$AIC
CDVineBIC(uT,13,fTcr$par,type=1)$BIC
CDVineLogLik(uT,13,fTcr$par,type=1)$loglik

fNgr<-BiCopEst(uN[,1],uN[,2],14,method="mle")
BiCopPar2Tau(14,fNgr$par,fNgr$par2)
CDVineAIC(uN,14,fNgr$par,type=1)$AIC
CDVineBIC(uN,14,fNgr$par,type=1)$BIC
CDVineLogLik(uN,14,fNgr$par,type=1)$loglik
fTgr<-BiCopEst(uT[,1],uT[,2],14,method="mle")
BiCopPar2Tau(14,fTgr$par,fTgr$par2)
CDVineAIC(uT,14,fTgr$par,type=1)$AIC
CDVineBIC(uT,14,fTgr$par,type=1)$BIC
CDVineLogLik(uT,14,fTgr$par,type=1)$loglik

```

```

#-----#
#           Herkömmliche VaR-Schätzmethoden
#-----#

library(PerformanceAnalytics)
library(MIS)
library(rugarch)

VaR_other_0.05<-matrix(0,ncol=5,nrow=dim(data)[1]-1000)
VaR_other_0.01<-matrix(0,ncol=5,nrow=dim(data)[1]-1000)
colnames(VaR_other_0.05)<-colnames(VaR_other_0.01)<-
  c("Historical Simulation","Variance-Covariance"
    ,"GARCH-n","GARCH-t","EWMA")
PR<-1/2*data$ATX+1/2*data$DAX

OD1<-data$DAX[1001:dim(data)[1]]
OD2<-data$ATX[1001:dim(data)[1]]
portRet<-1/2*OD1+1/2*OD2

specN<-ugarchspec(variance.model=list(garchOrder=c(1,1))
  , mean.model=list(armaOrder=c(0,0)
  , include.mean=TRUE)
  , distribution.model="norm")

specT<-ugarchspec(variance.model=list(garchOrder=c(1,1))
  , mean.model=list(armaOrder=c(0,0)
  , include.mean=TRUE)
  , distribution.model="std")

specEWMA<-ugarchspec(variance.model=list(garchOrder=c(1,1))
  , mean.model=list(armaOrder=c(0,0)
  , include.mean=FALSE)
  , distribution.model="norm"
  , fixed.pars=list(omega=0))

```

```

j<-1
for(i in 1:(dim(data)[1]-1000)){

  d1<-data[,3][j:(j+999)]
  d2<-data[,2][j:(j+999)]
  h<-as.matrix(cbind(d1,d2)/100)
  h<-as.xts(h,order.by=as.Date(data$Date[j:(j+999)],format="%d.%m.%Y"))

  VaR_other_0.05[i,1]<-VaR(R=h,method="historical",
    ,portfolio_method="component",weights=c(1/2,1/2))
  VaR_other_0.05[i,2]<-as.vector(VaR(R=h,
    ,method="gaussian",portfolio_method="component",
    ,weights=c(1/2,1/2))$VaR)

  fN<-ugarchfit(specN,PR[j:(j+999)])
  PrN<-ugarchforecast(fN,n.ahead=1)
  VaR_other_0.05[i,3]<-qnorm(0.05)*sigma(PrN)+fitted(PrN)

  fT<-ugarchfit(specT,PR[j:(j+999)])
  PrT<-ugarchforecast(fT,n.ahead=1)
  VaR_other_0.05[i,4]<-qnorm(0.05)*sigma(PrT)+fitted(PrT)

  VaR_other_0.01[i,3]<-qnorm(0.01)*sigma(PrN)+fitted(PrN)
  VaR_other_0.01[i,4]<-qnorm(0.01)*sigma(PrT)+fitted(PrT)

  ew<-ugarchfit(specEWMA,PR[j:(j+999)])
  ewF<-ugarchforecast(ew,n.ahead=1)
  VaR_other_0.05[i,5]<-qnorm(0.05)*sigma(ewF)
  VaR_other_0.01[i,5]<-qnorm(0.01)*sigma(ewF)
  j<-j+1
}

j<-1
for(i in 1:(dim(data)[1]-1000)){
  d1<-data[,3][j:(j+999)]
  d2<-data[,2][j:(j+999)]

```

```

h<-as.matrix(cbind(d1,d2)/100)
h<-as.xts(h,order.by=as.Date(data$Date[j:(j+999)]
        ,format="%d.%m.%Y"))

VaR_other_0.01[i,1]<-VaR(R=h,method="historical"
    ,portfolio_method="component",weights=c(1/2,1/2)
    ,p=0.99)

VaR_other_0.01[i,2]<-as.vector(VaR(R=h,method="gaussian"
    ,portfolio_method="component",p=0.99
    ,weights=c(1/2,1/2))$VaR)

j<-j+1

}

VaR_other_0.05[,c(1:2)]<-VaR_other_0.05[,c(1:2)]*-100
VaR_other_0.01[,c(1:2)]<-VaR_other_0.01[,c(1:2)]*-100

colnames(VaR_other_0.05)<-c("Historische Simulation"
    , "Varianz-Kovarianz-Methode", "GARCH-n"
    , "GARCH-t", "EWMA")
colnames(VaR_other_0.01)<-c("Historische Simulation"
    , "Varianz-Kovarianz-Methode", "GARCH-n"
    , "GARCH-t", "EWMA")

plot(portRet,col="blue",cex=0.4,pch=19
    ,xlim=c(0,dim(data)[1]-1000),ylim=c(-8,6)
    ,cex.lab=1.5,cex.main=1.8,ylab="Portfoliorendite
    / VaR(%)",xlab="Beobachtungsnummer"
    ,main="Herkömmliche VaR-Schätzmethoden")

lines(VaR_other_0.05[,1],col="black")
lines(VaR_other_0.05[,2],col="red")
lines(VaR_other_0.05[,3],col="mediumpurple3")
lines(VaR_other_0.05[,4],col="aquamarine3")
lines(VaR_other_0.05[,5],col="sienna4")

```

```

legend("topright", legend=colnames(VaR_other_0.05)
      , fill=c("black", "red", "mediumpurple3", "aquamarine3"
              , "sienna4"))

plot(portRet, col="blue", cex=0.4, pch=19
     , xlim=c(0, dim(data)[1] - 1000), ylim=c(-8, 6)
     , cex.lab=1.5, cex.main=1.8, ylab="Portfoliorendite
     / VaR(%)", xlab="Beobachtungsnummer"
     , main="Herkömmliche VaR-Schätzmethoden")

lines(VaR_other_0.01[, 1], col="black")
lines(VaR_other_0.01[, 2], col="red")
lines(VaR_other_0.01[, 3], col="mediumpurple3")
lines(VaR_other_0.01[, 4], col="aquamarine3")
lines(VaR_other_0.01[, 5], col="sienna4")

legend("topright", legend=colnames(VaR_other_0.01)
      , fill=c("black", "red", "mediumpurple3", "aquamarine3"
              , "sienna4"))

Vio0.05<-apply(VaR_other_0.05, 2, function(x)
  {length(which(x>=portRet))})
Vio0.01<-apply(VaR_other_0.01, 2, function(x)
  {length(which(x>=portRet))})

ExVi0.05<-round((length(data[, 1]) - 1000)*0.05)
ExVi0.01<-round((length(data[, 1]) - 1000)*0.01)

mErr<-abs(ExVi0.05 - Vio0.05) + abs(ExVi0.01 - Vio0.01)

#-----#
#           Bedingte Copula-GARCH-Methode
#-----#

library(mvtnorm)
library(docopulae)

```



```

VaRCopGARCH<-function(data1 , data2 , cond.dist , copulaNr
                      , alpha , os , ts , useSparseGrid){

  useSparseGrid<-useSparseGrid
  ncube = function(f, lowerLimit , upperLimit , ...) {
    # Vorbereitung auf integrateA
    integrateA(f , lowerLimit , upperLimit , ... , subdivisions=2)
  }

  ncube = nint_integrateNCube_integrate(ncube)
  unlockBinding('nint_integrateNCube'
               , environment(nint_integrate))
  assign('nint_integrateNCube' , ncube
        , envir=environment(nint_integrate))

  if (useSparseGrid) {
    # Vorbereitung auf SparseGrid
    ncube = function(dimension) {
      SparseGrid::createIntegrationGrid('GQU'
                                       , dimension , 25)
    }
    ncube = nint_integrateNCube_SparseGrid(ncube)
    unlockBinding('nint_integrateNCube'
                 , environment(nint_integrate))
    assign('nint_integrateNCube' , ncube
          , envir=environment(nint_integrate))

    nfunc = nint_integrateNFunc_recursive(ncube)
    unlockBinding('nint_integrateNFunc'
                 , environment(nint_integrate))
    assign('nint_integrateNFunc' , nfunc
          , envir=environment(nint_integrate))
  }
  if (length(data1)!=length(data2)){
    stop("Datensätze müssen dieselbe
         Länge aufweisen!")
  }
}

```

```

if(length(data1)<400){
  stop("Die Daten sollten mindestens eine
       Länge von 400 aufweisen!")
}
if(cond.dist!="norm"&cond.dist!="std"){
  stop("Die Fehler der Randverteilungen müssen
       normal- oder t-verteilt sein!")
}
if(alpha<0.01|alpha>0.2){
  stop("alpha sollte zwischen 0.01 und 0.2
       liegen!")
}
if(os>=length(data1)){
  stop("Der Testdatensatz ist mind. so groß
       wie der gesamte Datensatz!")
}
if(ts>=length(data1)){
  stop("Der Trainingsdatensatz ist mind. so
       groß wie der gesamte Datensatz!")
}
if(ts<=0|os<=0){
  stop("Sowohl die Größe des Trainings- als auch
       die des Testdatensatzes muss positiv sein!")
}
if(copulaNr!=1&copulaNr!=2&copulaNr!=3&copulaNr!=
   4&copulaNr!=5&copulaNr!=7&copulaNr!=8&copulaNr!=
   9&copulaNr!=10&copulaNr!=13&copulaNr!=14){
  stop("Keine gültige Copula ausgewählt!")
}

m<-matrix(0,nrow=os,ncol=1)
a<-alpha
if(useSparseGrid){
  f1<-function(U) {
    if (any(U == 0) || any(U == 1))
      return(0)
    dCopula(U,cop)
  }
}

```

```

} else {
  ff1 <- function(U) dCopula(c(U[1], U[2]), cop)
}

if (cond.dist == "norm") {
  ff1 <- function(X) f1(c(pnorm(X[1], m1, s1),
    pnorm(X[2], m2, s2))) * dnorm(X[1], m1, s1)
    * dnorm(X[2], m2, s2)
} else {
  ff1 <- function(X) f1(c(pt((X[1] - m1)/s1, df1),
    pt((X[2] - m2)/s2, df2))) * dt((X[1] - m1)/s1, df1)
    * dt((X[2] - m2)/s2, df2)
}
hh <- function(Vart) {
  s = nint_space(nint_intvDim(-Inf, Inf),
    nint_funcDim(function(x)
      nint_intvDim(-Inf, 2*Vart - x[1])))
  return(nint_integrate(ff1, s) - a)
}

if (useSparseGrid) {
  tff1 = nint_transform(ff1,
    nint_space(nint_intvDim(-Inf, Inf),
      nint_intvDim(-Inf, Inf)), 1:2, 'tan')

  hh <- function(Vart) {
    s = nint_space(nint_intvDim(atan(-Inf), atan(Inf)),
      nint_funcDim(function(x) nint_intvDim(atan(-Inf),
        atan(2*Vart - tan(x[1])))))
    return(nint_integrate(tff1$f, s) - a)
  }
}

```

```
if (copulaNr==1){
  co<-function(y) normalCopula(y)
}
if (copulaNr==2){
  co<-function(y, df) tCopula(y, df=df)
}
if (copulaNr==3){
  co<-function(y) claytonCopula(y)
}
if (copulaNr==4){
  co<-function(y) gumbelCopula(y)
}
if (copulaNr==5){
  co<-function(y) frankCopula(y)
}
if (copulaNr==7){
  co<-function(y) BB1Copula(y)
}
if (copulaNr==8){
  co<-function(y) BB6Copula(y)
}
if (copulaNr==9){
  co<-function(y) BB7Copula(y)
}
if (copulaNr==10){
  co<-function(y) BB8Copula(y)
}
if (copulaNr==13){
  co<-function(y) surClaytonCopula(y)
}
if (copulaNr==14){
  co<-function(y) surGumbelCopula(y)
}

for (i in 1:os){
  D1<-data1[i:(i+(ts-1))]
  D2<-data2[i:(i+(ts-1))]
```

```

G1<-garchFit ( formula=~garch ( 1 , 1 ) , D1
  , include.mean = T , cond.dist=cond.dist )
G2<-garchFit ( formula=~garch ( 1 , 1 ) , D2
  , include.mean = T , cond.dist=cond.dist )

if ( cond.dist=="norm" ) {
  m1<-G1@fit$par [ 1 ]
  s1<-sqrt ( G1@fit$par [ 2 ] + G1@fit$par [ 3 ]
    * ( D1 [ ts ] - G1@fit$par [ 1 ] ) ^ 2 + G1@fit$par [ 4 ]
    * G1@sigma.t [ ts ] ^ 2 )
  m2<-G2@fit$par [ 1 ]
  s2<-sqrt ( G2@fit$par [ 2 ] + G2@fit$par [ 3 ]
    * ( D2 [ ts ] - G2@fit$par [ 1 ] ) ^ 2 + G2@fit$par [ 4 ]
    * G2@sigma.t [ ts ] ^ 2 )

} else {
  m1<-G1@fit$par [ 1 ]
  s1<-sqrt ( G1@fit$par [ 2 ] + G1@fit$par [ 3 ]
    * ( D1 [ ts ] - G1@fit$par [ 1 ] ) ^ 2 + G1@fit$par [ 4 ]
    * G1@sigma.t [ ts ] ^ 2 )
  m2<-G2@fit$par [ 1 ]
  s2<-sqrt ( G2@fit$par [ 2 ] + G2@fit$par [ 3 ]
    * ( D2 [ ts ] - G2@fit$par [ 1 ] ) ^ 2 + G2@fit$par [ 4 ]
    * G2@sigma.t [ ts ] ^ 2 )
  df1<-G1@fit$par [ 5 ]
  df2<-G2@fit$par [ 5 ]
}

if ( cond.dist=="norm" ) {
  if ( copulaNr==1 | copulaNr==2 | copulaNr==7
    | copulaNr==8 | copulaNr==9 | copulaNr==10 ) {
    if ( copulaNr==1 | copulaNr==2 ) {
      if ( copulaNr==2 ) {
        cop<-co ( BiCopEst ( pnorm ( D1 , rep ( m1 , ts ) , G1@sigma.t )
          , pnorm ( D2 , rep ( m2 , ts ) , G2@sigma.t ) , copulaNr ) $par
          , df=BiCopEst ( pnorm ( D1 , rep ( m1 , ts ) , G1@sigma.t )
          , pnorm ( D2 , rep ( m2 , ts ) , G2@sigma.t ) , copulaNr ) $par2 )

```

```

    } else {
      cop<-co ( BiCopEst ( pnorm ( D1 , rep ( m1 , ts ) , G1@sigma . t )
        , pnorm ( D2 , rep ( m2 , ts ) , G2@sigma . t ) , copulaNr
        , method="itau" ) $par )
    }
  } else {
    cop<-co ( c ( BiCopEst ( pnorm ( D1 , rep ( m1 , ts ) , G1@sigma . t )
      , pnorm ( D2 , rep ( m2 , ts ) , G2@sigma . t ) , copulaNr ) $par
      , BiCopEst ( pnorm ( D1 , rep ( m1 , ts ) , G1@sigma . t )
        , pnorm ( D2 , rep ( m2 , ts ) , G2@sigma . t ) , copulaNr ) $par2 ) )
  }
} else {
  cop<-co ( BiCopEst ( pnorm ( D1 , rep ( m1 , ts ) , G1@sigma . t )
    , pnorm ( D2 , rep ( m2 , ts ) , G2@sigma . t ) , copulaNr ) $par )
}
} else {
  if ( copulaNr==1|copulaNr==2|copulaNr==7
    | copulaNr==8|copulaNr==9|copulaNr==10){
    if ( copulaNr==1|copulaNr==2){
      if ( copulaNr==2){
        cop<-co ( BiCopEst ( pt ( ( D1-m1 ) / G1@sigma . t , df1 )
          , pt ( ( D2-m2 ) / G2@sigma . t , df2 ) , copulaNr ) $par
          , df=BiCopEst ( pt ( ( D1-m1 ) / G1@sigma . t , df1 )
            , pt ( ( D2-m2 ) / G2@sigma . t , df2 ) , copulaNr ) $par2 )
        } else {
          cop<-co ( BiCopEst ( pt ( ( D1-m1 ) / G1@sigma . t , df1 )
            , pt ( ( D2-m2 ) / G2@sigma . t , df2 ) , copulaNr
            , method="itau" ) $par )
        }
      } else {
        cop<-co ( c ( BiCopEst ( pt ( ( D1-m1 ) / G1@sigma . t , df1 )
          , pt ( ( D2-m2 ) / G2@sigma . t , df2 ) , copulaNr ) $par
          , BiCopEst ( pt ( ( D1-m1 ) / G1@sigma . t , df1 )
            , pt ( ( D2-m2 ) / G2@sigma . t , df2 ) , copulaNr ) $par2 ) )
      }
    } else {
      cop<-co ( BiCopEst ( pt ( ( D1-m1 ) / G1@sigma . t , df1 )
        , pt ( ( D2-m2 ) / G2@sigma . t , df2 ) , copulaNr ) $par )
    }
  }
}

```

```

    }
  }
  m[i] <- uniroot(hh, interval=c(-15,2))$root
}
return(m)
}
l <- dim(data)[1] - 1000

m05 <- matrix(0, ncol=22, nrow=1)
#dauert sehr lange
#dazwischen immer wieder restarten und Pakete neu laden,
#sonst eventuell Fehlermeldungen
#Verwendung von SparseGrid bei divergenten Integralen,
#da dadurch Transformation

m05[,1] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 1, 0.05, 1, 1000, F)
m05[,2] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 2, 0.05, 1, 1000, T)
m05[,3] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 3, 0.05, 1, 1000, T)
m05[,4] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 4, 0.05, 1, 1000, F)
m05[,5] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 5, 0.05, 1, 1000, F)
m05[,6] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 7, 0.05, 1, 1000, T)
m05[,7] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 8, 0.05, 1, 1000, F)
m05[,8] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 9, 0.05, 1, 1000, T)
m05[,9] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 10, 0.05, 1, 1000, F)
m05[,10] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 13, 0.05, 1, 1000, F)
m05[,11] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 14, 0.05, 1, 1000, T)

m05[,12] <- VaRCopGARCH(data$ATX, data$DAX, "std", 1, 0.05, 1, 1000, T)
m05[,13] <- VaRCopGARCH(data$ATX, data$DAX, "std", 2, 0.05, 1, 1000, T)
m05[,14] <- VaRCopGARCH(data$ATX, data$DAX, "std", 3, 0.05, 1, 1000, T)
m05[,15] <- VaRCopGARCH(data$ATX, data$DAX, "std", 4, 0.05, 1, 1000, F)
m05[,16] <- VaRCopGARCH(data$ATX, data$DAX, "std", 5, 0.05, 1, 1000, F)
m05[,17] <- VaRCopGARCH(data$ATX, data$DAX, "std", 7, 0.05, 1, 1000, T)
m05[,18] <- VaRCopGARCH(data$ATX, data$DAX, "std", 8, 0.05, 1, 1000, F)
m05[,19] <- VaRCopGARCH(data$ATX, data$DAX, "std", 9, 0.05, 1, 1000, T)
m05[,20] <- VaRCopGARCH(data$ATX, data$DAX, "std", 10, 0.05, 1, 1000, F)
m05[,21] <- VaRCopGARCH(data$ATX, data$DAX, "std", 13, 0.05, 1, 1000, F)
m05[,22] <- VaRCopGARCH(data$ATX, data$DAX, "std", 14, 0.05, 1, 1000, T)

```

```
colnames(m05) <- c("GaussN", "tN", "ClayN", "GumbN", "FrankN", "BB1N",
                  "BB6N", "BB7N", "BB8N", "SurClayN", "SurGumbN", "GaussT",
                  "tT", "ClayT", "GumbT", "FrankT", "BB1T", "BB6T", "BB7T",
                  "BB8T", "SurClayT", "SurGumbT")
```

```
m01 <- matrix(0, ncol = 22, nrow = 1)
```

```
m01[,1] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 1, 0.01, 1, 1000, F)
m01[,2] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 2, 0.01, 1, 1000, T)
m01[,3] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 3, 0.01, 1, 1000, T)
m01[,4] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 4, 0.01, 1, 1000, F)
m01[,5] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 5, 0.01, 1, 1000, F)
m01[,6] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 7, 0.01, 1, 1000, T)
m01[,7] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 8, 0.01, 1, 1000, F)
m01[,8] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 9, 0.01, 1, 1000, T)
m01[,9] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 10, 0.01, 1, 1000, F)
m01[,10] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 13, 0.01, 1, 1000, F)
m01[,11] <- VaRCopGARCH(data$ATX, data$DAX, "norm", 14, 0.01, 1, 1000, T)
```

```
m01[,12] <- VaRCopGARCH(data$ATX, data$DAX, "std", 1, 0.01, 1, 1000, T)
m01[,13] <- VaRCopGARCH(data$ATX, data$DAX, "std", 2, 0.01, 1, 1000, T)
m01[,14] <- VaRCopGARCH(data$ATX, data$DAX, "std", 3, 0.01, 1, 1000, T)
m01[,15] <- VaRCopGARCH(data$ATX, data$DAX, "std", 4, 0.01, 1, 1000, F)
m01[,16] <- VaRCopGARCH(data$ATX, data$DAX, "std", 5, 0.01, 1, 1000, F)
m01[,17] <- VaRCopGARCH(data$ATX, data$DAX, "std", 7, 0.01, 1, 1000, T)
m01[,18] <- VaRCopGARCH(data$ATX, data$DAX, "std", 8, 0.01, 1, 1000, F)
m01[,19] <- VaRCopGARCH(data$ATX, data$DAX, "std", 9, 0.01, 1, 1000, T)
m01[,20] <- VaRCopGARCH(data$ATX, data$DAX, "std", 10, 0.01, 1, 1000, F)
m01[,21] <- VaRCopGARCH(data$ATX, data$DAX, "std", 13, 0.01, 1, 1000, F)
m01[,22] <- VaRCopGARCH(data$ATX, data$DAX, "std", 14, 0.01, 1, 1000, T)
```

```
colnames(m01) <- c("GaussN", "tN", "ClayN", "GumbN", "FrankN", "BB1N",
                  "BB6N", "BB7N", "BB8N", "SurClayN", "SurGumbN", "GaussT",
                  "tT", "ClayT", "GumbT", "FrankT", "BB1T", "BB6T", "BB7T",
                  "BB8T", "SurClayT", "SurGumbT")
```



```
VioCG0.05<-apply(m05,2,function(x){length(which(x>=portRet))})  
VioCG0.01<-apply(m01,2,function(x){length(which(x>=portRet))})
```

```
ExVi0.05<-round((length(data[,1])-1000)*0.05)  
ExVi0.01<-round((length(data[,1])-1000)*0.01)
```

```
mErrCG<-abs(ExVi0.05-VioCG0.05)+abs(ExVi0.01-VioCG0.01)
```

```
Ver0.05<-(abs(ExVi0.05-VioCG0.05)[1:11]  
+abs(ExVi0.05-VioCG0.05)[12:22])/2
```

```
Ver0.01<-(abs(ExVi0.01-VioCG0.01)[1:11]  
+abs(ExVi0.01-VioCG0.01)[12:22])/2
```

Verwendete R-packages

CDVine

Eike Christian Brechmann, Ulf Schepsmeier (2013). Modeling Dependence with C- and D-Vine Copulas: The R Package CDVine. *Journal of Statistical Software*, 52(3), 1-27. URL <http://www.jstatsoft.org/v52/i03/>.

copula

Marius Hofert, Ivan Kojadinovic, Martin Maechler and Jun Yan (2015). copula: Multivariate Dependence with Copulas. R package version 0.999-14 URL <http://CRAN.R-project.org/package=copula>

Jun Yan (2007). Enjoy the Joy of Copulas: With a Package copula. *Journal of Statistical Software*, 21(4), 1-21. URL <http://www.jstatsoft.org/v21/i04/>.

Ivan Kojadinovic, Jun Yan (2010). Modeling Multivariate Distributions with Continuous Margins Using the copula R Package. *Journal of Statistical Software*, 34(9), 1-20. URL <http://www.jstatsoft.org/v34/i09/>.

Marius Hofert, Martin Maechler (2011). Nested Archimedean Copulas Meet R: The naco-
pula Package. *Journal of Statistical Software*, 39(9), 1-20.
URL <http://www.jstatsoft.org/v39/i09/>.

docopulae

Andreas Rappold (2015). docopulae: Optimal Designs for Copula Models. R package version 0.3.2. <https://CRAN.R-project.org/package=docopulae>

e1071

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2015). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-7.
<https://CRAN.R-project.org/package=e1071>

fGarch

Diethelm Wuertz, Yohan Chalabi with contribution from Michal Miklovic, Chris Boudt, Pierre Chausse and others (2013). fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling. R package version 3010.82.
<https://CRAN.R-project.org/package=fGarch>

FinTS

Spencer Graves (2014). FinTS: Companion to Tsay (2005) Analysis of Financial Time Series. R package version 0.4-5. <https://CRAN.R-project.org/package=FinTS>

MTS

Ruey S. Tsay (2015). MTS: All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS) and Estimating Multivariate Volatility Models. R package version 0.33. <https://CRAN.R-project.org/package=MTS>

mvtnorm

Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, Torsten Hothorn (2016). mvtnorm: Multivariate Normal and t Distributions. R package version 1.0-5. URL <http://CRAN.R-project.org/package=mvtnorm>

Alan Genz, Frank Bretz (2009), *Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics*, Vol. 195., Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2

PerformanceAnalytics

Brian G. Peterson and Peter Carl (2014). PerformanceAnalytics: Econometric tools for performance and risk analysis. R package version 1.4.3541. <https://CRAN.R-project.org/package=PerformanceAnalytics>

rugarch

Alexios Ghalanos (2015). rugarch: Univariate GARCH models. R package version 1.3-6.

VineCopula

Ulf Schepsmeier, Jakob Stoeber, Eike Christian Brechmann, Benedikt Graeler, Thomas Nagler and Tobias Erhardt (2015). VineCopula: Statistical Inference of Vine Copulas. R package version 1.6-1. <https://CRAN.R-project.org/package=VineCopula>

Literatur

- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31 (3), 307–327.
- Demarta, S. & McNeil, A. J. (2005). The t copula and related copulas. *International Statistical Review/Revue Internationale de Statistique*, 111–129.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the Econometric Society*, 987–1007.
- Fricke, J. (2006). *Value-at-Risk Ansätze zur Abschätzung von Marktrisiken*. Deutscher Universitäts-Verlag.
- Hotta, L. K., Lucas, E. C. & Palaro, H. P. (2008). Estimation of VaR using copula and extreme value theory. *Multinational Finance Journal*, 12 (3/4), 205–218.
- Huang, J.-J., Lee, K.-J., Liang, H. & Lin, W.-F. (2009). Estimating value at risk of portfolio by conditional copula-GARCH method. *Insurance: Mathematics and Economics*, 45, 315–324.
- Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. Taylor & Francis.
- Jondeau, E. & Rockinger, M. (2006). The copula-garch model of conditional dependencies: An international stock market application. *Journal of international money and finance*, 25 (5), 827–853.
- Jorion, P. (2001). *Value at risk*. McGraw-Hill.
- Ljung, G. M. & Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika*, 65 (2), 297–303.
- Lu, X. F., Lai, K. K. & Liang, L. (2014). Portfolio value-at-risk estimation in energy futures markets with time-varying copula-GARCH model. *Annals of Operations Research*, 219 (1), 333–357.
- Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer.
- Ozun, A., Cifter, A. et al. (2007). Portfolio value-at-risk with time-varying copula: evidence from the Americas. *MPRA Paper, Munich Personal RePec Archive* (2711), 1–13.
- Palaro, H. P. & Hotta, L. K. (2006). Using Conditional Copula to Estimate Value at Risk. *Journal of Data Science*, 4, 93–115.
- Rockinger, M. & Jondeau, E. (2001). *Conditional dependency of financial series: an application of copulas*. (Banque de France Working Paper)
- Sklar, M. (1959). *Fonctions de répartition à n dimensions et leurs marges* (Bd. 8). (Publications de l'Institut de Statistique de L'Université de Paris)
- Wolke, T. (2008). *Risikomanagement*. Oldenbourg Verlag.