

Eine Client/Server–Architektur für Statistische Visualisierungen und Analyse von Finanzdaten

DISSERTATION

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von Dipl.–Wirt.–Math. Torsten Spillmann
geb. am 23.07.1974 in Haiger

eingereicht beim Fachbereich Mathematik
der Universität Siegen
Siegen 2004

urn:nbn:de:hbz:467-689

Die in dieser Dissertation verwendeten Copyrights, Warenzeichen und Logos
sind Eigentum ihres jeweiligen Inhabers.

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier.

Gutachter:

Prof. Dr. R.-D. Reiß
Dr. Michael Thomas

Datum der mündlichen Prüfung:

Donnerstag, den 28.10.2004

Vorwort

Die vorliegende Dissertation „Eine Client/Server-Architektur für Statistische Visualisierungen und Analyse von Finanzdaten“ gliedert sich in zwei Themengebiete. Im ersten Kapitel wurde eine spezielle Client/Server-Architektur entwickelt, um statistische Ergebnisse, insbesondere die des zweiten Kapitels „Finanzdatenanalyse“ zu visualisieren und zu überprüfen. In diesem Rahmen bot es sich an, die Architektur in drei Teile zu gliedern. Dabei wurde eine Three-Tier-Architektur eigens für statistische Visualisierungen entwickelt. Insbesondere wurden spezielle Schnittstellen für die Kommunikation, basierend auf der Middleware CORBA, erarbeitet. Für die statistische Visualisierungen wurde eine hocheffizient strukturierte Grafikkbibliothek bereitgestellt. Der in dieser Arbeit vorgestellte Prototyp kann vielseitig verwendet werden, z. B. wäre ein Einsatz im Bereich des E-Learnings denkbar.

Der Prototyp wurde in den Programmiersprachen C++ und JAVA implementiert. Durch die klare Struktur der Architektur ist es aber auch möglich, eine der Komponenten in einer anderen Programmiersprache zu entwickeln.

Im zweiten Teil liegt der Schwerpunkt auf der Finanzdatenanalyse mittels Studentischen Verteilungsmodellen. Die Analyse von Finanzdaten bekommt zur Zeit einen immer höheren Stellenwert. Durch diese Entwicklung ergeben sich interessante Herausforderungen. Banken und Versicherungen bauen bei der Analyse von Finanzdaten meist das Normalverteilungsmodell auf. Die Normalverteilung ist leicht zu handhaben, welches ihre Verwendung begründet.

Finanzdaten besitzen charakteristische Eigenschaften, sogenannte Stylized Facts. Diese sollten bei der Entwicklung von Modellen zur Analyse von Finanzdaten berücksichtigt werden. Das Normalverteilungsmodell hat seine Schwächen bei einzelnen Stylized Facts, deshalb wird in dieser Arbeit ein Modell mit einer Studentischen Verteilung benutzt. Durch die Weiterentwicklung des univariaten Studentischen Modells zu einem Studentischen Zeitreihenmodell wurden zusätzliche Stylized Facts in das Modell aufgenommen. Besonders hervorzuheben ist, dass dieses Modell die Volatilitätscluster der Finanzmärkte deutlich wiedergibt. Bei der Analyse wurde ein weiterer Stylized Fact, basierend auf den absoluten Renditen der Finanzwerte, gefunden. Es hat sich gezeigt, dass sich diese Eigenschaft empirisch bestätigt. Aufbauend auf diesen Erkenntnissen wurde ein entsprechender Simulationsalgorithmus erstellt.

II

Für die Portfolioanalyse muss das univariate Modell auf ein multivariates Modell erweitert werden. Zunächst wird ein multivariates Studentisches Verteilungsmodell untersucht. Ein Nachteil dieses Modells ist die mangelnde Flexibilität. Um diesen Nachteil zu umgehen, wurde ein Copula-Modell eingeführt. Hierbei ist es möglich, die Modellierung in zwei Schritte zu unterteilen. Im ersten Schritt erfolgt eine Analyse der Randverteilung, wobei auf die vorgestellten univariaten Modelle zurückgegriffen werden kann, im zweiten Schritt wird die Abhängigkeit betrachtet. Um die Möglichkeiten des Copulas-Modells optimal zu nutzen, war eine intensive Behandlung des Themengebietes notwendig. Neben theoretischen Resultaten wurden Schätzer und Simulationsalgorithmen angegeben. Besonders wurde sich mit der Fragestellung „Welche Copula gibt die Abhängigkeit am besten wieder?“ auseinander gesetzt. Neben dem üblichen Auswahlverfahren über die empirische Copula wurde eine weiteres Auswahlverfahren basierend auf den Tail-Dependence-Funktion vorgestellt.

Inhaltsverzeichnis

Vorwort	I
Inhaltsverzeichnis	II
Abbildungsverzeichnis	VI
Tabellenverzeichnis	X
Listings	XI
Abkürzungsverzeichnis	XIII
Mathematisches Symbolverzeichnis	XVI
1 Client/Server–Architektur	1
1.1 Einleitung	1
1.2 Statistische Visualisierungen	5
1.2.1 Statistische Plots	5
1.2.2 Beispiele	5
1.2.3 Farben und Linienstile	8
1.2.4 Ausgabe von statistischen Plots	9
1.2.4.1 Definition eines Grafikobjektes	9
1.2.4.2 Weltraum und Weltkoordinatensystem	10
1.2.4.3 Bildraum und Bildkoordinatensystem	11
1.2.4.4 Window–Viewport–Transformation	11
1.2.4.5 Projektion	13
1.2.4.6 Umsetzung in der Architektur	14
1.3 High–Level–Design der Architektur	15
1.3.1 Display–Server	16
1.3.2 Objekt–Server	17
1.3.3 Client	18
1.3.4 Datensicherheit	19
1.4 Auswahl der Middleware für die Umsetzung	20

1.4.1	Common Objekt Request Broker Architecture (CORBA)	21
1.4.2	Remote Method Invocation (RMI)	23
1.4.3	COM + DCOM	25
1.4.3.1	DirectX	27
1.4.4	Eigene Middleware	30
1.4.5	OpenGL	30
1.4.6	XploRe	31
1.4.7	Resümee	31
1.5	Umsetzung spezieller Aufgaben	33
1.5.1	Details zum Prototypen	33
1.5.2	Ausgabe eines Grafikobjektes	34
1.5.3	Erweiterung des Objekt-Servers um einen statistischen Plot	36
1.5.4	Manager zu Verwaltung: Beispiel Ausgabefenster	38
1.5.5	Events	40
1.5.6	Callback	41
1.5.7	Umsetzung von Farben und Linienstilen	43
2	Finanzdatenanalyse	45
2.1	Einführung	45
2.2	Allgemeine Modellbildung	47
2.2.1	Stylized Facts	47
2.3	Moderne Risikoanalyse	55
2.3.0.1	Gewinn- und Verlustverteilung	56
2.3.0.2	Risikomaße	58
2.3.0.3	Value-at-Risk	59
2.3.0.4	Tail Conditional Expectation	60
2.3.0.5	Expected Shortfall	60
2.3.0.6	Empirische Schätzer für die Risikomaße	61
2.4	Studentsche Modellierung von Finanzdaten	64
2.4.1	Univariate Modellierung	64
2.4.2	Univariater Fall	65
2.4.2.1	Eigenschaften	68
2.4.2.2	Schätzer	69
2.4.2.3	Datenanalyse	70
2.4.3	Studentsche Zeitreihe	73
2.4.3.1	Modellbildung der Zeitreihe	73

2.4.3.2	Untersuchung der absoluten Renditen	80
2.4.3.3	Simulationsalgorithmus	91
2.4.3.4	Schätzung der Parameter	95
2.4.4	Dichte der multivariaten Studentschen Verteilung	97
2.4.5	Risikomaß, Abhängigkeitsmaß und Risikominimierung . .	98
2.4.5.1	Pearson's ρ	99
2.4.5.2	Spearman's ρ	100
2.4.5.3	Kendall's τ	100
2.4.5.4	Tail-Dependence χ	104
2.4.5.5	Tail-Dependence $\bar{\chi}$	105
2.4.6	Multivariate Studentsche Dichte	107
2.4.6.1	Schätzer	108
2.4.6.2	Datenanalyse	113
2.4.7	Copulas	119
2.4.7.1	Copulas in der Finanzmathematik	119
2.4.7.2	Einführung in Copulas	120
2.4.8	Empirische Copula (Deheuvels Copula)	125
2.4.9	Copuladichte	126
2.4.10	Empirische Copulahäufigkeit	128
2.4.11	Schätzer für Copulas	128
2.4.11.1	Maximum-Likelihood-Methode	128
2.4.12	Weitere Schätzer	131
2.4.12.1	IFM-Methode	131
2.4.12.2	CML-Methode	132
2.4.12.3	Beispiele für die IFM- und CML-Methode . .	132
2.4.13	Erzeugen von Realisationen einer Copula	133
2.4.13.1	Allgemeine Vorgehensweise	133
2.4.13.2	Spezielle Algorithmen zur Stichprobenerzeugung	134
2.4.14	Auswahl einer Copula	138
2.4.14.1	Auswahl der elliptischen Copulas	139
2.4.14.2	Test auf Gauss-Copula	140
2.4.14.3	Auswahl der Archimedischen Copulas	142
2.4.14.4	Auswahl der Copula über die empirische Copula	143
2.4.14.5	Auswahl über die Tail-Dependence-Funktion .	144
2.4.15	Datenanalyse mit Finanzdaten	150

2.4.16	Abschlussbemerkung zum finanzmathematischen Teil . . .	154
A	Ergänzungen zur Client/Server–Architektur	157
A.1	Mathematische Grundlagen	157
A.1.1	Window–Viewport–Transformation	157
A.1.2	Rotation	159
A.1.3	Projektion	160
A.2	Entwurfsmuster oder Design Patterns	163
A.2.1	Factory oder Manager	163
A.2.2	Prototyp	164
A.2.3	Delegation	165
A.3	CORBA–Schnittstelle des Prototypen	166
A.4	Technische Details zum Prototypen	175
B	Finanzmathematischer Teil	177
B.1	Mathematische Grundlagen	177
B.1.1	Allgemeine mathematische Grundlagen	177
B.1.1.1	Sätze und Beweise	179
B.1.1.2	Grundlagen für Zeitreihen	182
B.1.2	Schätzer und Test's	183
B.1.2.1	EM–Schätzer (EM–Algorithmus)	184
B.1.3	Spezielle bivariate Copulas und deren Dichten	185
B.1.4	Simulationsalgorithmus	210
B.1.5	Statistische Plots: Parameter der Copulas	215
B.1.6	Autokorrelationsfunktion der absoluten Renditen	216
	Literaturverzeichnis	221
	Stichwortverzeichnis	229

Abbildungsverzeichnis

1.1	Grafiken verschiedener Wahlergebnisse	1
1.2	Statistische Plots	2
1.3	Aufbau der Client/Server–Architektur	3
1.4	Boxplot, Balkendiagramm, Kreisdiagramm	6
1.5	Histogramm, Kerndichte, Scatterplot	7
1.6	Scatterplot, Zeitreihe, bivariate Kerndichte	8
1.7	Verschiedene Linienstile	9
1.8	3D–Welt, 2D–Welt	10
1.9	2D–Bildraum	11
1.10	Window–Viewport–Transformation	12
1.11	Ablauf vom 3D–Raum zum Display	13
1.12	Three–Tier–Architektur	15
1.13	CORBA Systemarchitektur	22
1.14	RMI Systemarchitektur	26
1.15	DCOM Systemarchitektur	27
1.16	Aufbau des Prototypen	33
1.17	Ablaufschema der Ausgabe	35
1.18	Ausgabe eines 3D–Plots	36
1.19	Klassenstruktur der Grafikobjekte	37
1.20	Manager im Objekt–Server zur Verwaltung der Display–Server . .	39
1.21	Events	41
1.22	Ablauf eines Callbacks	42
2.1	Scatterplot des Dax	48
2.2	Zeitreihe des Dax, Autokorrelation des Dax	49
2.3	Scatterplot der Rendite, Autokorrelation der Rendite	50
2.4	Normalverteilung: Kerndichte und Flanken	52

2.5	Kerndichte mit Erwartungswert	53
2.6	Scatterplot zweier Aktien	55
2.7	Kurswerte und Renditen des Dax	71
2.8	Empirische Kerndichte und Normalverteilung	72
2.9	Empirische Kerndichte und Studentsche Verteilung	72
2.10	Q-Q-Plot: Normalverteilung, Studentschen Verteilung	73
2.11	Kerndichte: Dax-Rendite, Normal- und Studentschen Verteilung	74
2.12	Erzeugung der s_1, \dots, s_q	76
2.13	Studentschen-Zeitreihe; Simulationskern	77
2.14	Autokorrelationsfunktion (ACF) der Renditen R_t	81
2.15	Autokorrelationsfunktion (ACF) der absoluten Renditen $ R_t $. . .	82
2.16	Erwartungswert von $Y_0^{-1/2}$ und dessen Quadrat	84
2.17	Erwartungswert von $E(Y_0^{-1})$ und dessen Quadrat	89
2.18	Scatterplot einer Simulationszeitreihe	91
2.19	Beta-Kern	93
2.20	Gauss-Kern	94
2.21	Autokorrelationsfunktion der absoluten Renditen des Dax	95
2.22	Sprung-Kern	97
2.23	Contourplot, 3D-Scatterplot	108
2.24	Kerndichte der Renditen, Studentsche Dichte	114
2.25	Kerndichte des Studentschen Datensatzes	115
2.26	Differenz der beiden Kerndichten	116
2.27	Kerndichte der Renditen, Studentsche Dichte II	117
2.28	Differenz der beiden Kerndichten (Simulation – Henkel / Linde) .	118
2.29	Differenz der beiden Fehler	119
2.30	Fréchet-Hoeffding Schranken bei einer Copula	123
2.31	Simulation Gauss-Copula: $\rho = 0, \rho = 0.5, \rho = 0.9$	135
2.32	Simulation Student-Copula: $\alpha = 5, \rho = 0, \rho = 0.5, \rho = 0.9$	136
2.33	Simulation Cook-Johnson-Copula: $\delta = 2, \delta = 5, \delta = 10$	136
2.34	Simulation Morgenstern-Copula: $\delta = 0, \delta = 0.5, \delta = 1$	137
2.35	Simulation Gumbel-Copula: $\delta = 3, \delta = 10, \delta = 100$	138
2.36	Flanken der Copula	145
2.37	Empirische Copula der Bayer- und Commerzbank-Renditen . . .	151
2.38	Contourplot der Copulas	152
2.39	Tail-Dependence-Funktion Gauss/Empirische Copula	153

2.40	Tail–Dependence–Funktion Clayton/Empirische Copula	154
2.41	Tail–Dependence–Funktion Farlie/Empirische Copula	154
2.42	Banken, Gauss–Copula, Clayton–Copula	155
A.1	Zentralprojektion	161
A.2	Projektion: x–Wert	162
A.3	Design–Pattern: Factory Methode	163
A.4	Design–Pattern: Prototype	164
A.5	Struktur der Plotobjekte	165
A.6	Screenshot des Prototypen	176
B.1	Gauss–Copula: Contourplot, 3D–Plot	186
B.2	Gauss–Copuladichte: Contourplot, 3D–Plot	186
B.3	Student–Copuladichte: Contourplot, 3D–Plot	188
B.4	Logistic–Extreme–Copula: Contourplot, 3D–Plot	189
B.5	Logistic–Extreme–Copuladichte: Contourplot, 3D–Plot	190
B.6	Plackett Copula: Contourplot, 3D–Plot	192
B.7	Plackett–Copuladichte: Contourplot, 3D–Plot	192
B.8	Clayton–Copula: Contourplot, 3D–Plot	194
B.9	Clayton–Copuladichte: Contourplot, 3D–Plot	194
B.10	Frechet Copula: Contourplot, 3D–Plot	195
B.11	Mardia Copula: Contourplot, 3D–Plot	196
B.12	Cuadras–Auge– Copula: Contourplot, 3D–Plot	197
B.13	Bivariate Gumbel–Logistic–Copula: Contourplot, 3D–Plot	198
B.14	FGM–Copula: Contourplot, 3D–Plot	198
B.15	Dichte der FGM–Copula: Contourplot, 3D–Plot	199
B.16	Ali–Mikhail–Haq–Copula: Contourplot, 3D–Plot	199
B.17	Ali–Mikhail–Haq–Copuladichte: Contourplot, 3D–Plot	200
B.18	Frank–Copula: Contourplot, 3D–Plot	200
B.19	Marshall–Olkin–Copula: Contourplot, 3D–Plot	201
B.20	Marshall–Olkin–Copuladichte: Contourplot, 3D–Plot	202
B.21	Gumbel–Exponential–Copula: Contourplot, 3D–Plot	203
B.22	Gumbel–Exponential–Copuladichte: Contourplot, 3D–Plot	204
B.23	Gumbel–Copula: Contourplot, 3D–Plot	204
B.24	Galambos–Copula: Contourplot, 3D–Plot	205
B.25	Huessler–Reiss–Copula: Contourplot, 3D–Plot	205

B.26 Joe-Copula: Contourplot, 3D-Plot	206
B.27 BB1-Copula: Contourplot, 3D-Plot	206
B.28 BB2-Copula: Contourplot, 3D-Plot	207
B.29 BB3-Copula: Contourplot, 3D-Plot	207
B.30 BB4-Copula: Contourplot, 3D-Plot	208
B.31 BB5-Copula: Contourplot, 3D-Plot	208
B.32 BB6-Copula: Contourplot, 3D-Plot	209
B.33 BB7-Copula: Contourplot, 3D-Plot	210

Tabellenverzeichnis

1.1	CORBA–Services I	24
1.2	CORBA–Services II	25
1.3	Gegenüberstellung von CORBA, DCOM/COM und RMI I	28
1.4	Gegenüberstellung von CORBA, DCOM/COM und RMI II	29
1.5	Vor– und Nachteile einer eigenen Middleware	30
1.6	Methoden der Grafikobjekte	34
1.7	Events	41
2.1	Stylized Facts	56
2.2	Geschätzte Parameter für die beiden Modelle	71
2.3	Schätzung von α und σ der Zeitreihe mit MLE–Student	96
2.4	Geschätzte Parameter für die Commerzbank– und Degussa–Aktie	113
2.5	Geschätzte Parameter für die Henkel– bzw. Linde–Aktien	116
2.6	Ergebnisse der Simulation	148
2.7	Parameter der Copulas für die Bayer/Commerzbank–Renditen	153
A.1	Schritte der Transformation	161
B.1	Parameter der Copulas für die statistischen Plots	215
B.2	Autokorrelationsfunktionen der absoluten Renditen I	216
B.3	Autokorrelationsfunktionen der absoluten Renditen II	217

Listings

1.1	Beispielschnittstelle für eine Grafikobjekt	37
1.2	Aufruf zur Registrierung eines Prototypen	38
1.3	Event-Methode	40
1.4	Interface TColor	43
1.5	Interface TLine	44
A.1	Schnittstelle der Client/Server-Architektur	166
B.1	Algorithmus der Studentschen Zeitreihe	210

Abkürzungsverzeichnis

ACF	Autokorrelationsfunktion
API	Application Programming Interface
CDR	Common Data Representation
CML	Canonical Maximum Likelihood
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
Cov	Covarianz
CVaR	Conditional Value at Risk
Dax	Deutscher Aktien Index
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DII	Dynamic Invocation Interface
DSI	Dynamic Skeleton Interface
EM	Expectation–Maximization
ES	Expected Shortfall
E–Schritt	Erwartungsschritt
GUID	Global Unique Identifier
HTTP	Hyper Text Transfer Protocol
IDL	Interface Definition Language
IFM	Method of Inference Functions of Margins
iid	Independent Identical Distributed
IIOP	Internet Inter–ORB Protocol
IQR	Interquantil Range
JVM	Java Virtual Machine
MDA	Maximum Domain of Attraction
MICO	Mico is CORBA

midl	Microsoft Interface Definition Language
ML	Maximum–Likelihood
MLE	Maximum Likelihood Estimator
M–Schritt	Maximierungsschritt
MTS	Microsoft Transaction Server
OLE32	Object Linking and Embedding
OMG	Object Management Group
ORB	Object Request Broker
P+L	Profit+Loss
QQ–Plot	Quantil–Quantil Plot
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SCM	Service Control Manager
SCM	Service Control Manager
SSI	Static Skeleton Interface
SSL	Secure Sockets Layer
SQL	Structured Query Language
TCE	Tail Conditional Expectation
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modelling Language
VaR	Value at Risk
VF	Verteilungsfunktion
VM	Virtual Machine

Mathematisches Symbolverzeichnis

\mathbb{N}	natürlichen Zahlen	
\mathbb{Z}	ganzen Zahlen	
\mathbb{R}	reellen Zahlen	
$\bar{\mathbb{R}}$	reellen Zahlen mit $-\infty$ und ∞	
\mathbf{I}	Einheitsintervall $[0, 1]$	
$[]$	Gaussklammer	
$\det(A)$	Determinante von A	
F	Verteilungsfunktion	
F^{-1}	Quantilfunktion	S. 59
\hat{F}	empirische Verteilungsfunktion	
\hat{F}^{-1}	empirische Quantilfunktion	
$F^{[u]}$	Exzedenten-Funktion	S. 60
$F^{(u)}$	Exzess-Funktion	S. 60
$E(X)$	Erwartungswert der ZV X	
$Var(X)$	Varianz der ZV X	
$Cov(X, Y)$	Kovarianz der ZV X und Y	
$Corr(X, Y)$	Korrelation der ZV X und Y	
ACF	Autokorrelationsfunktion	S. 81
$\delta(X)$	Kurtosis	S. 51
$\gamma(X)$	Schiefe	S. 54
L_t	Verlustvariable	S. 56
V_t	Marktpreis	S. 56
$\hat{\mu}$	Stichproben-Mittelwert	
$\hat{\sigma}^2$	Stichproben-Varianz	

U	Gleichverteilung auf $[0, 1]$	
$t_{\alpha, \mu, \sigma}$	univariate Studentsche Verteilung	S. 65
$f_{\alpha, \mu, \Sigma}^{S^d}$	d -dim. Dichte der Studentschen Verteilung	S. 108
$t_p(\mu, \Sigma, \alpha)$	p -dim. Studentsche Verteilung	S. 108
Φ_{μ, σ^2}	VF der univ. Normalverteilung	
φ_{μ, σ^2}	Dichte der uni. Normalverteilung	S. 66
Φ^{-1}	Quantilfunktion der univ. Normalverteilung	
ϕ	Dichte der Standard-Normalverteilung	
$N(\mu, \Sigma)$	multivariate Normalverteilung	
$\Gamma(\alpha, \beta)$	Gamma-Verteilung	S. 75
$f_{r, s}^G$	Dichte der Gamma-Verteilung	S. 65
$\Gamma(\lambda)$	Gamma-Funktion	S. 65
$\Psi(s)$	Digamma-Funktion	S. 69
S_t	ZV: Kurs des Finanzwertes	S. 55
s_t	Realisationen der Kurse	
\tilde{R}_t	arithmetische Rendite	S. 48
R_t	logarithmische Rendite	S. 48
VaR	Value-at-Risk	
C	Copula	
\bar{C}	Survival-Copula	
C'	Sub-Copula	
V_H	H-Volumen	
\bar{C}	empirische Copula	
\bar{c}	empirische Copula-Dichte	
C^G	Gauss-Copula	S. 129
c^G	Dichte der Gauss-Copula	S. 129
ρ	Pearson's rho	S. 99
ρ_S	Spearman's rho	S. 100
τ	Kendall's tau	S. 100
$\chi, \bar{\chi}$	Tail-Dependence-Funktionen	S. 104
K^β	Beta-Kern	S. 92
K^{Gauss}	Gauss-Kern	S. 92
K^S	Sprung-Kern	S. 93

Kapitel 1

Client/Server–Architektur

1.1 Einleitung

Statistische Visualisierungen haben im heutigen Informationszeitalter einen sehr hohen Stellenwert. Man findet sie im Fernsehen, in den Printmedien und auch im Internet. Aus ihnen kann der Betrachter statistische Ergebnisse ablesen, wie z. B. die Sitzverteilung der letzten Wahl. Ebenfalls sind sie in Bereichen wie Finanz-

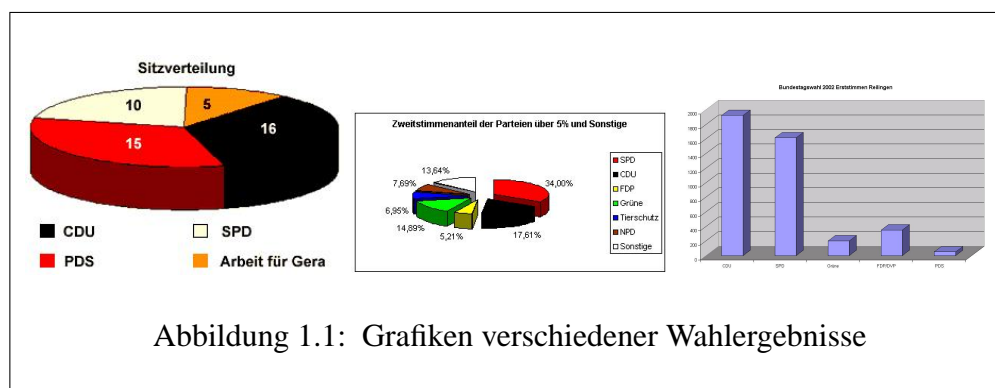


Abbildung 1.1: Grafiken verschiedener Wahlergebnisse

sektor, Meteorologie, Medizin und Physik¹ zu finden. Diese Liste kann noch um viele Gebiete erweitert werden.

Die Grundlage dieser Grafiken bilden immer mathematische oder besser statistische Algorithmen (siehe Nagel et. al. [64], Kap 1.3). Durch die Anwendung

¹Grafik: Entnommen aus: voyager.physik2.uni-rostock.de, www.mediinfo.de, www.umweltdaten.nuernberg.de

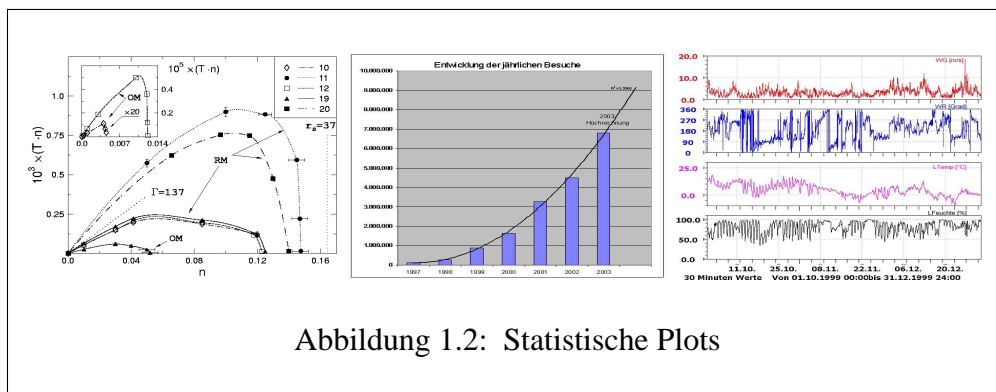


Abbildung 1.2: Statistische Plots

dieser Algorithmen können die Visualisierungen berechnet werden. Je nach Art des Algorithmus und Anzahl der Daten kann dies eine sehr komplexe Aufgabe sein.

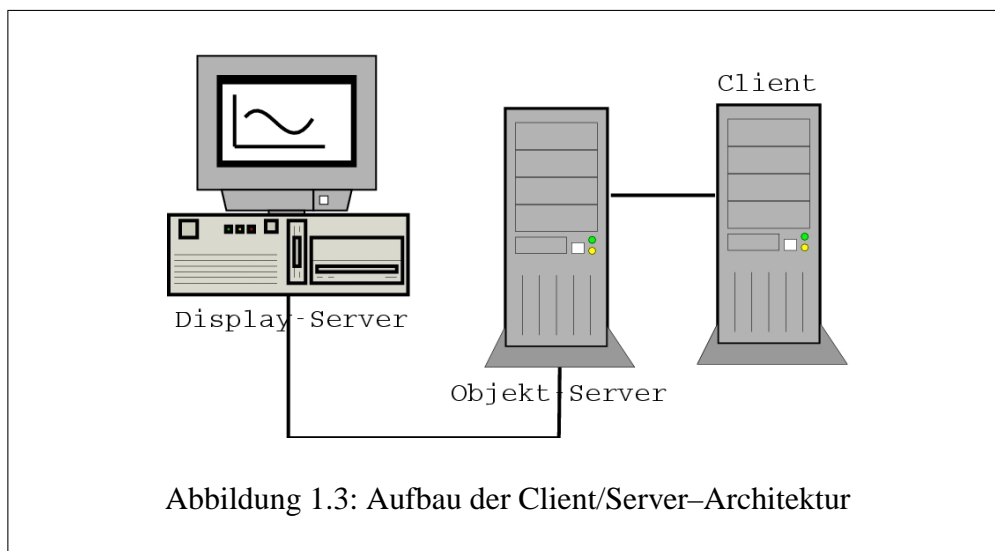
Im ersten Teil der Arbeit wird eine Client/Server-Architektur für die Ausgabe statistischer Visualisierungen vorgestellt. Sie soll folgende Aufgaben erfüllen:

- verschiedene statistische Visualisierungen werden für den Betrachter ausgegeben,
- die Architektur soll einem Programmierer die Möglichkeit bieten, sie leicht um eigene statistische Plots zu erweitern. Es soll dem Programmierer dazu eine universelle Grafikbibliothek zur Verfügung gestellt werden, mit der er leicht statistische Plots visualisieren kann,
- die Interaktion durch einen Benutzer soll möglich sein,
- die Architektur soll auf möglichst vielen Plattformen lauffähig sein, d. h. sie soll nicht an ein bestimmtes Betriebssystem oder eine bestimmte Hardware gebunden sein.

Die Client/Server-Architektur soll in vielen Bereichen — wie z. B. Web-Anwendungen oder auch im Bereich E-Learning — zum Einsatz kommen. Ebenso kann sie zum Beispiel in lokalen Intranets von Firmen oder Institutionen zur Visualisierung von Geschäftsdaten eingesetzt werden. Zusammengefasst kann sie also in sämtlichen Bereichen zum Einsatz kommen, wo Datenmengen ausgewertet und anschaulich dargestellt werden sollen. Bei ihrem Einsatz ermöglicht der

hier vorgestellte Prototyp der Client/Server–Architektur die Betrachtung von statistischen Visualisierungen, ohne das Zusatzprogramme notwendig sind; d. h. für die Visualisierung wird auf schon installierte Komponenten wie z. B. dem Web–Browser zurückgegriffen.

Da statistischen Visualisierungen oft große Datenmengen zu Grunde liegen, wird die Architektur auch an diese Bedürfnisse angepasst. Die Berechnungen für die Grafiken müssen in angemessener Geschwindigkeit erfolgen. Außerdem dürfen beim Einsatz im Netz nur die nötigsten Daten für die Ausgabe transportiert werden. Dies ist der Aspekt der Datensicherheit. Um dieses zu erreichen, wurde die Client/Server–Architektur in drei Komponenten unterteilt, den Objekt–Server, den Display–Server und einem Client. Dies ist besonders dann von Vorteil, wenn eine große Datenmenge verarbeitet und visualisiert werden soll. Der Aufbau



der Architektur und die Aufgaben der Komponenten werden im Abschnitt High–Level–Design 1.3 beschrieben. Im anschließenden Abschnitt geht es um die Umsetzung des High–Level–Designs, das heißt es wird erklärt und diskutiert, warum welche Middleware² eingesetzt wird und welche Programmiersprachen für die Umsetzung benutzt werden. Im nächsten Abschnitt wird die Umsetzung spezieller Aufgaben in der Implementierung beschrieben, dabei werden Programmierstruk-

²Middleware ist eine Software, die es ermöglicht, dass verschiedenen Komponenten miteinander kommunizieren können, wenn sie z. B. verteilt und in verschiedenen Programmiersprachen erstellt worden sind.

turen vorgestellt und erläutert. Der vorgestellte Prototyp findet seine Verwendung im zweiten Teil der Arbeit: bei der Analyse der Finanzdaten.

Es wird zunächst auf statistische Visualisierungen, genauer deren Aufgaben und Aufbau, eingegangen.

1.2 Statistische Visualisierungen

In diesem Abschnitt werden alle für die Ausgabe relevanten Faktoren vorgestellt und näher beschrieben. Den Ausgangspunkt der statistischen Visualisierungen³ bilden die statistischen Plots. Sie werden im Folgenden eingeführt und anschließend an einigen Beispielen dargestellt. Danach wird die Frage vertieft, wie es zu einer Ausgabe kommt, und welche Voraussetzungen dafür erfüllt sein müssen.

1.2.1 Statistische Plots

Unter einem *statistischen Plot* wird in dieser Dissertation eine Grafik verstanden, die statistische Eigenschaften, welche durch einen mathematischen bzw. statistischen Algorithmus berechnet wurden, visualisiert. Unter statistischen Eigenschaften werden Funktionale einer Verteilung verstanden, die parametrisch oder nicht parametrisch sein können. Den Berechnungen liegt ein Datensatz oder eine Funktion die eine statistische Eigenschaft beschreibt, zu Grunde, worauf der Algorithmus angewandt wird. Mit der Wahl des Algorithmus und den freien Parametern können die Berechnungen und somit die Ausgabe manipuliert werden (siehe dazu Nagel et al. [64]). Die Eigenschaften, die ein Betrachter vermittelt bekommt, sind also stets subjektiv und abhängig von den verwendeten statistischen Plots und den zu Grunde liegenden Algorithmen. Um eine objektive Datenanalyse zu gewährleisten ist es notwendig, alle dem Plot zu Grunde liegenden Berechnungen zu kennen. Es können zum Beispiel durch diese Plots neu entwickelte, statistische Modelle anschaulich dargestellt, oder auch Schätzer⁴ für diese Modelle verifiziert werden.

1.2.2 Beispiele

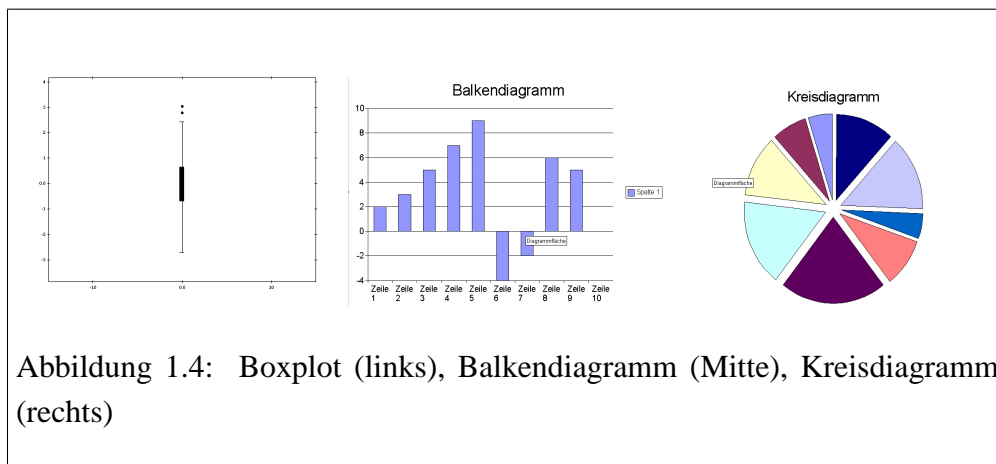
Mit den unterschiedlichen Plots, wie Boxplot, Balkendiagramm⁵ (Nagel et al. [64], Seite 13), Kreisdiagramm (Nagel et al. [64], Seite 12), Scatterplot, Histo-

³Die Client/Server-Architektur kann auch für nicht statistische Visualisierungen genutzt werden. Es muß jedoch überprüft werden, ob die Grafikbibliothek die neugestellten Anforderungen unterstützt.

⁴Ein Schätzer ist ein Wert, mit dem über einen mathematischen Algorithmus ein Parameter des Modells gefunden wird. (siehe Anhang: Schätzfunktion, Schätzstatistik, S. 184).

⁵Es wird auch als Säulendiagramm bezeichnet.

gramm⁶ oder Kerndichte, werden spezielle Eigenschaften eines Datensatzes wiedergegeben⁷. Ein Boxplot (Abb. 1.4 (links)) ist wie folgt definiert:



Definition 1 (Boxplot). Ein Boxplot ist ein Instrument der graphischen Datenanalyse, welches gleichzeitig ein Lokationsmaß, den Median, als Streuungsmaß den Interquartil Range (IQR), mögliche Ausreißer oberhalb einer Schranke⁸ von $x_{[\frac{3n}{4}]:n} + 1.5IQR$ bzw. unterhalb einer Schranke von $x_{[\frac{n}{4}]:n} - 1.5IQR$ sowie Hinweise auf Symmetrie oder Schiefe angibt, Falk et. al. [27].

Das Balkendiagramm (Abb. 1.4 Mitte) stellt ebenso wie das Kreisdiagramm (Abb. 1.4 rechts) das Verhältnis verschiedener Daten⁹ zueinander dar. Ein Histogramm (Abb. 1.5 links) gibt die Häufigkeiten für das Auftreten von bestimmten Elementen in einem Datensatz wieder. Bei einer großen Stichprobenanzahl kann die dem diskreten Datensatz zu Grunde liegende Verteilung erkannt werden (siehe Nagel et al. [64], Kapitel 2.3). Diese Daten müssen zumindest eine größere Anzahl ordinaler Merkmale¹⁰ besitzen, damit es zu einer sinnvollen Darstellung kommen kann. Sonst gäbe es nur wenige Balken im Histogramm, die somit nur eine geringe Aussage über die Verteilung liefern. In diesem Fall ist ein Balken- oder Kreisdiagramm sinnvoller.

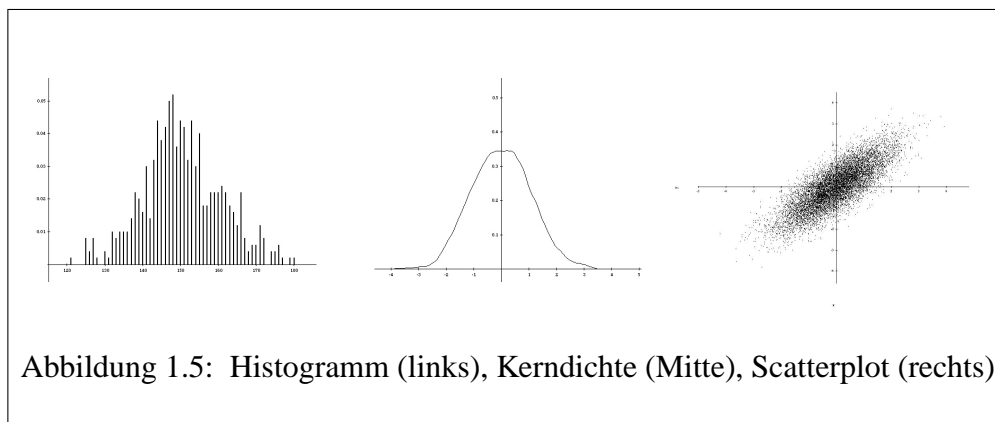
⁶Die Definition des Histogramms ist zu finden in Nagel et al. [64], Kapitel 2.2.2 .

⁷Weitere Plots sind Nagel et al. [64] zu entnehmen.

⁸Mit $[x]$ wird die größte ganze Zahl bezeichnet, die kleiner x ist.

⁹Gruppen von Punkten

¹⁰Sie müssen in verschiedene Gruppen eingeteilt werden können, wobei eine Ordnung zu Grunde liegen muss, aber der Abstand zwischen den einzelnen Elementen muss nicht messbar sein.



Wenn den Daten eine stetige Verteilung zu Grunde liegt, bildet die Kerndichte einen nichtparametrischen Schätzer für die Dichte. Die Kerndichte (Abb. 1.5 Mitte) ist eine Funktion, die abhängig von der Wahl des Kerns und der zur Berechnung benutzten Bandbreite ist, und wird definiert als:

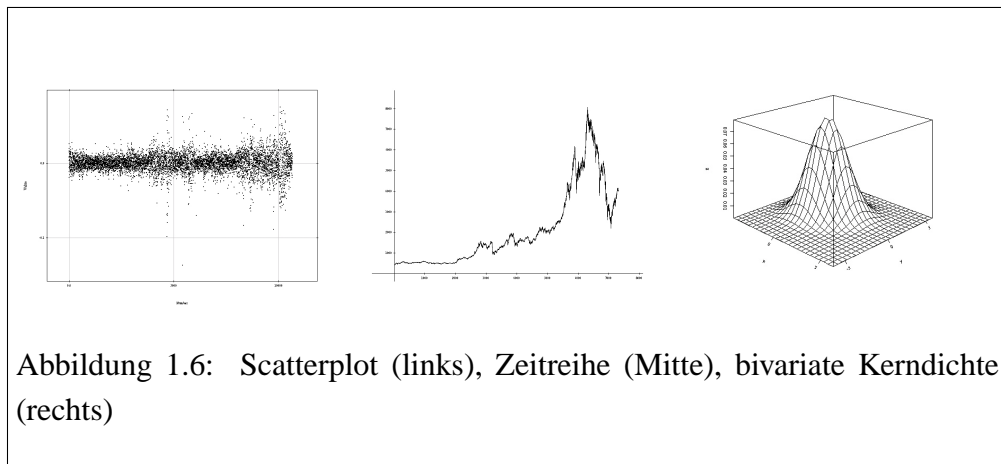
Definition 2 (Kern-Dichteschätzer). *Es sei $k : \mathbb{R} \rightarrow \mathbb{R}$ eine Funktion mit $\int k(x)dx = 1$, der Kern. Die Abbildung*

$$\hat{f}_n(t) := \frac{1}{nh} \sum_{i=1}^n k\left(\frac{t - x_i}{h}\right) = \int k\left(\frac{t - x}{h}\right) F_n(dx), \quad t \in \mathbb{R}, \quad (1.1)$$

heißt *univariater Kern-Dichteschätzer mit Fensterbreite (oder Bandbreite) $h > 0$, Falk et. al. [27].*

Die Daten werden dabei nicht einzeln in verschiedene Gruppen sortiert, sondern fließend in einer Umgebung betrachtet. Möchte man die Werte eines Datensatzes betrachten, empfiehlt es sich einen Scatterplot¹¹ (Abb. 1.6 (links)) zu verwenden. Der Scatterplot gibt zweidimensionale Punkte (x, y) in einem Koordinatensystem aus.

¹¹auch Punktediagramm genannt



Definition 3. Scatterplot¹²

Unter einem Scatterplot zu den Daten $(x_1, y_1), \dots, (x_n, y_n)$ im \mathbb{R}^2 versteht man einen Plot dieser Punkte im x, y -Koordinatensystem.

Sind $(x_1, y_1), \dots, (x_n, y_n)$ Realisierungen von n unabhängigen Wiederholungen des Zufallsvektors (X, Y) , so bewirkt ein betragsmäßig großer Korrelationskoeffizient¹³ von X und Y eine gewisse Linearität im Scatterplot, Falk et. al. [27]. Der Scatterplot wird häufig für die Visualisierung von Zeitreihen, zum Beispiel

bei Finanzdaten, wie einem Aktienkurs (Abb. 1.6 Mitte), eingesetzt.

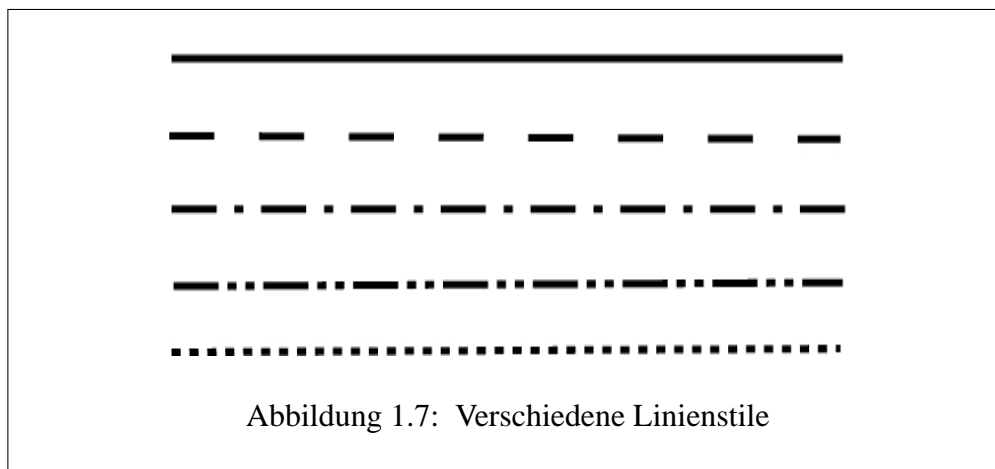
1.2.3 Farben und Linienstile

Um die statistischen Plots besser darstellen zu können, ist es notwendig, Farben zu verwenden. Durch Farben können mehrere verschiedene Plots, die in einem Fenster ausgegeben werden, unterschieden werden. Außerdem kann durch Farben und Graustufen die räumliche Darstellung verbessert werden. Eine andere Möglichkeit zur Unterscheidung zweier Plots in einem Fenster ist, dass sie mit verschiedenen Linienstilen dargestellt werden. In Abbildung 1.7 sind unterschiedliche abgebildet.

Die Umsetzung in der Client/Server-Architektur kann in Abschnitt 1.5.7 auf Seite 43 nachvollzogen werden.

¹²entnommen aus Falk, Becker und Marohn [27]

¹³Definition siehe Anhang 37, S. 177



1.2.4 Ausgabe von statistischen Plots

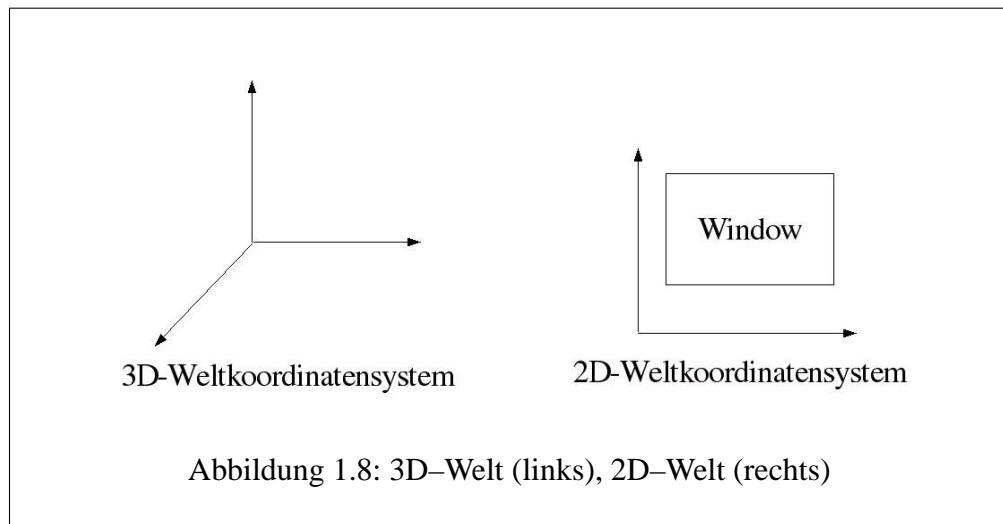
In den vorherigen Abschnitten wurden unter statistischen Plots Grafiken verstanden, die bestimmte Eigenschaften¹⁴ anschaulich darstellen. Im Folgenden werden die statistischen Plots aus einem anderen Blickwinkel betrachtet: Es wird nicht auf die mathematischen und statistischen Grundlagen eingegangen, sondern auf die Elemente, die zur Visualisierung eines statistischen Plots benötigt werden. Thematisiert wird die Problematik der Ausgabe aus Sicht der grafischen Datenverarbeitung behandelt.

1.2.4.1 Definition eines Grafikobjektes

Wie in Abschnitt 1.2.2 dargestellt, gibt es viele verschiedene statistische Plots, die alle ein anderes Aussehen haben. Werden sie aus graphischer Sicht betrachtet, so stellt man dennoch eine Gemeinsamkeit fest: Sie sind alle aus Grafikprimitiven zusammengesetzt. Unter Grafikprimitiven werden Punkt, Linie, Polygon, Kreis, Kreisabschnitt und Textelement verstanden. Wird ein Plot nicht aus der Sicht der Statistik, sondern aus Sicht der grafischen Datenverarbeitung (einem Teilgebiet der Informatik) betrachtet, so wird von einem *Grafikobjekt* gesprochen. Ein Grafikobjekt enthält alle Informationen, wie sich ein Plot aus den Grafikprimitiven zusammensetzt und kann diese für die Ausgabe berechnen und bereitstellen. Im Folgenden werden nun weitere wichtige Bestandteile erklärt, die für statistische Visualisierungen notwendig sind.

¹⁴statistische Funktionale

1.2.4.2 Weltraum und Weltkoordinatensystem

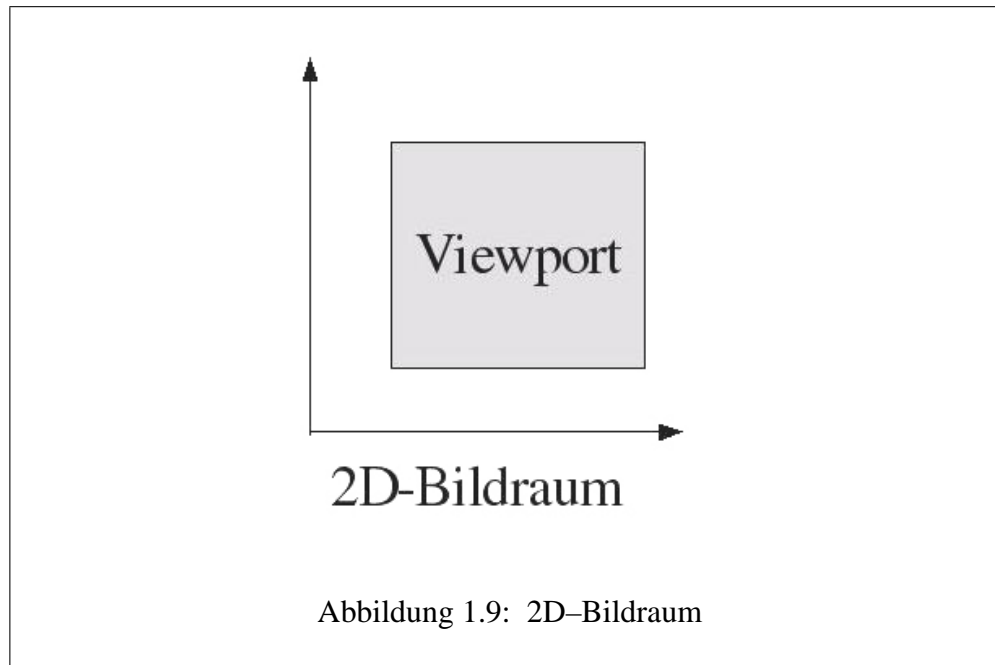


In einem Weltraum¹⁵ leben sämtliche Grafikobjekte. Er gibt die reale Anordnung, Größe und Form der Grafikobjekte wieder. Durch das Weltkoordinatensystem ist es möglich Objekte in dem Weltraum anzulegen, dabei können Größe, Form und Ort des Objektes festgelegt werden. Außerdem werden durch das Weltkoordinatensystem zwei Objekte vergleichbar. Diese Eigenschaft ist besonders wichtig für statistische Visualisierungen, weil dadurch die statistischen Eigenschaften mehrere Plots verglichen werden können. Durch statistische Plots können Aussagen und Folgerungen über die zu Grunde liegenden Daten oder Funktionale und deren statistischen Eigenschaften gemacht werden.

Das Weltkoordinatensystem ist, je nach Art des statistischen Plots, zwei- oder dreidimensional. Ein Ausschnitt des Weltkoordinatensystems wird als Window bezeichnet. Es ist festgelegt, dass nur die Grafikobjekte, die in diesem Window liegen, ausgegeben werden sollen. Das Window muss ebenfalls im Weltkoordinatensystem spezifiziert werden. Eine Aussage über die ideale Größe des Windows kann nicht gemacht werden, da sie von dem zu visualisierenden Objekt, in diesem Fall dem statistischen Plot, abhängt. Daher sollte sie der Ersteller des Plots spezifizieren können.

¹⁵Hier ist nicht das All oder das Universum gemeint.

1.2.4.3 Bildraum und Bildkoordinatensystem



Der Bildraum besitzt ebenfalls ein Koordinatensystem. Es beschreibt die Größe der möglichen Ausgabe auf dem Ausgabemedium¹⁶. Kann nicht der Bildraum, sondern nur ein Ausschnitt zur Ausgabe verwendet werden, so muss dieser im Bildkoordinatensystem festgelegt werden. Dieser Ausschnitt wird dann als *Viewport* bezeichnet.

Es sei noch angemerkt, dass wenn von Koordinatensystem in den beiden Räumen gesprochen wird, dies noch nicht in Verbindung mit einem Koordinatensystem, welches in einem Plot dargestellt wird, zu bringen ist.

1.2.4.4 Window-Viewport-Transformation

Wir wissen jetzt was ein Grafikobjekt ist, wo es lebt und wo es ausgegeben werden soll. In diesem Abschnitt soll erklärt werden, wie es zu einer Ausgabe kommt, und welche Berechnung dazu durchgeführt werden muss. Es geht darum, wie ein Grafikobjekt aus dem Weltraum¹⁷ in dem Bildraum ausgegeben wird. Die Berech-

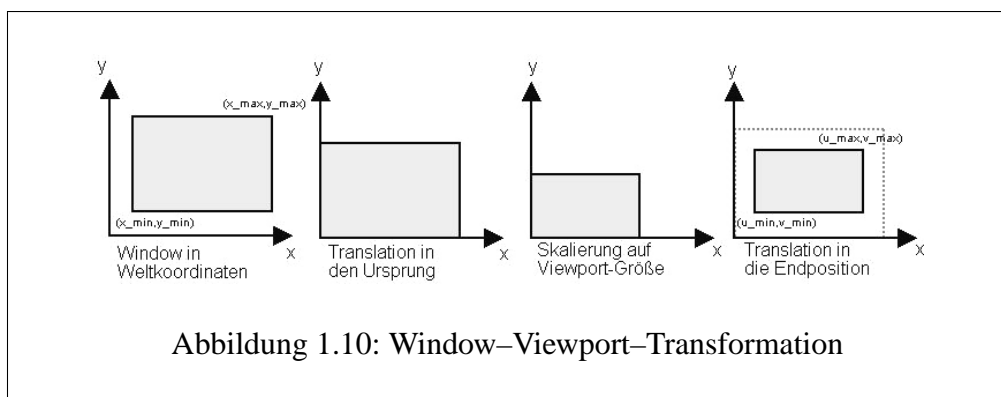
¹⁶z. B. Monitor

¹⁷Der Weltraum ist der Raum in dem die Grafikobjekte in realer Größe leben.

nung wird *Window–Viewport–Transformation* genannt und setzt sich aus folgenden Schritten zusammen:

1. **Clipping**
2. **Translation¹⁸ in den Koordinatenursprung**
3. **Skalierung**
4. **Translation in den Viewport**

Zunächst werden durch das Clipping alle Grafikobjekte, oder auch Teile der Grafikobjekte, die außerhalb des Windows¹⁹ liegen, entfernt, danach wird das Window in den Koordinatenursprung des Weltkoordinatensystems verschoben. Nachdem das Window auf die Größe des Viewports skaliert worden ist, wird der Ausschnitt in den Viewport verschoben. Bei allen Transformationen, die bei der Window–Viewport–Transformation stattfinden, werden nur solche benutzt, die das Größenverhältnis der Objekte aus dem Weltraum nicht verändern, und die Seitenverhältnisse jedes einzelnen Objektes erhalten. Es soll zu keiner Verzerrung eines Objektes kommen. Anschaulich ist die Window–Viewport–Transformation in Abbildung 1.10 dargestellt. Wie sie mathematisch umgesetzt wurde, ist dem Anhang A.1.1 auf Seite 157 zu entnehmen.



¹⁸Verschiebung

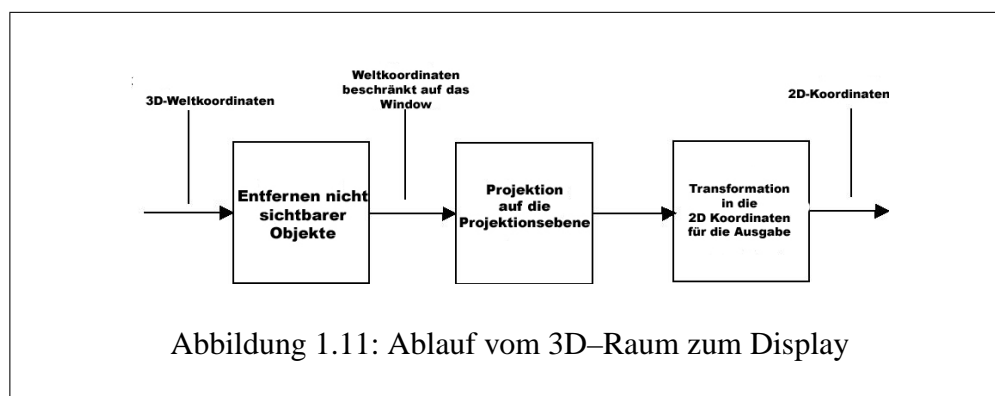
¹⁹Falls ein Window im Weltraum definiert wurde, ansonsten entfällt das Clipping, da der komplette Weltraum ausgegeben werden soll.

1.2.4.5 Projektion

Die Window–Viewport–Transformation bildet nur von einem zweidimensionalen Weltraum in einen zweidimensionalen Bildraum ab. Damit auch Objekte aus einem 3D–Weltraum in einem 2D–Bildraum ausgegeben werden können, muss das 3D–Window in ein 2D–Window transformiert werden.

Dies geschieht durch eine Projektion, die 3D–Objekte in einen 2D–Übergangsraum transformiert. Für diese Projektion gibt es mehrere Möglichkeiten. In dieser Arbeit wurde eine Projektion verwendet, die Grafikobjekte und damit die statistischen Plots realitätsnah wiedergibt. Nur wenn die Projektion diese Eigenschaft besitzt, kann der Betrachter den statistischen Plot und dessen Eigenschaften richtig interpretieren. Der Aufbau und die mathematischen Berechnungen können in Abschnitt A.1.3 auf Seite 160 nachvollzogen werden. Folgende Schritte werden für die Ausgabe eines 3D–Objektes benötigt und in Abbildung 1.11²⁰ anschaulich dargestellt:

1. Clipping im 3D–Window
2. Projektion
3. Window–Viewport–Transformation (ohne Clipping)



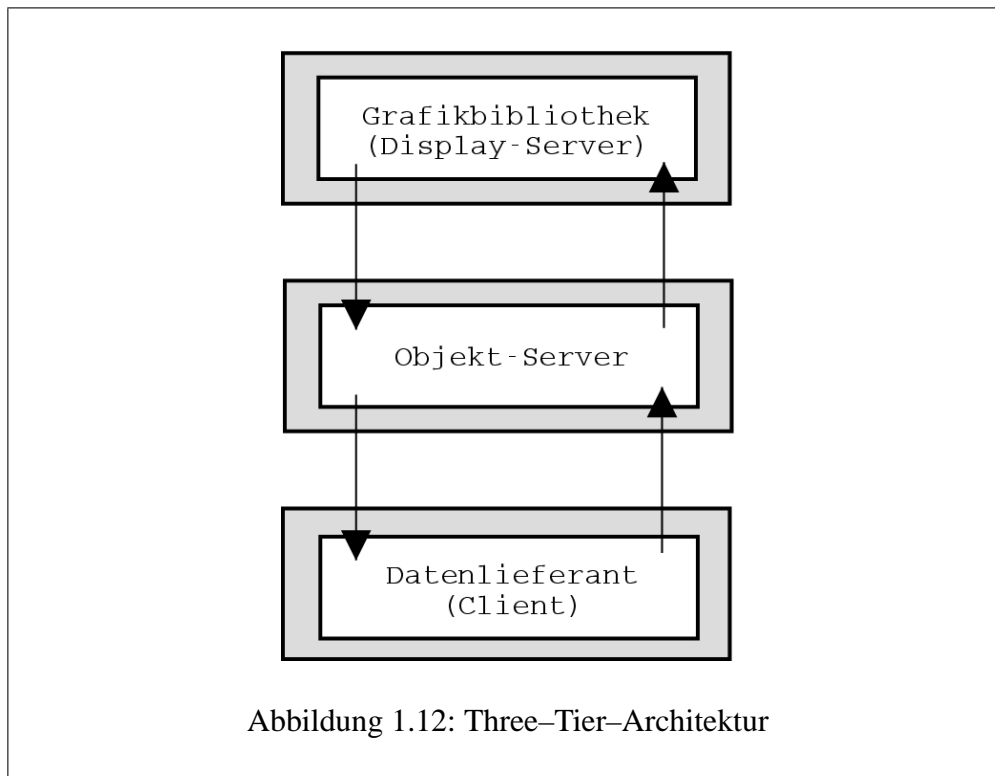
²⁰Entnommen aus Freisleben und Baumgart [29]

1.2.4.6 Umsetzung in der Architektur

In den vorhergehenden Abschnitten wurde die formale Vorgehensweise der Ausgabe eines Grafikobjektes aus dem Weltraum in den Bildraum beschrieben. In der Architektur werden die Grafikobjekte im Objekt-Server erzeugt. Er berechnet die Daten für die Ausgabe im Display-Server. Diese sind zweidimensionale reelle Daten, d. h. der Objekt-Server führt das Clipping und bei dreidimensionalen Daten die Projektion durch. Erst im Display-Server finden die weiteren Transformationen der Window-Viewport-Transformation statt. Diese sind nicht so rechenintensiv, was den Display-Server entlastet. Außerdem wird durch die Übergabe von zweidimensionalen Daten der Datentransfer beschleunigt. Bei dreidimensionalen Daten fällt durch die Projektion eine Koordinate der Daten weg und durch das Clipping werden nur die sichtbaren Daten übertragen.

1.3 High-Level-Design der Architektur

Ziel dieser Arbeit ist es eine Anwendung zu entwickeln, die statistische Visualisierungen berechnet und ausgibt. Das Einsatzgebiet der Architektur soll möglichst breit sein, wie z. B. Einsatz lokal als auch im Web. Diese Anwendung muss also verschiedenste Aufgaben erfüllen. Dies kann verdeutlicht werden, wenn der Ablauf wie es zu einer Ausgabe eines statistischen Plot kommt, betrachtet wird. Als erstes müssen die statistischen Daten erzeugt werden. Dies geschieht indem Berechnungen mit einem Datensatz durchgeführt werden, oder die Daten über ein Funktional²¹ berechnet werden. Anschließend wird der Plot aus diesen Daten berechnet, damit dann die Ausgabe (z. B. auf dem Bildschirm) erfolgen kann.



Die Client/Server-Architektur wurde als Three-Tier-Lösung (Abb. 1.12) umgesetzt, wobei die zu erfüllenden Aufgaben auf drei Komponenten verteilt wurden. Es war erforderlich die Aufgabenbereiche klar voneinander abzugrenzen. Jede dieser Komponenten sollte ihre Aufgaben möglichst schnell ausführen. Au-

²¹In diesem Fall liegt kein Datensatz den Berechnungen zu Grunde.

Berdem sollten sie auf verschiedenen Rechnern mit unterschiedlicher Hard- und Software laufen können. Um eine höchst mögliche Flexibilität zu erreichen, soll eine Anpassung der Client/Server-Architektur an ein anderes System²² möglichst durch Abänderung nur einer Komponente geschehen²³. Damit eine Komponente leicht ausgetauscht werden kann, muss die Kommunikation zwischen den Komponenten klar definiert sein. Diese wird in einer Schnittstelle definiert und stellt einen Vertrag dar, an den sich die Komponente halten muss, um mit den anderen in Verbindung treten zu können. Die Schnittstellen müssen bei der Neuerstellung von Komponenten immer berücksichtigt werden. Einem Betrachter von statistischen Visualisierungen, die durch die Architektur erstellt worden sind, fällt dabei nicht auf, ob die Komponenten lokal, oder verteilt auf verschiedenen Rechnern laufen. Die Aufteilung in Komponenten ist nur für den Entwickler sichtbar.

Damit die Client/Server-Architektur stets an neue Bedürfnisse und Ansprüche angepasst werden kann, ist es sinnvoll, vorhandene Komponenten um Neuerungen zu erweitern, oder, wenn eine Neuerstellung nötig ist, auf schon vorhandene Teile zurückzugreifen. Deshalb wurden in der Architektur objektorientierte Strukturen verwendet, die es ermöglichen, durch Vererbung vorhandene Klassen einzubinden. Außerdem bekommt die Architektur durch das objektorientierte Design eine bessere Übersicht. Dies wird besonders in der Struktur der CORBA-Schnittstelle berücksichtigt.

In den nächsten Abschnitten werden die einzelnen Komponenten der Client/Server-Architektur vorgestellt. Die Namen der Komponenten wurden nach ihren Aufgaben ausgewählt. Jede Komponente kann dabei intern, relativ zu einem bestimmten Request²⁴, einen Client oder einen Server darstellen.

1.3.1 Display-Server

Der Display-Server ist für die Ausgabe des statistischen Plots zuständig. Wie schon erwähnt, setzt sich der statistische Plot²⁵ aus Grafikprimitiven zusammen, die er vom Objekt-Server geliefert bekommt. Die Ausgabedaten werden im Display-Server abgespeichert, damit dieser auf sie auch unabhängig vom Objekt-

²²Hardware oder Betriebssystem

²³In sehr komplizierten Fällen werden auch zwei Komponenten der Architektur abgeändert werden müssen.

²⁴Def.: Der Aufruf und die eventuelle Rückgabe eines Ergebnisses heißt *Request*.

²⁵das Grafikobjekt

Server zugreifen kann, wenn er diese für eine Ausgabe benötigt, unter der Voraussetzung, dass sich nichts an den Daten verändert hat. Dies entlastet die Kommunikation zwischen Display- und Objekt-Server und kann die Ausgabe beschleunigen. Der Display-Server enthält eine Grafikkbibliothek, mit der die Primitiven visualisiert werden. Diese enthält die Window-Viewport-Transformation (siehe Abschnitt 1.2.4.4, S. 157), die vor der Ausgabe einer Primitiven durchgeführt werden muss. Weiterhin kann der Display-Server, falls es für den statistischen Plot notwendig ist, ein Koordinatensystem²⁶ berechnen und ausgeben.

Ein Display-Server kann außerdem mehrere Ausgabefenster verwalten, d. h. es können mehrere statistische Plots unabhängig voneinander in einem Display-Server ausgegeben werden. Des Weiteren kann er für jedes dieser Ausgabefenster eine Schnittstelle zur Interaktion für den Betrachter bereitstellen. Im Fall von Interaktion leitet der Display-Server die entsprechenden Events, die durch den Betrachter erzeugt worden sind, an den Objekt-Server weiter.

1.3.2 Objekt-Server

Die Grafikobjekte bilden den Ausgangspunkt für die statistischen Visualisierungen. Sie werden durch den Objekt-Server erzeugt. Der Objekt-Server enthält eine Factory (siehe dazu Anhang A.2, S. 163 oder Gamma et. al. [31]), die diese Aufgabe übernimmt. Über sie werden die neuen Grafikobjekte erzeugt. Nähere Erklärungen über den Aufbau und deren Funktionsweise werden im Anhang A.2.2 auf Seite 164 gegeben. Neben den Grafikobjekten verwaltet der Objekt-Server auch den entsprechenden Weltraum der Grafikobjekte. Für jeden Display-Server, bzw. für jedes neue Ausgabefenster im Display-Server, wird ein neuer Weltraum erzeugt und verwaltet.

Der Objekt-Server bildet den Ausgangspunkt für die Kommunikation der einzelnen Komponenten der Client/Server-Architektur. Er verwaltet die Verbindungen zum Display-Server und stellt dem Client die Grafikobjekte zur Verfügung. Außerdem können über den Objekt-Server Eigenschaften des Display-Servers gesetzt werden. Der Objekt-Server übernimmt die Kontrolle der Kommunikation und macht die Architektur dadurch sicherer, d. h. es kommt zu keinen falschen Zugriffen. Ebenso werden alle intensiven Berechnungen, die die Grafikobjekte und

²⁶Das Koordinatensystem wird anhand des Bildraumes und des statistischen Plots berechnet. Es ist nicht gleichzusetzen mit dem Welt- oder Bildkoordinatensystem.

damit die statistischen Plots betreffen, vom Objekt-Server durchgeführt. Diese Berechnungen können die Projektion von 3D-Grafikobjekten, die Rotation dieser, oder das Entfernen von verdeckten Linien sein. Außerdem berechnet er Daten, die an den Display-Server zur Ausgabe weitergegeben werden. Diese Daten sind die Grafikprimitiven, aus denen sich das Grafikobjekt, der Plot, zusammensetzt. Da diese Berechnungen hier ausgeführt werden, wird der Display-Server entlastet. Die Berechnungen bilden einen Teil der Grafikkbibliothek in der Architektur. Es kann aber nicht ohne den Display-Server zu einer Ausgabe kommen. Die Funktionen, die in diesem Teil der Grafikkbibliothek enthalten sind, haben nichts mit der direkten Ausgabe zu tun²⁷, sondern bereiten diese nur vor. Ein Teil dieser Berechnungen muss nur ausgeführt werden, wenn diese vom Display-Server angefordert werden. Der Objekt-Server bietet dem Display-Server eine Möglichkeit zur Kommunikation über Events. Die Events werden beim Auftreten bearbeitet und die entsprechenden Aktionen werden ausgeführt, zum Beispiel die Rotation der Grafikobjekte. Tritt dieses Event auf, so werden die Daten für die Ausgabe im Objekt-Server berechnet. Danach werden die berechneten Daten zur Ausgabe an den Display-Server übergeben.

1.3.3 Client

Die statistischen Daten für die Berechnungen liefert der Client. Er ist die „Datenbank“, die den Ausgangsdatensatz liefert. Alle mathematischen und statistischen Berechnungen, die nicht mit dem statistischen Plot zusammenhängen, werden im Client durchgeführt. Auch diese Berechnungen können sehr intensiv sein, was bei der Entwicklung des Clients zu berücksichtigen ist. Der Client wird die Komponente sein, die am meisten an die eigenen Bedürfnisse angepasst wird, da er die Daten für den Plot liefert, diesen im Objekt-Server erzeugt, und die Ausgabe einleitet. Er fordert das entsprechende Grafikobjekt vom Objekt-Server an und übergibt an dieses die neu berechneten Daten. Für den Client stellen die Grafikobjekte die statistische Grafikkbibliothek dar. Es sieht nicht die dahinter liegende Struktur der Grafikkbibliothek im Objekt-Server und Display-Server. Idealerweise sollte er auch die einzige Komponente sein, die verändert werden muss.

Um die Interaktion mit einem Betrachter des Display-Server zu gewährleisten, kann der Client auf Callbacks reagieren, z. B. bei dem Verschieben eines Sliders²⁸

²⁷Die Funktionen zur direkten Ausgabe sind im Display-Server enthalten.

²⁸dt. Schieberegler

durch Neuberechnung der Daten. Callbacks sind Events, die vom Display-Server über den Objekt-Server an den Client weitergeleitet werden. In diesem Fall tritt der Client als Server auf und arbeitet das Callback ab. Dies führt dann zu einer Neuberechnung der Daten für den Plot mit den übergebenen Parametern.

1.3.4 Datensicherheit

Datensicherheit ist heute ein sehr wichtiges Thema. Man möchte einem Unbekannten nicht mehr Informationen zur Verfügung stellen, wie unbedingt notwendig ist. Dies wird auch bei der Client/Server-Architektur berücksichtigt, indem der Objekt-Server nur die Grafikprimitiven eines Plots an den Display-Server liefert. Bei der Kommunikation zwischen Objekt- und Display-Server wird zumeist kein Datensatz übertragen, sondern nur die Daten der Grafikprimitiven. Im Fall eines zweidimensionalen Scatterplot kann es zur Übertragung eines Datensatzes kommen²⁹. In diesem Fall muss sich der Ersteller des Plots dessen bewusst sein und abwägen, ob die Datensicherheit oder die Visualisierung wichtiger ist. Datensätze werden aber vom Client an den Objekt-Server übergeben, deshalb ist es notwendig, deren Kommunikation besonders zu schützen. Den Bedürfnissen der Datensicherheit muss bei der Umsetzung in einem Prototypen unbedingt entsprochen werden, besonders wenn die Komponenten auf verteilten Rechnern laufen. Die Komponenten sollten über eine sichere Verbindung, wie zum Beispiel über SSL³⁰, miteinander kommunizieren. Sonst werden nur Daten übertragen, die zur Visualisierung benötigt werden. Dies sind im Allgemeinen nur Funktionale der eigentlichen Daten. Aus diesen können nur selten die Ausgangsdaten berechnet werden.

²⁹Dies ist nur ganz selten der Fall, da die Daten meist vor der Ausgabe durch die Window-Viewport-Transformation neu skaliert werden.

³⁰Secure Sockets Layer

1.4 Auswahl der Middleware für die Umsetzung

Im letzten Abschnitt ging es um das High-Level-Design der Architektur. Nun geht es einen Schritt weiter, die zu stellenden Anforderungen an die Architektur werden diskutiert und festgelegt. Damit die Komponenten miteinander kommunizieren können, wird eine sogenannte Middleware benötigt. Middleware wird wie folgt definiert³¹:

Middleware bezeichnet eine Softwareschicht zwischen dem Übertragungsnetzwerk und den Anwendungen. Ihre Aufgabe ist es, von den Eigenheiten und der Komplexität der verwendeten Infrastruktur zu abstrahieren und den Anwendungen eine reibungslose, standardisierte Interaktion zu ermöglichen. Unter dem Begriff des Dienstes selbst ist eine Programmeinheit zu verstehen, die eine abgeschlossene Aufgabe erfüllt und von einem Dienstanbieter entwickelt, gewartet und angeboten wird. Das Spektrum möglicher Dienste reicht von systemnahen Diensten über komplexere, benutzernahe Anwenderdienste bis hin zu Standardwerkzeugen der Bürokommunikation. Dienste gliedern sich oft nahtlos in eine Middleware ein, die dafür sorgt, dass die Dienstanbieter einen Dienst finden und mit ihm interagieren können.

Eine Middleware stellt vor allem Dienste für Identifikation, Authentifizierung, Zugriff und Informationsaustausch oder Sicherheit zur Verfügung, um so vernetztes Arbeiten zu erleichtern. Heutzutage beinhalten viele Applikationen allerdings noch selber solche Dienste, was zu konkurrierenden und inkompatiblen Standards führt.

Die Middleware, die für die Umsetzung benutzt werden soll, muss folgende Anforderungen erfüllen:

Sie sollte, wegen der klaren Struktur und der Vererbungsmöglichkeit objektorientiert sein, und eine sichere Kommunikation unterstützen. Außerdem sollte sie die Kommunikation zwischen verschiedenen Programmiersprachen gewährleisten, damit die Komponenten in verschiedenen Programmiersprachen umgesetzt werden können, was Geschwindigkeit und Flexibilität der Architektur erhöht. Außerdem sollte die Kommunikation unabhängig vom Standort der Komponenten stattfinden können³². In den nächsten Abschnitten werden verschiedene Alternativen für die Entwicklung der Architektur vorgestellt. Am Ende des Kapitels folgt ein

³¹entnommen aus: Fraunhofer IIS – Embedded Communication

³²Verteilungstransparenz

Resümee, welches die Auswahl der Middleware und der Programmiersprachen begründet.

1.4.1 Common Objekt Request Broker Architecture (CORBA)

CORBA³³ ist eine Entwicklung der Object Management Group (OMG). Die OMG ist eine nicht-kommerzielle Arbeitsgemeinschaft, die 1989 gegründet wurde. CORBA ist ein Standard, mit dem Programme oder Programmteile verschiedener Programmiersprachen miteinander verbunden werden können. Dieser Standard hat einen objektorientierten Aufbau. Er ist verteilungstransparent, d. h. die Komponenten³⁴ können auf verschiedenen Rechnern, die durch Netzwerk miteinander verbunden sind, laufen. Die Kommunikation der einzelnen Komponenten wird in Schnittstellen festgelegt. Diese werden mit der sogenannten Interface Definition Language (IDL)³⁵ erstellt, und mittels Language Mapping³⁶ in die entsprechende Programmiersprache übersetzt³⁷. Die Syntax der IDL ähnelt der von C++ und JAVA. Der CORBA-Standard hält sich an die Trennung zwischen Interface und Implementierung. Dabei muss nur die Implementierung in der gewünschten Programmiersprache erfolgen. Die Programmiersprache und somit auch die Middleware-Software wird nach den Anforderungen an die entsprechende Komponente ausgewählt. Eine andere Möglichkeit zur Erstellung von Schnittstellen kann über Dynamic Skeleton Interface (DSI) und Static Skeleton Interface (SSI) vorgenommen werden. Bei dieser Möglichkeit muss sich der Programmierer selbst darum kümmern, wie und wann die Schnittstellenfunktionen mit einer Implementation ausgefüllt werden.

Der wichtigste Bestandteil CORBAs ist der Object Request Broker (ORB) (siehe Abb. 1.13³⁸). Er lokalisiert den Aufenthaltsort und den zuständigen Objektadapter für ein Objekt. Letztgenannter ist für die Aktivierung und Deaktivierung der Objekte zuständig. Ein Objekt besteht zum einen aus einem CORBA-Objekt,

³³Zusätzliche Informationen findet man unter www.omg.org.

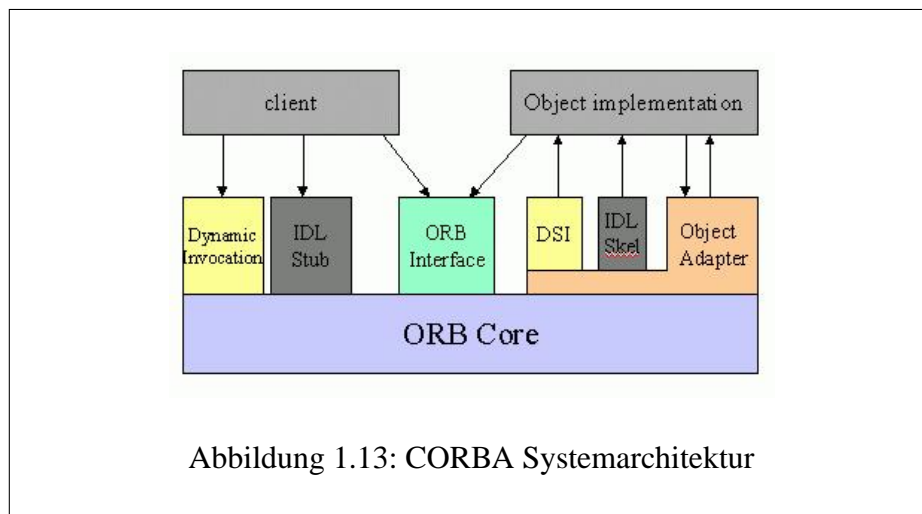
³⁴Client und Server

³⁵Wenn in diesem Abschnitt von der IDL gesprochen wird, ist damit die OMG IDMG IDL gemeint.

³⁶Language Mapping beschreibt die Umsetzung zwischen Programmiersprache und Schnittstellensprache.

³⁷Dies geschieht mit dem IDL-Compiler

³⁸Abb. entnommen aus dast.nlanr.net/Guides



zum anderen aus dem Servant oder einem Proxy³⁹. Der Servant ist die Implementation des CORBA-Objektes und realisiert dessen Semantik. Das CORBA-Objekt im Client kann über einen Proxy auf den Servant im Server zugreifen. Der Proxy wird nach dem Erhalt einer Objektreferenz durch den ORB⁴⁰ erzeugt. Proxies kommen sowohl auf der Client⁴¹, wie auch auf der Server-Seite⁴² zum Einsatz. Die Stubs und Skeletons werden bei statischen Aufrufen verwendet. Bei ihrem Einsatz ist die Objekt-Schnittstelle zur Übersetzungszeit vorhanden⁴³. Ist dies nicht der Fall, so bietet CORBA die Möglichkeit über das Dynamic Invocation Interface (DII; Client-seitig) und über das Dynamic Skeleton Interface (DSI; Server-seitig) die Schnittstelle zur Laufzeit dynamisch zu erzeugen und für die Verarbeitung bekannt zu geben. Damit kann die Kommunikation zwischen dem CORBA-Objekt und dessen Proxy stattfinden. Hinter dem Proxy kann sich ein Servant befinden, der mit diesem verknüpft ist und die Ausführung der Funktionen übernimmt. Für die Datenübertragung wird die Common Data Representation (CDR) verwendet. Das verwendete Kommunikationsprotokoll⁴⁴ ist das Internet Inter-ORB Protocol (IIOP).

³⁹Hier muss der Programmierer entscheiden, wie die Funktionen spezifiziert werden sollen.

⁴⁰Client-ORB

⁴¹Sie heißen Stubs auf der Client-Seite.

⁴²Auf der Server-Seite ist dies der POA, der die Zugriffe weiterleitet.

⁴³Die Schnittstelle ist bei der Programmerstellung (Kompilation und Linken) vorhanden.

⁴⁴Das Kommunikationsprotokoll beschreibt den Datenversand zwischen verschiedenen Rechnern. In dem Protokoll wird festgelegt, wie die Rechner miteinander kommunizieren.

CORBA unterstützt die verschiedensten Betriebssysteme (Linux, Unix, Windows, etc.) mit den unterschiedlichsten Programmiersprachen. Language Mappings sind zum Beispiel für C++, JAVA, COBOL oder Smalltalk vorhanden (Hofmann et. al. [44]).

CORBA bietet zusätzlich zum Basisumfang noch eine Sammlung von Dienstspezifikationen an. Sie werden CORBA–Services genannt, ihre Schnittstellen werden mit der IDL beschrieben. In Tabelle 1.1 werden sie kurz vorgestellt (entnommen aus Hofmann et. al.[44]). Zum Einsatz kommen sie hauptsächlich bei unternehmenskritischen Anwendungen. Die Schnittstellen werden ebenso wie die selbst definierten verwendet. Die Dokumentation der Services ist unter www.omg.org zu finden.

1.4.2 Remote Method Invocation (RMI)

Verteilte Anwendungen wie die Client/Server–Architektur, können unter JAVA mit der Remote Method Invocation (RMI) umgesetzt werden. Das RMI erlaubt es Objekten von einer JAVA Virtual Machine (VM) auf die Methoden anderer Objekte zuzugreifen, die in einer anderen virtuellen Maschine (VM) laufen. Das Konzept ist angelehnt an das von CORBA. JAVA–Objekte können über Proxies⁴⁵ auf Methoden anderer Objekte zugreifen, dabei wird die Schnittstelle durch das RMI bereitgestellt. Es gibt keine besondere dafür definierte Sprache, um eine solche Schnittstelle zu definieren. Alle Schnittstellen werden von einer JAVA–Klasse⁴⁶ abgeleitet. RMI umfasst damit eine Menge von JAVA–Klassen, Schnittstellen und Tools. Der Aufbau von RMI ist Abbildung 1.14⁴⁷ zu entnehmen. Die Kommunikation erfolgt über das HTTP–Protokoll. Die RMI–Registry verwaltet die JAVA–Objekte und ermöglicht den Zugriff auf diese. RMI ist ebenso wie CORBA objektorientiert, unterstützt aber hauptsächlich die Kommunikation von JAVA–Komponenten untereinander. Es wird keine andere Programmiersprache berücksichtigt, RMI bietet aber eine Schnittstelle für CORBA.

Das dynamische Laden von Objekten ermöglicht es, die Objekte erst zur Laufzeit bekannt zu geben, d. h. es kann eine Klasseninformation zu einem Remote–

⁴⁵Stub und Skeleton

⁴⁶java.rmi.*

⁴⁷Entnommen aus Irmscher [47].

Service	Beschreibung
Collection Service	Verwaltung von Objekten mit gemeinsamen Kriterien z. B. in Maps, Sets oder Queues.
Concurrency Control Service	Sperren von Threads oder Transaktionen über einen Lockmanager.
Event Service	Dynamisches Erzeugen von Events durch CORBA-Objekte.
Externalization Service	Objekte können in einen String verwandelt werden. Ebenso ist es möglich, aus einem String ein Objekt zu erzeugen.
Licensing Service	Operationen, die es ermöglichen, verwendete Ressourcen nach bestimmten Kriterien zu berechnen.
Life Cycle Service	Definiert Operationen, über welche Objekte erzeugt, kopiert, verschoben und gelöscht werden können, Hofmann et. al. ([44], S. 335).
Naming Service	Zugriff auf CORBA-Objekte anhand ihres Namens.
Notification Service	Erweitert den Event Service, indem jede beliebige Datenstruktur als Ereignis übertragen werden kann und nicht nur die im Event Service typisierten Ereignisse (Hofmann et. al. [44], S. 335).
Persistent Object Service	Objekte können mit ihren Eigenschaften abgespeichert werden und nach Programmstart wieder geladen werden.
Property Service	Über den Namen oder Werte können Objekte eindeutig identifiziert und verwendet werden.
Query Service	Stellt Abfrageoperationen an Objekte zur Verfügung, und ist eine Übermenge von SQL (Hofmann et. al. [44], S. 335).
Relationship Service	Ermöglicht die dynamische Erzeugung von Assoziationen zwischen Objekten und das Traversieren dieser Links (Hofmann [44], S. 335).

Tabelle 1.1: CORBA-Services I

Service	Beschreibung
Security Service	Sorgt für eine sichere Kommunikation bei verteilten Objekten.
Time Service	Zeitsynchronisation zwischen verschiedenen Objekten.
Trading Service	CORBA-Objekte können nach ihren Fähigkeiten identifiziert werden.
Object Transaction Service	Verwirklicht das Konzept der Transaktionen (Hofmann [44], S. 335).

Tabelle 1.2: CORBA-Services II

Objekt⁴⁸, auf das ein Client eine Referenz besitzt, dynamisch als Bytecode⁴⁹ geladen werden. Der Client braucht somit keinen direkten Zugriff auf den Client-Proxie (Stub), sondern kann diesen zur Laufzeit anhand seiner Bedürfnisse nachladen. Hier besteht eine Trennung zwischen Schnittstellen und Implementation. Ein weiterer Vorteil ist, dass der Server zur Laufzeit verändert werden kann, ohne dass der Client etwas davon merkt. Notwendig ist nur die Erhaltung der Schnittstelle.

1.4.3 Component Object Model + Distributed COM

Das Component Object Model (COM)⁵⁰ wurde 1993 von Microsoft© für Windows-Plattformen veröffentlicht. Es bildet die neue Grundlage für die Verwaltung von Systemkomponenten, da auf ihm die verteilten Anwendungen, die auf OLE oder ActiveX beruhen, basieren.

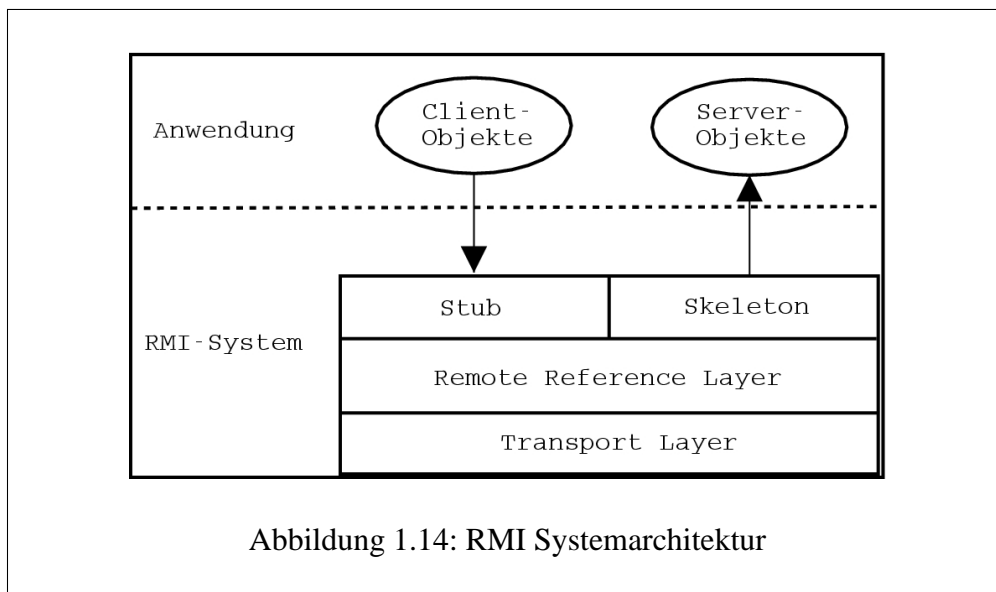
Jedes Objekt und seine Funktionalität wird bei COM in einer Komponente gekapselt. Der Zugriff auf diese erfolgt über eine öffentliche Schnittstelle, auf die Implementation gibt es keinen Zugriff. Die Schnittstellen werden mit der Microsoft Interface Definition Language (midl) definiert. Dabei bekommt jede Schnittstelle eine eindeutige Kennnummer, den GUID⁵¹ zugeordnet, über die der Zugriff auf die entsprechende Implementation erfolgen kann. Die Zuordnung der Ken-

⁴⁸Mit Remote Objekt wird der Servant bezeichnet, der auf Anfragen des Clients wartet.

⁴⁹Dies wird auch als Object Serialization bezeichnet.

⁵⁰Abb. entnommen aus msdn.microsoft.com/library

⁵¹Globally Unique Identifier



nummern mit dem jeweiligen Objekt erfolgt durch die Windows-Registry, welche in das Betriebssystem eingebunden ist. Ebenso stellt das Betriebssystem alle Mechanismen für die Kommunikation zwischen Client und Server zur Verfügung. COM realisiert dadurch den Nachrichtenverkehr. Die Verbindung wird ebenso wie bei CORBA über client- und serverseitige Proxies bewerkstelligt. Die Aktivierung und Lokalisation übernimmt bei COM der Service Control Manager (SCM). Ebenso wie in CORBA können die Schnittstellen statisch oder dynamisch erzeugt werden.

Verschiedene Programmiersprachen, wie z. B. C++, JAVA⁵² oder VBA, werden vom COM miteinander verbunden, d. h. Clients können somit in Excel, WSH, Python oder C++ erstellt werden. Der COM-Standard ist zunächst an einen lokalen Rechner gebunden, um verteilte Komponenten realisieren zu können, wurde COM erweitert. Die Erweiterung heißt Distributed-COM (DCOM) und unterstützt die Kommunikation von über dem Netz verteilten COM-Objekten. Es stellt Dienste eines Servers unter Berücksichtigung der Sicherheitsrichtlinien im Netz zur Verfügung.

Basierend auf COM (DCOM) wurden sehr rasch verschiedene Konzepte für die Erweiterung von Server-Komponenten entwickelt, und Technologien wie Automation und ActiveX haben in nahezu allen Entwicklungsumgebungen Einzug

⁵²Hiermit ist das Visual J++ gemeint

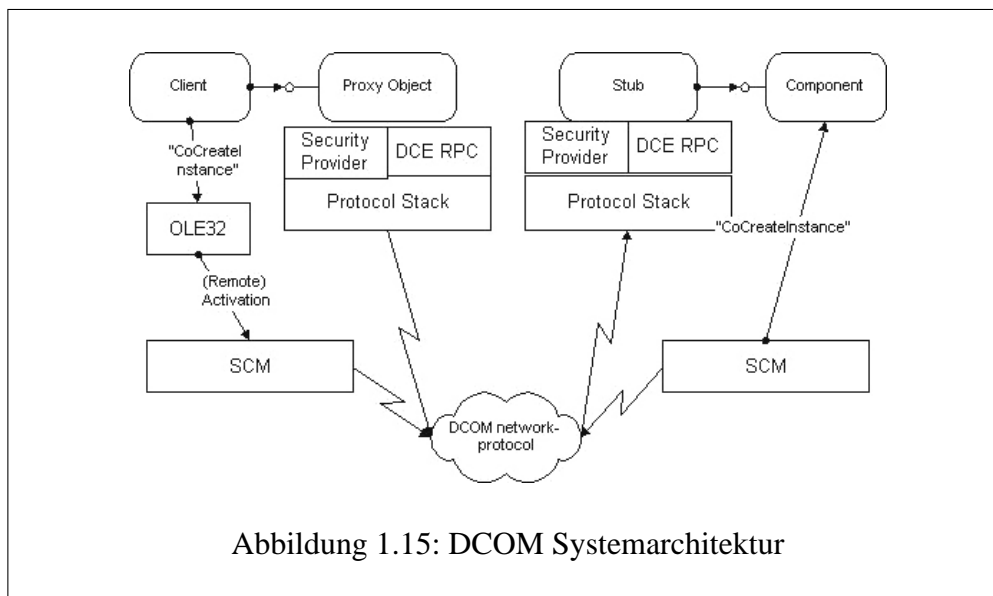


Abbildung 1.15: DCOM Systemarchitektur

gehalten (Hofmann et. al. [44]). COM enthält neben DCOM noch weitere Erweiterungen wie Klassenfabriken, Sicherheitsdienst, ActiveX, Transaktionen (MTS) und COM+.

1.4.3.1 DirectX

DirectX ist eine low-level API⁵³, die von Microsoft entwickelt wurde, um multimediale Ausgaben zu erzeugen. Sowohl die Ausgabe von 2D- und 3D-Grafiken und Audiodaten, als auch Interaktion mit dem Benutzer⁵⁴ werden unterstützt. Außerdem ist sie netzwerkfähig, d. h. Berechnung und Ausgabe können auf verschiedenen Rechnern stattfinden. DirectX hält sich an den oben angesprochenen COM-Standard, d. h. für multimediale Erweiterungen ist sie sehr geeignet. DirectX wurde hauptsächlich für die schnelle Ausgabe von 3D-Grafiken und Sounds entwickelt, was besonders bei der Entwicklung von PC-Spielen benötigt wird. DirectX in Verbindung mit COM kann zur Erstellung einer Client/Server-Architektur verwendet werden.

⁵³Definition API: Application Programming Interface, Anwendungsprogrammierschnittstelle. Eine von einem Betriebssystem oder einem Anwendungsprogramm vorgegebene Schnittstelle, über die anderen Anwendungen standardisierte Software-Werkzeuge zur Verfügung gestellt werden (entnommen von www.net-lexikon.de).

⁵⁴Eine Schnittstelle für Eingabegeräte ist in DirectX eingebunden.

	CORBA	DCOM/COM	JAVA/RMI
	Common Object Request Broker	Distribute Component Model	JAVA Remote Method Invocation
Ersteller:	Object Management Group (OMG)	Microsoft	JAVA Sun
Schnittstellensprache:	IDL	MIDL	JAVA
Protokoll:	Internet Inter-ORB Protocol (IIOP)	Object Remote Procedure Call (ORPC)	JAVA Remote Method Protocol (IRMP)
Basisschnittstelle:	CORBA.Object	IUnknown, IDispatch	java.rmi.Remote(Interface), java.rmi.RemoteObject
Wiederverwendbarkeit:	Schnittstellenebene, Implementierungsebene (TIE-Ansatz)	Containment und Aggregation Modell	Vererbung
Objektidentifikation:	Schnittstellename	Class-ID über die Registry	
Objektlokalisierung:	Object Request Broker (ORB) und Object Adapter (OA)	Service Control Manager (SCM)	RMI-Registry
Objektaktivierung:	Object Adapter (OA)	Service Control Manager	RMI-Registry
Objektzerstörung:	Explizit über IDL-Methoden in C++, implizit über Garbage Collection in JAVA	Implizit über verteilte Referenzanzählung	Reference-Counting Garbage Collection

Tabelle 1.3: Gegenüberstellung von CORBA, DCOM/COM und RMI I

CORBA	DCOM/COM	JAVA/RMI
Common Broker	Distribute Component Object Model	JAVA Remote Method Invocation
Methodenaufrufe:	synchron und asynchron über Event/Notification Service 2000 asynchrone Aufrufe, Callbacks durch Connection Points	
Parameterübergabe:	Schnittstellen als Objektreferenz, oder per Wert	Remote-Objects als Referenz, Non-Remote-Objects als Kopie
Ausnahmebehandlung:	Standard-Exception-Mechanismus, zusätzliche Definition in der Schnittstelle	java.rmi.RemoteException und HRESULT-Wert
Dynamik:	DI, DSI, Typinformationen über Interface Repository	dynamical class loading über COM-Automation
		IDispatch-Schnittstelle, Type-Library

Tabelle 1.4: Gegenüberstellung von CORBA, DCOM/COM und RMI II

Vorteile	Nachteile
bessere Laufzeit	komplexer Aufbau
speziell an die Bedürfnisse angepasst	zeitintensive Entwicklung
geringer Overhead	keine Anbindung an bestehende Projekte
	eingeschränkte Portabilität
	schlechte Erweiterbarkeit
	Fehleranfälligkeit

Tabelle 1.5: Vor- und Nachteile einer eigenen Middleware

1.4.4 Eigene Middleware

Eine Alternative zu den bisher vorgestellten Middleware-Möglichkeiten wäre eine Eigenentwicklung. Sie könnte genau an die Ansprüche angepasst werden, die für die Umsetzung der Client/Server-Architektur benötigt werden. Ein optimaler Funktionsumfang wäre damit gewährleistet. Es gäbe keinen Overhead, kein zusätzlicher Ballast, der nicht für die Ausgabe von statistischen Visualisierungen benötigt⁵⁵ wird. Die Kommunikation könnte durch ein eigenes Protokoll erfolgen, welches auf das TCP/IP-Protokoll aufsetzen könnte.

Eine eigene Middleware zu erstellen ist generell möglich. Der Aufwand ist aber nicht gerechtfertigt, da die Erstellung eines eigenen Prototypen für die Forschungstätigkeit den zeitlichen Rahmen sprengen würde. Außerdem entstehen Probleme bei der Einbindung in eine schon vorhandene Komponente. Diese Möglichkeit scheidet wegen der vielen negativen Aspekte aus (siehe Tabelle 1.5).

1.4.5 OpenGL

OpenGL ist ein Industriestandard, der eine Schnittstelle für die Verwendung der Grafik-Hardware zur Verfügung stellt. Es ist ein Hardware unabhängiger low-level Standard mit einem sehr begrenzten Umfang an Grafikprimitiven. Aus diesen können komplexere Modelle zusammengesetzt werden. Es werden Transformationen, Projektionen, Kameratransformationen, Ray-Tracing, Reflexionsmodelle, Licht, Texturen und vieles mehr unterstützt (siehe dazu Slusallek [82]). Die

⁵⁵z. B. Methoden für die Darstellung von Texturen sein, falls diese nicht bewusst benötigt werden.

Möglichkeiten von OpenGL sind besonders auf die realistische Darstellung von 3D–Welten ausgerichtet.

Interaktion über Callbacks und Events werden von OpenGL zusätzlich unterstützt. Desweiteren können, verteilte Anwendungen mit OpenGL erstellt werden, d. h. Berechnungen können auf einem Server ausgeführt werden, wobei die Ausgabe auf einem Client im Netz (siehe Segal und Akeley [81]) erfolgt.

OpenGL ist eine Alternative zu DirectX, da es nicht an die Betriebssysteme von Microsoft gebunden ist.

1.4.6 XploRe

XploRe⁵⁶ ist eine statistische Software mit interaktiver, grafischer Benutzerumgebung. Ihr liegt eine objektorientierte, statistische Programmiersprache zu Grunde, mit der die Eingaben erfolgen. In XploRe ist bereits eine große Anzahl an statistischen Modellen, Schätzern, Tests und Plots eingebunden.

Die Ausgabe kann über interaktive Grafiken erfolgen, mit denen komplexe Sachverhalte dargestellt werden können. XploRe ist ebenfalls als Client/Server–Architektur einsetzbar. Es besteht die Möglichkeit für die Ausgabe einen JAVA–Client zu benutzen, wobei die Berechnungen im XploRe–Server stattfinden. Der Client bietet ein lokales graphisches Benutzer–Interface, mit dem es zur Interaktion mit dem Betrachter kommt. Diese Architektur ist auch für den Einsatz im Web oder Intranet geeignet, dort besonders im Bereich E–Learning. Sie verfolgt den Ansatz der Erstellung einer eigenen Middleware. Es wurde eine eigenes Kommunikationsprotokoll definiert, welches direkt auf das TCP/IP–Protokoll aufsetzt (siehe Eigene Middleware 1.4.4).

1.4.7 Resümee

Für die Umsetzung des Prototypen der Client/Server–Architektur wurde CORBA verwendet. CORBA ist ein öffentlicher, nicht–kommerzieller und gut dokumentierter Standard. Die Middleware CORBA ist nicht wie COM/DCOM an die Betriebssysteme von Microsoft gebunden. Für COM/DCOM spricht allerdings, dass die DirectX–Bibliotheken als COM–Objekte mitgeliefert werden. Da aber in der

⁵⁶Weitere Details sind unter Härdle et. al. [40], Cizek et. al. [16] und Härdle et. al. [41] zu finden.

Client/Server–Architektur nur einfache statistische Plots visualisiert werden sollen, ist DirectX nicht unbedingt notwendig. Deren Grafikbibliothek zielt genauso wie OpenGL auf das Erstellen von realistischen 3D–Welten ab, die besonders bei der Programmierung von Spielen benötigt werden.

Auf die Verwendung bestehender Client/Server–Architekturen für statistische Visualisierungen, wie z. B. XploRe, wurde verzichtet, da diese zu stark an ein statistisches Softwarepaket gebunden sind. Die Erweiterung der Architektur wird erschwert, da eine Einarbeitung in das Softwarepaket und die zu Grunde liegende statistische Programmiersprache notwendig ist.

Auch auf die Erstellung einer eigenen Middleware für statistische Visualisierungen, auf die die Client/Server–Architektur aufsetzen soll, wurde verzichtet. Der Aufwand steht nicht im Verhältnis zu den Vorteilen einer eigenen Middleware (siehe Tabelle 1.5). Soll die Client/Server–Architektur in einem kommerziellen Rahmen zum Einsatz kommen, ist diese Alternative wieder neu zu überdenken. Da in dieser Arbeit nur ein Prototyp erstellt werden soll, und dieser möglichst überschaubar und dessen Aufbau leicht verständlich sein soll, wird auf eine bereits vorhandene und damit bekannte Middleware zurück gegriffen. Der Aufbau und die Funktionsweise der Middleware ist bekannt, daher gestaltet sich die Einarbeitung leichter, als eine komplette Neuerstellung des Prototypen. Die Neuerstellung beinhaltet neben der Implementation der drei Komponenten, auch die Spezifikation der Kommunikation⁵⁷.

Da eine bekannte Middleware mit Standard–Programmiersprachen zur Umsetzung des Prototypen verwendet werden soll, fällt die Entscheidung auf die Middleware CORBA. Sie ist außerdem gut dokumentiert und kostenlos. Sie unterstützt viele Programmiersprachen, für den Prototypen werden in diesen Fall C++ und JAVA benutzt. C++, da es eine leistungsstarke und schnelle Programmiersprache ist, welche die Objektorientiertheit unterstützt, und JAVA, da sie hauptsächlich bei Web–Applikationen verwendet wird. Dies ermöglicht, verbunden mit der Middleware CORBA, einen flexiblen Einsatz des Prototypen.

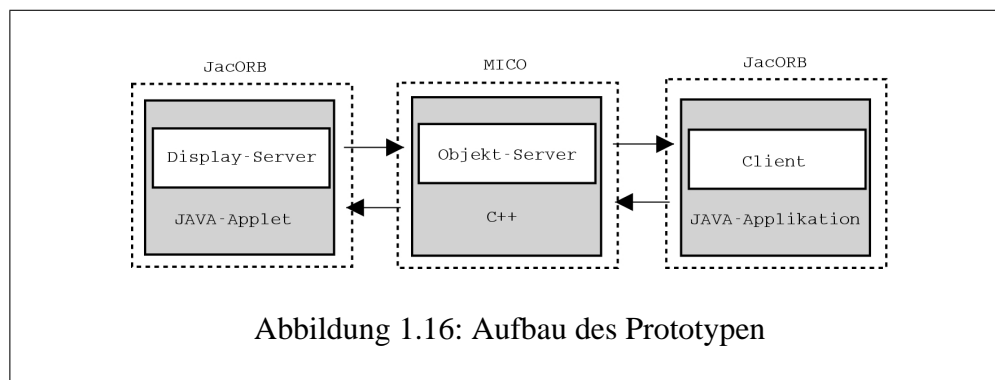
Anzumerken ist, dass mit allen anderen vorgestellten Alternativen die Erstellung eines Prototypen auch möglich gewesen wäre.

⁵⁷z. B. die Erstellung eines eigenen Protokolls.

1.5 Umsetzung spezieller Aufgaben in der CORBA-Implementierung

Die Umsetzung der Client/Server-Architektur für statistische Visualisierungen wird mit der Middleware CORBA durchgeführt. In dieser Arbeit soll ein Prototyp erstellt werden, der auf Web-Seiten eingesetzt werden kann, um dort statistische Plots zu visualisieren. Er soll Interaktivität unterstützen, damit er in multimedialen Konzepten und im E-Learning Bereich eingesetzt werden kann.

1.5.1 Details zum Prototypen



Der Prototyp besteht aus drei Komponenten (siehe Abb.1.16). Der Display-Server gibt den statistischen Plot aus, und soll in einer Web-Seite integriert werden. Für die Umsetzung bietet sich die Programmiersprache JAVA an, da ein Applet auf Web-Seiten leicht einzubinden ist. Die JAVA Virtual Machine (JVM) ist in den meisten Browsern schon installiert, daher ist kein zusätzlicher Download⁵⁸ notwendig. Als Middleware-Produkt wurde JacORB⁵⁹ von der Freien Universität Berlin verwendet. Da der Client ebenso in JAVA geschrieben wurde, findet auch dort das Middleware-Produkt JacORB Verwendung. Der Objekt-Server führt die intensivsten Berechnungen für die statistischen Plots⁶⁰ aus, deshalb wurde hier die

⁵⁸Zusätzliche Installationen und Anpassung an ein System ist somit nicht erforderlich.

⁵⁹www.jacorb.org

⁶⁰Im Client können auch intensive Berechnungen ausgeführt werden. Diese haben aber nicht direkt etwas mit dem statistischen Plot zu tun.

Programmiersprache C++⁶¹ benutzt. Für den Objekt-Server kommt das Produkt MICO⁶² als Middleware zum Einsatz. Den Ausgangspunkt des Prototypen bildet die CORBA-Schnittstelle (siehe Anhang A.3, S. 166 ff.).

Nachfolgend wird auf spezielle Implementierungs-Details des Prototypen eingegangen.

1.5.2 Ausgabe eines Grafikobjektes

Die Aufgabe der Client/Server-Architektur ist es, statistische Plots zu visualisieren. Der statistische Plot wird als Grafikobjekt im Objekt-Server verwaltet. Alle Grafikobjekte werden von einer gemeinsamen CORBA-Schnittstelle abgeleitet, die die benötigten Methoden (siehe Tabelle 1.6) enthält. Den Ausgangspunkt für

Methode	Beschreibung
cloneObject	Erzeugung realer Objekte
destroy	Zerstörung des Objektes
name	Lieferung des Namens des Plots
draw	Realisierung der Ausgabe
setData	Übergabe der für den Plot notwendigen Daten

Tabelle 1.6: Methoden der Grafikobjekte

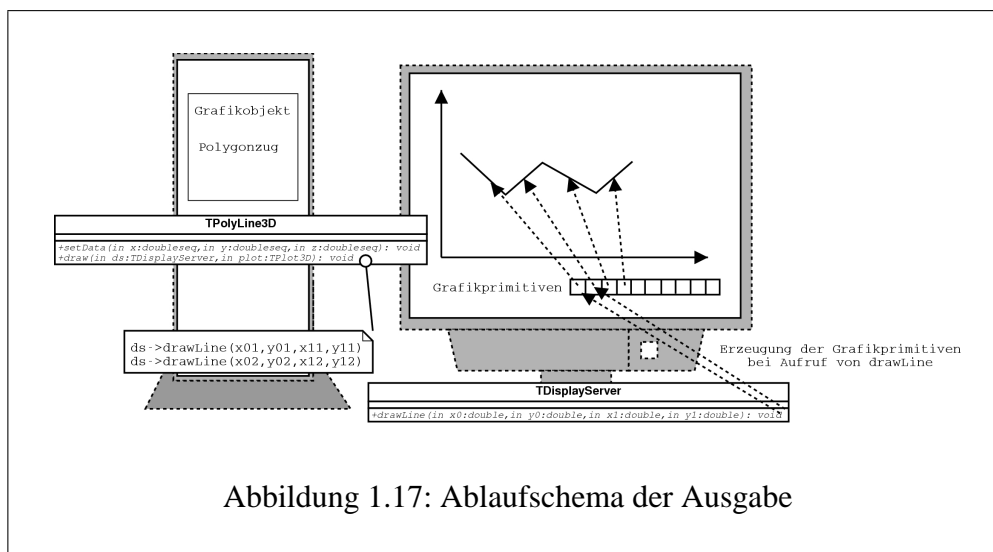
die Ausgabe bildet der Client. Er liefert die Daten der Ausgabe und bestimmt mit welchem Plot diese ausgegeben werden sollen. Der Client erzeugt in dem Objekt-Server ein Grafikobjekt, welches den Plot darstellt, was durch Klonen des Prototypen (siehe Anhang A.2, Seite 163) geschieht. Dabei können beliebig viele Grafikobjekte beim Objekt-Server angefordert werden, eine Begrenzung erfolgt nur durch das jeweilige Betriebssystem und die Middleware. Die Erzeugung der Grafikobjekte wird von einer Factory (siehe Anhang A.2, Seite 163) übernommen, danach muss der Datensatz an das Grafikobjekt übergeben werden, bzw. das Funktional für die Ausgabe spezifiziert werden. Damit ist dann der statistische Plot festgelegt. Außerdem muss der Client die Art und Größe des Weltraumes festlegen. Die Art wird durch die verwendeten Grafikobjekte festgelegt, die Größe

⁶¹gcc.gnu.org

⁶²www.mico.org

wird vom Ersteller des Clients frei gewählt. Ebenfalls müssen die Eigenschaften des Bildraumes gesetzt werden. Die Verwaltung des Welt- und Bildraumes sind in dem Plot-Objekt gekapselt⁶³. Jeder Display-Server bekommt sein eigenes Plot-Objekt des Objekt-Servers zugewiesen. Es enthält eine Referenz auf den Display-Server, und setzt so die Eigenschaften des Display-Servers.

Wenn vom Client alle Eigenschaften gesetzt worden sind, wird durch einen Methodenaufruf die Ausgabe eingeleitet. Es wird jetzt beispielhaft angenommen, dass ein 3D-Grafikobjekt visualisiert werden soll. Der Objekt-Server führt, bevor dieses Objekt ausgegeben wird, die Projektion durch (siehe Abschnitt 1.2.4.5, S. 13). Ebenso würden, falls notwendig, alle anderen Berechnungen wie Clipping⁶⁴, Rotation oder Entfernen von verdeckten Kanten und Linien, vor der Projektion durchgeführt. Dies sind meist sehr rechenintensive Operationen und entlasten den Display-Server, da sie im Objekt-Server ausgeführt werden. Nach den Berechnungen werden dem Display-Server, der über die Referenz im Plot-Objekt ermittelt wurde, die Daten des Plots übergeben. Die Daten bestehen aus den Grafikprimitiven, aus denen sich der statistische Plot zusammensetzt. Der



Display-Server speichert diese Grafikprimitiven in einer Liste (siehe Abb. 1.17) damit der spätere Zugriff auf diese gewährleistet ist, ohne dass die Daten neu

⁶³Nähere Informationen zur Umsetzung der Plot-Objekte sind dem Anhang (A.2.3) zu entnehmen.

⁶⁴Dies wird meistens vom Fenstermanager im Display-Server durchgeführt.

vom Objekt-Server angefordert werden müssen. Die Kommunikation zwischen Display- und Objekt-Server wird dadurch entlastet und rechtfertigt den Speicherverbrauch durch die Speicherung im Display-Server. Nachdem die Window-Viewport-Transformation bei allen Grafikprimitiven durchgeführt worden ist, werden diese ausgegeben. Falls nötig berechnet der Display-Server auch ein Koordinatensystem, und gibt dieses mit aus. Mit der Ausgabe der Grafikprimitiven auf dem Bildschirm endet die Ausgabe eines 3D-Grafikobjektes, welches einen dreidimensionalen statistischen Plot darstellen soll (siehe Abb. 1.18).

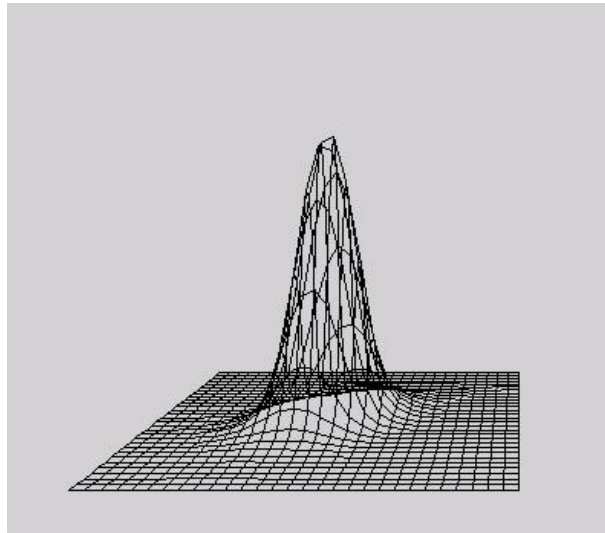


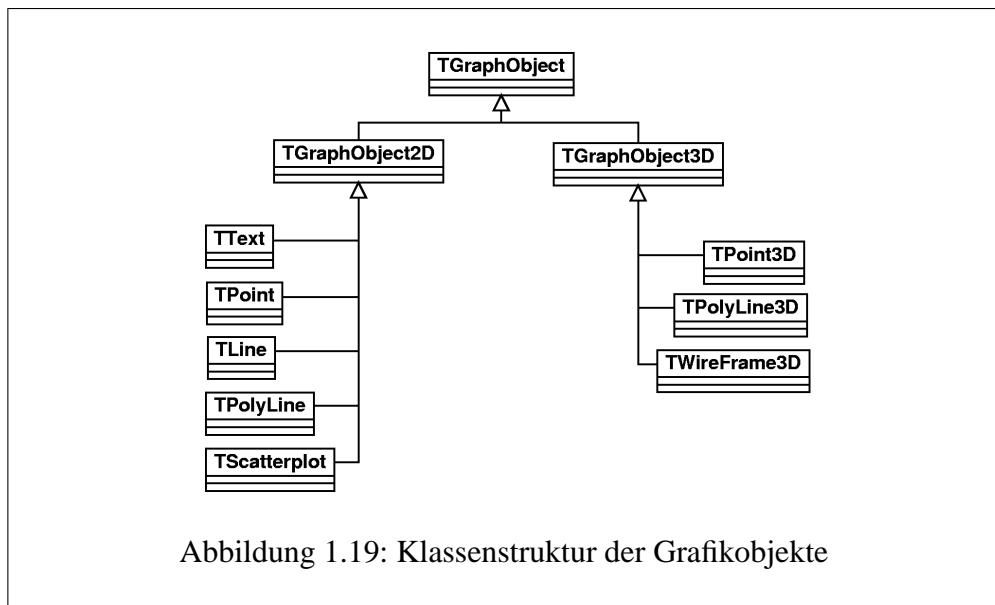
Abbildung 1.18: Ausgabe eines 3D-Plots

1.5.3 Erweiterung des Objekt-Servers um einen statistischen Plot

Die Erweiterbarkeit der Client/Server-Architektur und somit des Prototypen spielt eine wichtige Rolle. Im Prototypen der Architektur sind zwar schon einige statistischen Plots eingefügt, es sind aber nicht alle vorhanden, da es eine sehr große Anzahl an statistischen Plots gibt. Wenn ein Client einen Plot benötigt, der noch nicht vorhanden ist, sollte es möglich sein, die Client/Server-Architektur um einen sol-

chen zu ergänzen. Ausgangspunkt dafür bilden die Grafikobjekte, die in der Architektur den statistischen Plot repräsentieren.

Ein Grafikobjekt enthält eine CORBA-Schnittstelle und den entsprechenden Servant. Für einen neuen Plot muss zunächst eine eigene Schnittstelle erstellt werden. Da die Grafikobjekte in einer objektorientierten Struktur (siehe Abb. 1.19)



angelegt wurden, muss die Schnittstelle von der Elternschnittstelle⁶⁵ *TGraphObject2D* oder *TGraphObject3D* abgeleitet werden. Eine Schnittstelle könnte folgendermaßen aussehen:

Listing 1.1: Beispielschnittstelle für eine Grafikobjekt

```

interface TUnivariatePlot : TGraphObject2D {
    void setData(in doubleseq x, in doubleseq y);
};
  
```

Dies ist eine Schnittstelle für einen zweidimensionalen Plot, dem als Parameter zwei univariate Datensätze⁶⁶ übergeben werden. Alle anderen Methoden bekommt die Schnittstelle von *TGraphObject2D* vererbt (siehe Tabelle 1.6, S. 34).

Nach der Definition und dem Übersetzen der Schnittstelle durch den IDL-Compiler, muss im Servant jede einzelne Methode mit Leben gefüllt werden.

⁶⁵siehe dazu auch Anhang A.3, S. 166.

⁶⁶Definition von doubleseq, siehe Anhang A.3, S. 166.

Hier werden die statistischen Berechnungen für den neuen Plot untergebracht. Nachdem der neue Plot erstellt wurde, d. h. CORBA–Objekt und Servant erzeugt wurden, geht es darum, wie der neue Plot beim Objekt–Server angemeldet wird.

Der Objekt–Server erzeugt sämtliche Plots mit einer Factory (siehe Anhang A.2.1, S. 163), dabei wird jeder Plot als Prototyp bei dem Objekt–Server registriert (siehe Anhang A.2.2, S. 164). Mit Prototypen werden in diesem Abschnitt die statistischen Plots⁶⁷ bezeichnet, die zum Klonen benötigt werden. Die Prototypen werden alle in einer Liste im Objekt–Server gespeichert und sind über einen eindeutigen Namen⁶⁸ identifizierbar. Reale statistische Plots werden durch Klone der Prototypen erzeugt, und bekommen dann durch die Methode *setData* die Daten für die Berechnungen übergeben (siehe dazu auch Abschnitt 1.2). Die Registrierung beim Objekt–Server geschieht durch folgenden Aufruf:

Listing 1.2: Aufruf zur Registrierung eines Prototypen

```
registerPrototype(new NewPlotServant._this(orb));
```

Mit diesem Methodenaufruf wird der neue Plot ebenfalls beim Object Request Broker angemeldet, und steht nun als Grafikobjekt zur Verfügung. Dieser Aufruf kann an verschiedenen Stellen erfolgen. Zum einen kann der neue Plot im Objekt–Server registriert werden, d. h. der oben erwähnte Aufruf würde in das Listing des Objekt–Servers eingefügt werden. Danach muss der komplette Objekt–Server neu erstellt werden, dafür steht aber der neue Plot bei jedem Start des Objekt–Servers zur Verfügung. Dies sollte, falls möglich, vermieden werden. Als zweite Möglichkeit kann der Aufruf über den Client erfolgen. Dies hat den Vorteil, dass der Objekt–Server nicht neu erstellt werden muss, allerdings kann der neue statistische Plot nur im Zusammenhang mit diesem Client benutzt werden. Da der statistische Plot nur diesem Client zur Verfügung steht, ist diese eine sehr unflexible Möglichkeit.

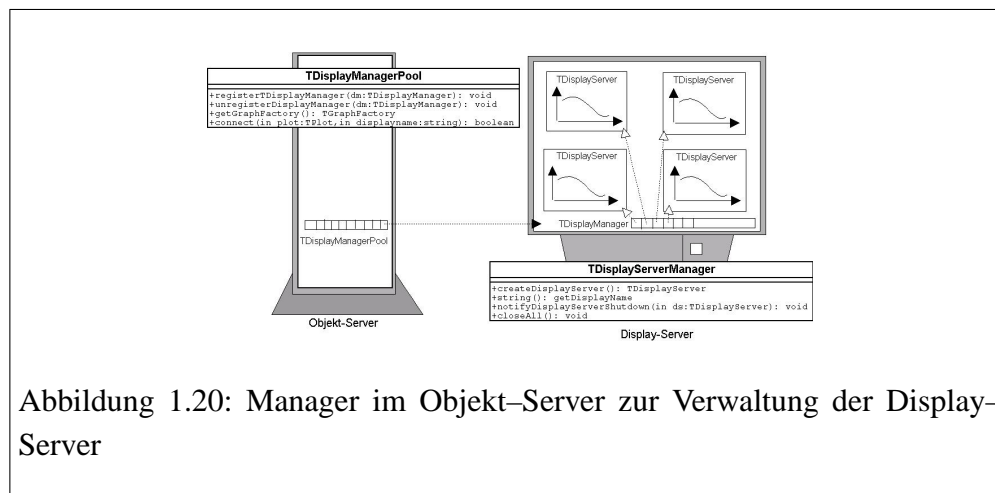
Die beste Möglichkeit ist es, einen separaten Server mit den zusätzlichen Grafikobjekten zu erstellen, und diesen in die Client/Server–Architektur einzubinden. Somit steht jedem Client die komplette Anzahl an Grafikobjekten zur Verfügung.

1.5.4 Manager zu Verwaltung: Beispiel Ausgabefenster

Manager werden im Prototypen der Architektur sehr häufig verwendet. Sie haben

⁶⁷Sie sind Grafikobjekte ohne Daten.

⁶⁸Der Name des statistischen Plots kann über die Methode *name()* erfragt werden.



die Aufgabe Objekte zu verwalten. Diese Objekte haben alle ähnliche Eigenschaften, da sie von der gleichen Klasse oder der gleichen Schnittstelle abgeleitet sind. Nachdem ein solches Objekt erzeugt wurde, wird es beim Manager angemeldet. Der Manager hat jetzt die Aufsicht über das Objekt bekommen, d. h. nur über den Manager wird auf das Objekt und dessen Methoden zugegriffen. Dies hat zum einen den Vorteil, dass die Objekte nicht ohne weiteres gelöscht werden können, es kommt nicht zu Zugriffen auf nicht vorhandene Objekte. Zum anderen können die Methodenaufrufe der Objekte durch den Manager kontrolliert werden. Außerdem kann durch den Manager ein Methodenaufwurf an alle Objekte weitergegeben werden, es muss also nicht jedes Objekt einzeln angesprochen werden. Soll ein Objekt auch über einen Manager erzeugt werden, so wird der Manager „Factory“ genannt. Die Factory⁶⁹ enthält die gleichen Methoden wie der Manager, zusätzlich aber noch mindestens eine Methode zur Erzeugung eines Objektes. Mit dem Einsatz einer Factory gibt es keine Möglichkeit mehr, auf ein Objekt direkt zuzugreifen⁷⁰. Die Umsetzung einer Factory bzw. Manager wird im Anhang A.2 (Seite 163) als Entwurfsmuster⁷¹ beschrieben.

Eingesetzt wird ein Manager zum Beispiel bei der Verwaltung der Display-Server im Objekt-Server. Er hat die Aufgabe alle Referenzen auf die Display-Server abzuspeichern, und bei Bedarf eine Referenz auf einen Display-Server

⁶⁹Die Factory soll in diesem Fall auch die Zugriffe auf die Objekte kontrollieren.

⁷⁰Bei dem Manager gibt es eine direkte Zugriffsmöglichkeit auf die Objekte bei deren Erzeugung.

⁷¹engl. Design-Pattern

auszugeben (siehe Abb. 1.20). Damit keine leeren Referenzen ausgegeben werden, überprüft der Manager zuvor, ob der Display-Server noch existiert. Dies führt der Manager bei allen angemeldeten Display-Servern durch, damit keine Ausgabe ins Unbekannte geht. Um dem Manager diese Überprüfung zu erleichtern, sollten alle Display-Server dort auch wieder abgemeldet werden, was aber erst dann geschehen sollte, wenn die Display-Server nicht mehr benötigt werden und zerstört werden können. Der Manager überwacht mit den Referenzen die Zugriffe auf den Display-Server also die Kommunikation zwischen Objekt-Server und Display-Server.

1.5.5 Events

Events sind der Ausgangspunkt für die Interaktion durch den Benutzer. Durch ein Event kann der normale Programmablauf unterbrochen werden, um spezielle Aufgaben auszuführen, z. B. Reaktionen auf die Interaktion des Benutzers. In der Client/Server-Architektur bilden die Events die Grundlage für die Kommunikation zwischen Display-Server und Objekt-Server. Normalerweise greift der Objekt-Server immer nur auf den Display-Server zu, aber nicht umgekehrt.

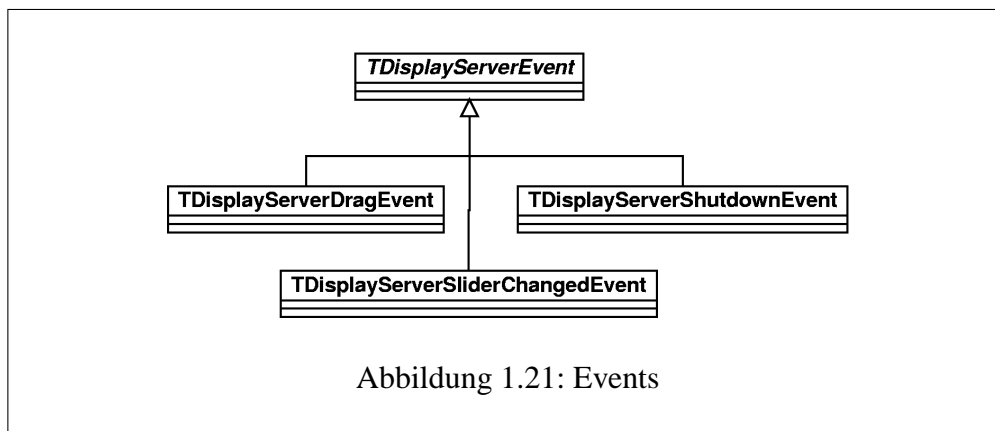
Tritt im Display-Server eine Aktion auf, die einen Event auslöst, so kann er dies über eine Methode dem Objekt-Server mitteilen (siehe Listing 1.3).

Listing 1.3: Event-Methode

```
void notifyDisplayServerEvent(in TDisplayServerEvent ev)
```

In dieser Methode wird dem Objekt-Server ein Event-Objekt übergeben. Dieses Event-Objekt ist von dem Event *TDisplayServerEvent* abgeleitet worden (siehe Abb. 1.21). In Tabelle 1.7 sind alle vorhandenen Events aufgelistet. Durch die Vererbung kann dem Objekt-Server über diese Methode jedes Event mitgeteilt werden. Mit einer Typüberprüfung kann der Objekt-Server feststellen, um welches Event es sich handelt und die entsprechende Aktion einleiten. Wird im Display-Server bei einem 3D-Plot zum Beispiel die Maus bei gedrückter Taste bewegt, so wird ein *TDisplayServerDragEvent* an den Objekt-Server übergeben. Der Objekt-Server stellt fest, um welches Event es sich handelt und weiß jetzt, dass er die Berechnungen für eine Rotation durchzuführen hat⁷². Die notwendigen Parameter für die Berechnung sind als Attribute im Event-Objekt enthalten. Dies können zum

⁷²Diese Verhalten wurde im Objekt-Server so umgesetzt.



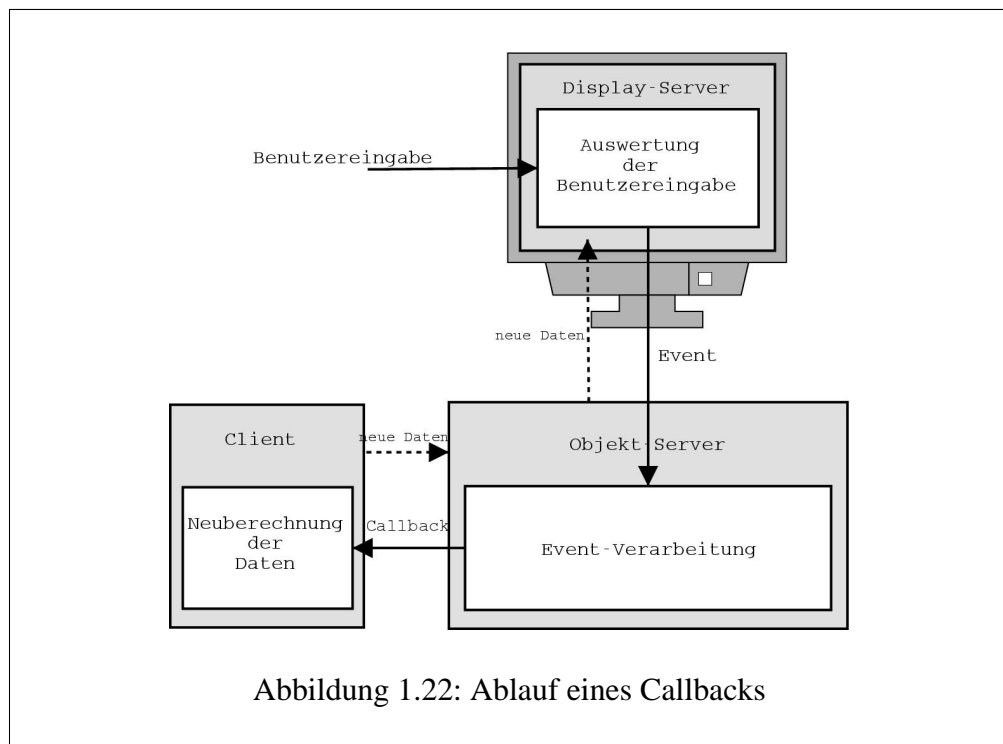
Event	Beschreibung
TDisplayServerEvent	Von dieser Klasse werden alle anderen realen Event-Klassen abgeleitet.
TDisplayServerShutdownEvent	Event, welches ausgelöst wird, wenn der Display-Server geschlossen wird.
TDisplayServerDragEvent	Event, welches ausgelöst wird, wenn die Maus bei gedrückter Maustaste bewegt wird.
TDisplayServerSliderChangedEvent	Event, welches ausgelöst wird, wenn der Slider bewegt wird.

Tabelle 1.7: Events

Beispiel die Koordinaten des Zoomfensters oder der Wert eines Sliders sein. Die Berechnungen für die Rotation werden ausgeführt, und auf dem Display-Server erscheint die gedrehte Ausgabe des Plots.

1.5.6 Callback

Bei statistischen Modellen spielt die Parameterwahl eine wichtige Rolle. Mit der Änderung eines Parameters ändert sich auch das Aussehen eines statistischen Plots. Möchte man dem Benutzer die Möglichkeit geben, dass er Parameter eines Modells ändern kann, so muss der Display-Server die Änderung eines Parameters über eine Eingabemöglichkeit bereitstellen, d. h. die Änderung eines Parameters



muss dem Client mitgeteilt werden, um die Daten mit dem neuen Parameter berechnen zu können, und diese wieder dem Display-Server zur Ausgabe mitzuteilen. Dies geschieht, indem er im Objekt-Server ein neues Grafikobjekt erzeugt und das alte zerstört.

Der Client bekommt die neuen Parameter über ein Callback vom Objekt-Server geliefert. Das Callback wird durch den Objekt-Server erzeugt, nachdem er vom Display-Server ein Event erhalten hat. Der Objekt-Server entscheidet anhand des Events vom Display-Server, welchen Parameter er an den Client übergeben muss, dieser wird bei der Erstellung des Clients festgelegt. Außerdem muss der Client so erweitert werden, dass er bezüglich des Callbacks als Server agieren kann. Er muss die Daten für das neue Grafikobjekt berechnen, welches die Grundlage der Ausgabe bildet. Außerdem muss der Client dem Display-Server mitteilen, dass er eine Benutzerschnittstelle benötigt, z.B. einen Slider. Mit diesem Slider⁷³ kann der Anwender dann den Parameter in einem festgelegten Bereich frei wählen. Durch diese Bereitstellung kommt es erst zur Interaktion. Der

⁷³dt. Schieberegler

Objekt-Server bekommt von Client mitgeteilt, dass er auf die Benutzeraktionen reagieren soll, und diese, wenn notwendig, als Callback an den Client weitergeben soll. Für diese Kommunikation müssen spezielle Schnittstellen bereitgestellt werden, da der Client in diesem Fall ebenfalls ein Server ist und Methodenaufrufe abarbeiten muss. Damit der Client als Server agieren kann, muss er einen Servant bzgl. des Callbacks enthalten. Außerdem benötigt der Client einen laufenden Object Request Broker, damit die Anfragen vom Objekt-Server bearbeitet werden können.

Durch den Einsatz von Events ist es möglich, die Visualisierungen interaktiv zu gestalten. Aber erst durch Callbacks kann ein Benutzer multimedial Einfluss auf die erzeugten und visualisierten Daten nehmen, es können Parameter der Funktionale geändert werden. Mit dem Einsatz von Callbacks werden statistische Modelle greifbarer und anschaulicher. Durch den Einsatz von Callbacks kann die Client/Server-Architektur im Bereich E-Learning zum Einsatz kommen.

1.5.7 Umsetzung von Farben und Linienstilen

Jede Grafikprimitive bekommt ein zusätzliches Attribut, welches die Farben und den Linienstile beinhaltet. Das Attribut der Farben besitzt den Type **TColor**, welches in der IDL-Schnittstelle definiert wurde. Es hat folgenden Aufbau:

Listing 1.4: Interface TColor

```
interface TColor {
    readonly attribute short red;
    readonly attribute short green;
    readonly attribute short blue;
    readonly attribute short alpha;
    void setAlpha (in short pAlpha);
    void setR(in short mred);
    void setG(in short Gred);
    void setB(in short Bred);
    void setRGB(in short r, in short g, in short b);
    long getRGB(); // returns rgb-Value with alpha=255(opaque)
    long getARGB(); // returns argb-Value
    void destroy ();
};
```

Das Attribut verwaltet die Farben im RGB-Farbsystem, dieses wurde gewählt, weil auch dieses Farbsystem dem Monitor zu Grunde liegt. Die Farben Rot, Grün

und Blau können dabei einen ganzzahligen Wert im Bereich zwischen 0 und 255 annehmen. Zusätzlich gibt es noch eine Möglichkeit die Deckung der Farben zu spezifizieren. Damit können Objekte, die hinter anderen verdeckt sind, sichtbar gemacht werden. Dies geschieht indem der Parameter alpha gesetzt wird. Vollkommend deckend sind Grafikprimitiven für $\alpha = 255$, je näher alpha an null liegt, desto transparenter werden diese. Für die Linienstile wurde kein eigenes Interface spezifiziert, sondern eine zusätzliche Methode den Grafikobjekten hinzugefügt, mit denen der Linienstil gesetzt werden kann.

Listing 1.5: Interface TLine

```
interface TLine : TGraphObject2D {
    void setLineStyle (in octet pLineStyle);
    void setData (in double x0, in double y0,
                 in double x1, in double y1);
};
```

Dies geschieht durch das Setzen eines ganzzahligen Wertes von Typ Byte. Dieser Wert wird vom Display-Server entsprechend interpretiert und der Linienstil wird dann dementsprechend gesetzt.

Kapitel 2

Finanzdatenanalyse

2.1 Einführung

Aktien, Fonds und Optionen sind bekannte Anlagen. Sie gehören zu den mit Risiko behafteten Finanzwerten. Investiert man in diese Finanzwerte, können sowohl große Gewinne als auch hohe Verluste erzielt werden. Da die Gewinne sehr reizvoll sind, wird versucht durch eine Strategie das Risiko zu verringern. Risiko bedeutet, dass der Zeitpunkt und die Höhe des Verlustes¹ nicht bekannt sind. Zur Risikominderung gibt es verschiedene Strategien:

- Eine mögliche Strategie ist es einen Finanzwert zu finden, der mit dem entgegengesetzten Risiko behaftet ist. Meist sind dies künstlich erschaffene Finanzwerte². Diese Vorgehensweise wird als Hedging³ bezeichnet. Durch den Kauf einer dieser Finanzwerte wird das Risiko des Verlustes auf null gesetzt, allerdings kann ebenso kein Gewinn erzielt werden (siehe Schierenbeck und Wiedemann [86]). Diese Strategie wird in dieser Arbeit nicht näher betrachtet.
- Eine zweite mögliche Strategie ist die Strategie der Portfolio Selection, die auf Markowitz zurück geht. Bei der Diversifikation soll das Risiko auf verschiedene Finanzwerte verteilt werden. Es wird nicht nur ein Finanzwert über einen bestimmten Zeitraum gehalten, sondern es wird ein Portfolio an

¹Ebenso sind die Zeitpunkte und Höhe der Gewinne unbekannt.

²z. B. Collar, Cap, Floor.

³Definition **Hedging**: Verringerung des Risikos durch negativ korrelierte Einzelpositionen. Die Risiken der einen Position werden durch die Chancen der anderen teilweise kompensiert [30].

Finanzwerten aufgebaut. Das Portfolio soll durch die Gewichtung der einzelnen Finanzwerte so strukturiert werden, dass das Gesamtrisiko verringert (minimiert) wird (siehe Markowitz [60]). Wichtig ist bei dieser Methode die Bestimmung und die Beschreibung des zufälligen Risikos. Es müssen die einzelnen Risiken und deren Abhängigkeit zueinander analysiert werden.

In den weiteren Abschnitten wird daher ein grundlegendes mathematisches Modell zur Beschreibung der Finanzwerte vorgestellt. Es werden die Eigenschaften von Finanzwerten erklärt und diese in Bezug zu den mathematischen Modellen gesetzt. Die Modelle werden dann so abgeändert, dass sie immer stärker der Realität entsprechen, d.h. die Stylized Facts⁴ der Finanzdaten wiedergeben. Anschließend wird auf die Bestimmung des Risikos eingegangen.

⁴Diese werden später eingeführt.

2.2 Allgemeine Modellbildung

Folgendes Modell bildet die Grundlage für die spätere Risikokalkulation und allen weiterentwickelten Modellen. Wir betrachten einen Zufallsvektor S_t

$$S_t = (S_{t,1}, \dots, S_{t,d})$$

zum Zeitpunkt t . Dieser Zufallsvektor beschreibt den Wert des Portfolio mit d Finanzwerten zum Zeitpunkt t . Wird von Realisationen gesprochen, so sind die vergangenheitsbezogenen Kurswerte gemeint. Werden in einem Modell mehrere Finanzwerte behandelt, so werden sie in einem multivariaten Modell betrachtet. Ist nur ein Finanzwert vorhanden, liegt ein univariates Modell zu Grunde. Die historischen Daten zu einem Portfolio werden ausgedrückt durch

$$s_t = (s_{t,1}, \dots, s_{t,d}), \quad t = -n, \dots, 0. \quad (2.1)$$

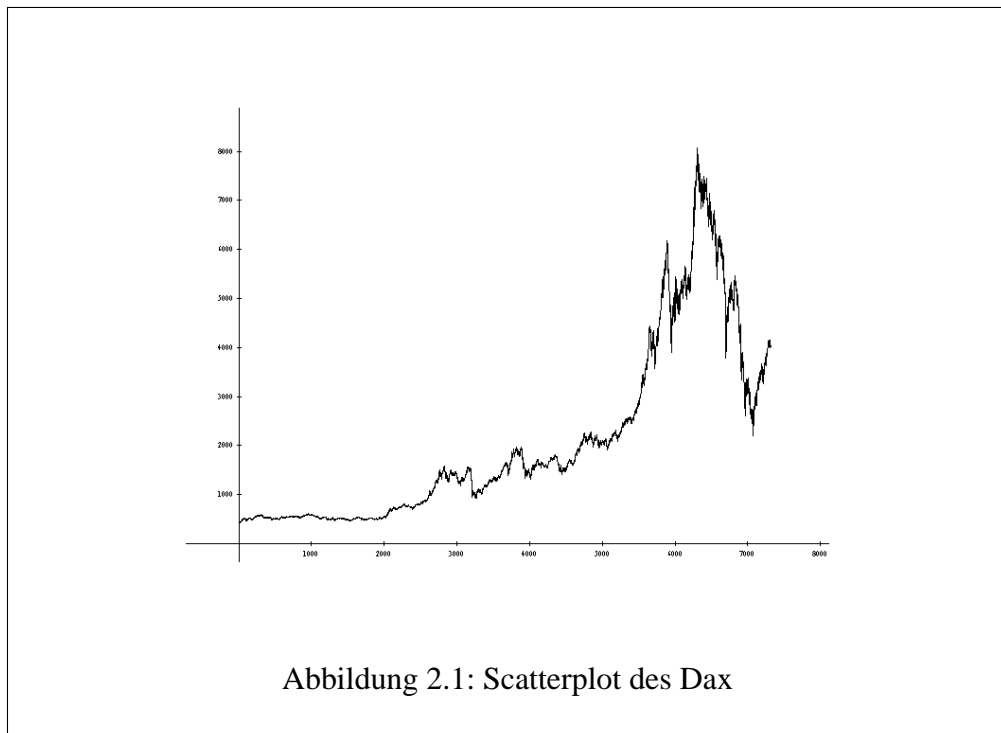
Das Betrachtungsintervall läuft von n Tage aus der Vergangenheit bis zum heutigen Zeitpunkt. Zur Vereinfachung wird der Zeitraum auf positive Werte transformiert: $t = 0, \dots, n$. Aus diesen Daten soll das Risiko des Finanzwertes bzw. des Portfolios für die Zukunft geschätzt werden. Sei nun S_t der Zufallsvektor der Kurswerte eines Aktienportfolios. Somit sind die $s_{t,i}$, $i = 1, \dots, d$, die vergangenen Kurswerte, die Realisationen der Aktien im Portfolio (einer Zufallsvariable S_t). Folglich können die Kurswerte einer Aktie, welche den Risikofaktor der Aktie widerspiegeln, als Zeitreihe in einem Scatterplot ausgegeben werden. Dabei sind die Kurswerte in Abhängigkeit zur Zeit geplottet. Der Zusammenhang zwischen Risiko und Kurswert einer Aktie wird später erklärt.

Im Folgenden werden spezielle Eigenschaften von Finanzdaten vorgestellt und erklärt, bezeichnet werden sie als Stylized Facts. Diese Stylized Facts sollten bei der Entwicklung von neuen Modellen für Finanzdaten umgesetzt werden, da sie die Eigenarten der Finanzdaten beschreiben.

2.2.1 Stylized Facts

Den Ausgangspunkt für die statistische Modellierung bilden die historischen Daten in Form von z. B. Aktienkursen, Zinssätzen oder Wechselkursen. Zunächst wird ein einzelner Finanzwert $S_{t,1}$ betrachtet, d. h. gegeben sind die historischen Daten $\{s_{t,1}\}$, $t = 1, \dots, n$. Werden diese historischen Daten⁵ in einem Scatterplot

⁵In diesem Abschnitt wird der DAX als Datensatz für alle empirischen Berechnungen verwendet.



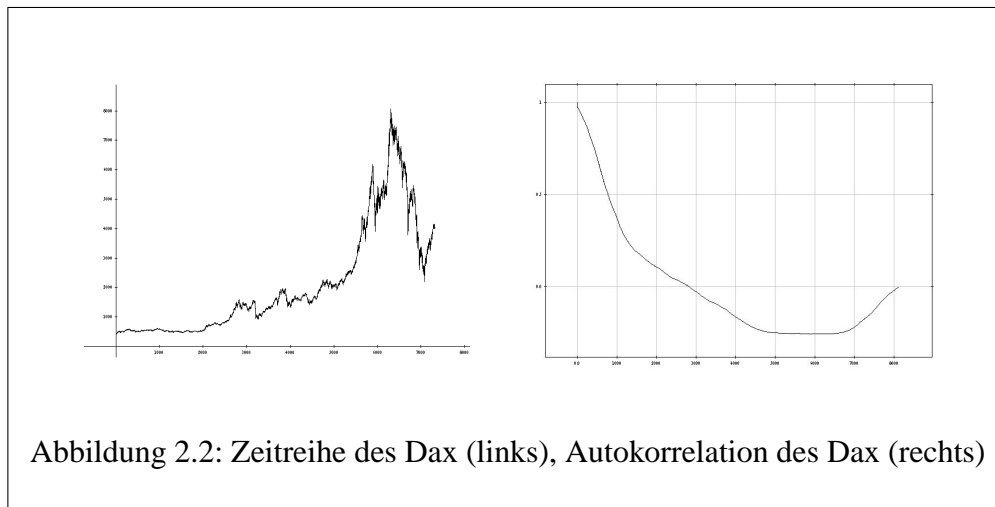
als Zeitreihe geplottet, so stellt man fest, dass der heutige Wert von den vergangenen Werten abhängt (siehe Abb. 2.2 (links)). Die Kurswerte der Aktie sollen das Risiko dieses Finanzwertes wiedergeben. Meist wird die Volatilität⁶ des Finanzwertes als Risikofaktor betrachtet. Berechnet man die Autokorrelationsfunktion der Kurswerte, stellt man eine starke Zeitabhängigkeit fest (siehe Abbildung 2.2 (rechts)). Diese Erkenntnis führt zum ersten Stylized Fact.

Stylized Fact 1:

Der Werteverlauf der Kurswerte ist stark autokorreliert.

Es ist immer von Bedeutung, zu welchem Zeitpunkt die Realisation auftritt. Durch diese Abhängigkeit kann die zu Grunde liegende Verteilung nur sehr schwer bestimmt (geschätzt) werden. Um dieses Problem zu umgehen, werden nicht die absoluten Werte betrachtet, sondern die relativen Änderungen dieser Werte. Dies führt zu dem Begriff der Renditen. Renditen werden folgendermaßen beschrieben:

⁶Varianz



- *arithmetische Rendite*

$$\tilde{R}_t = \frac{S_t - S_{t-1}}{S_{t-1}} \quad (2.2)$$

- *logarithmische Rendite*⁷

$$R_t = \log\left(\frac{S_t}{S_{t-1}}\right) = \log(S_t) - \log(S_{t-1}) \quad (2.3)$$

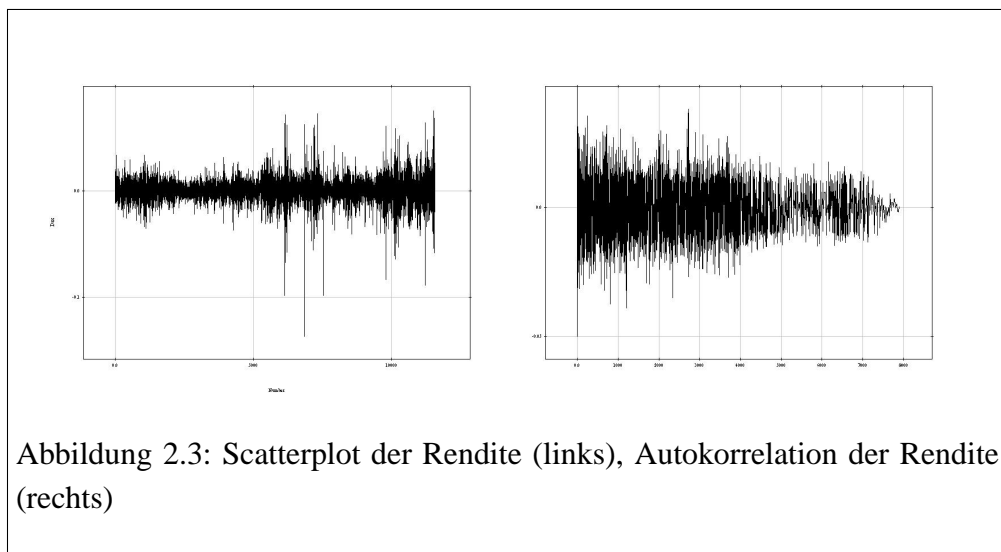
Für kleine Differenzen zwischen S_t und S_{t-1} unterscheiden sich die beiden Renditen kaum, denn es gilt folgender Zusammenhang:

$$R_t = \log\left(\frac{S_t}{S_{t-1}}\right) = \frac{S_t - S_{t-1}}{S_{t-1}} + o\left(\left(\frac{S_t}{S_{t-1}} - 1\right)\right) \text{ für } \frac{S_t}{S_{t-1}} \rightarrow 1$$

In der Realität ist die Differenz zweier Kurswerte benachbarter Tage sehr klein, deshalb kann man die Änderungen durch log-Renditen ausdrücken. Die Renditen stellen somit den Risikofaktor⁸ der Aktie dar. Empirische Untersuchungen haben gezeigt, dass die Renditen nahezu unkorreliert in Abhängigkeit der Zeit sind, was durch die Autokorrelationsfunktion der Renditen des Finanzwertes ausgedrückt wird. Es ist zu beachten, dass die Autokorrelationsfunktion nur die lineare Abhängigkeit bzgl. der Zeit beschreibt. Gewinn und Verlust treten zufällig

⁷kurz: log-Rendite. Sie wird auch als geometrische Rendite bezeichnet.

⁸Dies wird später noch erklärt.



auf, unabhängig von der Vergangenheit. Durch die Unkorreliertheit ist es leichter möglich, die zu Grunde liegende Verteilung der Renditen zu bestimmen. Über die Verteilung kann das Risiko leichter geschätzt werden. Es wird meistens durch die Varianz der Verteilung⁹ geschätzt.

Stylized Fact 2:

Die Renditen sind nahezu unkorreliert.

Betrachtet man Renditen genauer, so stellt man eine weitere Eigenschaft von Finanzwerten fest. Die Volatilität der Renditen ist nicht konstant, sondern es gibt Phasen starker Volatilität, die sich mit Phasen schwacher Volatilität abwechseln. In der Realität entspricht dies den Ruhephasen und den Phasen der starken Volatilität des Marktes.

Stylized Fact 3:

Renditen treten in Volatilitätsclustern auf.

Die Renditen bilden den Risikofaktor¹⁰ der Wertpapiere, durch sie soll das Risiko bestimmbar sein. Dies geschieht meist durch ein parametrisches Modell, welches an die zu Grunde liegenden Renditen angepasst wird. Die Grundlage dafür ist häufig die Annahme der Normalverteilung. Dieses Modell wird, auf Grund der

⁹Die Verteilung ist die später eingeführte Gewinn+Verlust-Verteilung.

¹⁰Dieser wird durch die Varianz ausgedrückt.

leichten Handhabbarkeit, bevorzugt, da sich die Parameter leicht schätzen lassen. Außerdem hat das Normalverteilungsmodell wichtige Eigenschaften. Eine davon wird durch den Wert der Kurtosis repräsentiert:

Definition 4 (Kurtosis). Die Kurtosis einer Zufallsvariable X ist definiert durch

$$\delta(X) = \frac{E((X - EX)^4)}{\text{Var}(X)^2}, \quad (2.4)$$

falls die ersten vier Momente existieren. $\text{Var}(X)$ ist die Varianz der Zufallsvariable X und es gilt $\text{Var}(X) = E((X - EX)^2)$.

Die Kurtosis ist invariant bzgl. Transformationen mit Lokations- und Skalenparameter. Diese Eigenschaft hilft bei der Berechnung der Kurtosis. Falls das Normalverteilungsmodell zu Grunde gelegt wird, gilt für die Kurtosis:

- **Kurtosis der Normalverteilung:** Sei $\mathcal{L}(X) = N_{\mu, \sigma^2}$ und $\mathcal{L}(Y) = N_{0,1}$, dann gilt für die Kurtosis

$$\begin{aligned} \delta(X) &= \delta(Y) \\ &= 3, \end{aligned} \quad (2.5)$$

denn es gilt

$$\begin{aligned} E(Y) &= 0, \\ E(Y^2) &= 1, \\ E(Y^4) &= 3. \end{aligned}$$

- Die Kurtosis bezüglich der Normalverteilung kann also neu definiert werden als

Definition 5 (Kurtosis bzgl. der Normalverteilung). Die Kurtosis bzgl. der Normalverteilung k ist definiert als

$$k = \delta - 3 \quad (2.6)$$

für den Schätzer folgt

$$\hat{k} = \hat{\delta} - 3 \quad (2.7)$$

Für die Normalverteilung ist also die Kurtosis $k = 0$ bzw. $\delta = 3$. Mit der Berechnung der Kurtosis kann somit die Abweichung einer empirischen Verteilung von der Normalverteilung beschrieben werden. Für Finanzdaten kann empirisch nachgewiesen werden, dass diese eine Kurtosis von $k > 0$ bzw. $\delta > 3$ besitzen.

Definition 6 (Schätzer der Kurtosis). Die Kurtosis aus Definition 4 kann geschätzt werden durch

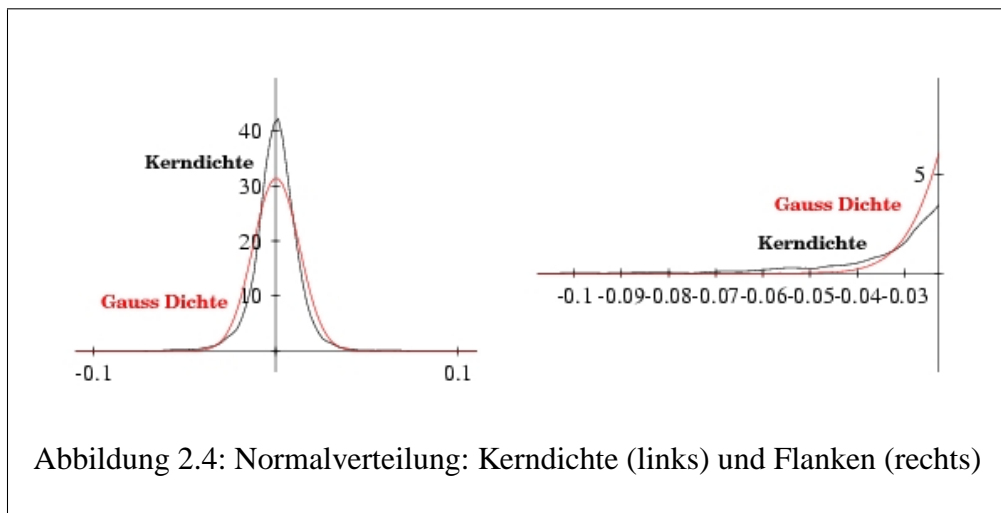
$$\hat{\delta} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})^4 / \hat{\sigma}^4, \quad (2.8)$$

welches die **empirische Kurtosis** für eine Stichprobe $x_i, i = 1, \dots, N$ ist. Dabei bezeichnet $\hat{\mu}$ den empirischen Mittelwert und $\hat{\sigma}^2$ die empirischen Varianz.

Dies bedeutet, dass die Verteilung der Finanzdaten stärkere Tails (Flanken) und einen höheren Gipfelpunkt als die Normalverteilung haben, den Stylized Fact 4. Die Masse in den Flanken ist stärker, wie es die Normalverteilung wiedergeben kann.¹¹

Stylized Fact 4:

Die Verteilungen von (log-) Renditen sind leptokurtisch.



¹¹Die Verteilung ist dann leptokurtisch. Dies bedeutet, dass die Verteilung stärker Tails und einen höheren Gipfel als die Normalverteilung hat.

Dieser Stylized Fact zeigt auf, dass durch das Normalverteilungsmodell das Risiko in den Flanken unterschätzt wird, da sich die hohen Gewinne und Verluste in den Flanken befinden. Da die Normalverteilung in der Flanke immer unter der zu Grunde liegenden Verteilung¹² liegt, wird immer eine geringere Wahrscheinlichkeit für das Eintreten eines Verlustes berechnet werden. Dies wird als geringeres Risiko gedeutet¹³. Um diese Schwäche zu umgehen, müsste als Modell eine Verteilung benutzt werden, welche stärkere Flanken besitzt, wie zum Beispiel die Studentsche Verteilung¹⁴.

Weitere Stylized Facts, die empirisch begründet werden, sind:

Stylized Fact 5:

Die Verteilung der (log-) Renditen hat einen Erwartungswert von $\mu = 0$.

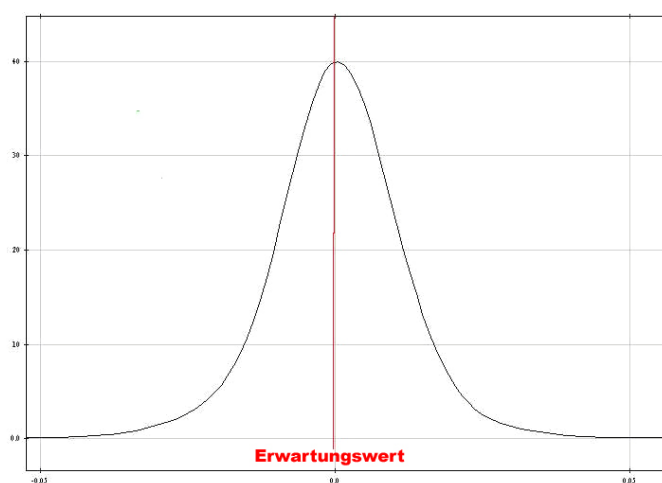


Abbildung 2.5: Kerndichte mit Erwartungswert

¹²siehe Abb. 2.4 (links): Es wird die Dichte betrachtet.

¹³Bei der Verwendung des Normalverteilungsmodell wird das berechnete Risiko mit einem festgelegten Faktor multipliziert, meist mit Faktor drei.

¹⁴Sie wird auch als t-Verteilung bezeichnet.

Stylized Fact 6:

(log-) Renditen haben eine näherungsweise symmetrische Verteilung, d. h.
 $\gamma(X) = 0$

Definition 7 (Schiefe (Skew)). Für eine Zufallsvariable X ist die Schiefe definiert durch

$$\gamma(X) = \frac{E((X - EX)^3)}{\sqrt{\text{Var}(X)}^3}. \quad (2.9)$$

Definition 8 (Schätzer für die Schiefe). Für eine Stichprobe $x_i, i = 1, \dots, n$, kann die Schiefe empirisch folgendermaßen berechnet werden:

$$\hat{\gamma} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{\mu})^3 / \hat{\sigma}^3. \quad (2.10)$$

Die bisher vorgestellten Stylized Facts treten bei den meisten Finanzdaten auf. Betrachtet man mehrere Finanzwerte eines Marktes, z. B. mehrere Aktien¹⁵ des Dax, so stellt man in diesem Portfolio von Finanzwerten fest¹⁶, dass gemeinsame Extrema auftreten, d. h. die Finanzwerte stehen in Abhängigkeit zueinander. Diese Eigenschaft wird als Flankenabhängigkeit¹⁷ bezeichnet (siehe Abb. 2.6).

Stylized Fact 7:

Es besteht die Tendenz, dass gemeinsame Extremwerte auftreten. Das Phänomen wird als Flankenabhängigkeit bezeichnet.

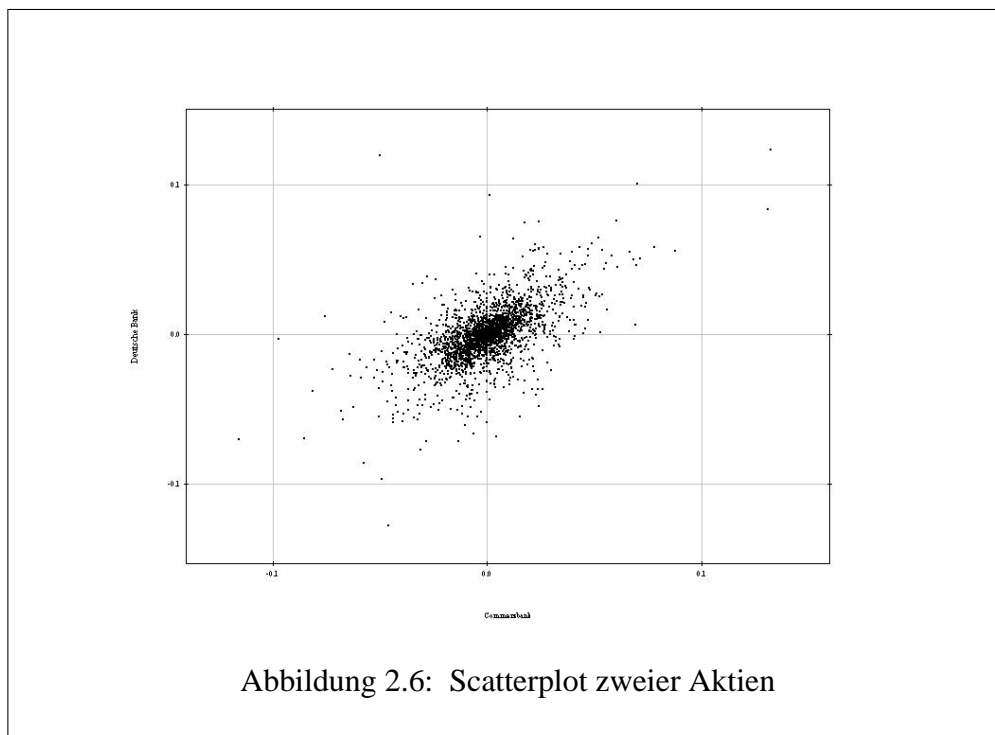
Dieser Stylized Fact ist besonders wichtig, wenn ein Portfolio von Finanzwerten betrachtet wird. Außerdem kann beobachtet werden, dass größere Verluste öfter als sehr hohe Gewinne auftreten.

Die Eigenschaften der Finanzwerte sind nun vorgestellt. Der folgende Abschnitt behandelt nun, wie das Risiko von Finanzwerten bestimmt werden kann.

¹⁵hier: Aktien der Commerzbank und der Deutschen Bank

¹⁶multivariate Modelle

¹⁷Tail Dependence



2.3 Moderne Risikoanalyse

Ziel der Risikoanalyse ist, das Risiko eines Finanzwertes oder eines Portfolios zu bestimmen. Nach diesem Schritt kann bei einem Portfolio anhand der Risikoanalyse die Zusammensetzung so geändert werden, dass das Risiko verringert werden kann¹⁸. Wie dabei das Risiko geschätzt wird, wird zunächst noch nicht betrachtet. Ausgangspunkt der Risikoanalyse sind die Renditen, mit ihnen soll die Gewinn- und Verlustverteilung¹⁹ geschätzt werden. Für die Schätzung des Risikos wird zunächst von einem Finanzwert, einer Aktie, ausgegangen. Dabei besteht folgender Zusammenhang zwischen den Preisen S_T , S_0 und den log-Renditen R_1, \dots, R_T :

$$S_T = S_0 \exp \left(\sum_{t \leq T} R_t \right). \quad (2.11)$$

¹⁸Es wird angenommen, dass das Portfolio noch keine optimale Zusammensetzung hat.

¹⁹kurz: P+L-Verteilung

Stylized Fact	Bedeutung
Univariat	
Fact 1	Kurswerte sind stark autokorreliert.
Fact 2	Renditen sind nahezu unkorreliert.
Fact 3	Renditen treten in Volatilitätsclustern auf.
Fact 4	Verteilung der (log-)Renditen ist leptokurtisch.
Fact 5	Verteilung der (log-)Renditen hat einen Erwartungswert $\mu = 0$.
Fact 6	(log-) Renditen haben eine symmetrische Verteilung.
Multivariat	
Fact 7	Tendenz für gemeinsame Extremwerte.

Tabelle 2.1: Stylized Facts

Es werden nun unter Vernachlässigung von Effekten²⁰, die durch Schließung der Börse hervorgerufen werden, die täglichen Preise und Renditen betrachtet. Seien die $\{R_t\}$, $t \in \mathbb{Z}$ eine Zeitreihe von Renditen, dann soll gelten

$$E(R_t | S_{t-1}, \dots, S_0) = 0. \quad (2.12)$$

Diese Unkorreliertheit wird in den meisten Fällen auch durch die empirischen Daten bestätigt. Die Renditen geben nur die relativen Änderungen einer Aktie an. Das Risiko soll aber für eine bestimmte Stückzahl einer Aktie bzw. für eine Portfolio geschätzt werden. Dies führt zur Einführung der Gewinn- und Verlustverteilung.

2.3.0.1 Gewinn+Verlust-Verteilung (Reiss und Thomas [73])

Seien V_0 und V_T die Marktpreise eines einzelnen spekulativen Gutes (Finanzwert) oder eines Portfolios zum Zeitpunkt $t = 0$ und $t = T$ (t wird meist in Tagen angegeben). Der Gewinn oder der Verlust für eine feste Periode $[0, T]$ wird dann ausgedrückt durch die Verlust- (Gewinn/Verlust-)Variable

$$L_T = -(V_T - V_0). \quad (2.13)$$

²⁰siehe Reiss und Thomas [73], Seite 311

Dabei ist zu beachten, dass der Verlust ein positiver Wert ist. Der Marktpreis kann ebenfalls durch die Renditen ausgedrückt werden, da gilt

$$R_{(T)} := \sum_{t \leq T} (-R_t). \quad (2.14)$$

Damit kann der Verlust für einen einzelnen Finanzwert zum Zeitpunkt T geschrieben werden als

$$\begin{aligned} L_T &= V_0(1 - \exp(-R_{(T)})) \\ &\approx V_0 R_{(T)}. \end{aligned} \quad (2.15)$$

Der Marktpreis V_t setzt sich aus dem Produkt der Stückzahl h_t und des Preises der Aktie S_t zum Zeitpunkt t zusammen:

$$V_t = h_t S_t. \quad (2.16)$$

Für einen Portfolio wird der Verlust dann angegeben durch

$$V_t = \sum_j V_{t,j}, \quad (2.17)$$

dabei ist $V_{t,j} = h_j S_{t,j}$ der Wert des j -ten Finanzwertes im Portfolio.

Der Verlust bzw. der Gewinn eines Portfolios kann auch noch anders geschrieben werden. Seien $\mathbf{w} = (w_1, \dots, w_d)$ Gewichte, die berechnet werden mit $w_j = \frac{V_{0,j}}{V_0}$. Es gilt $\sum w_i = 1$. Dann kann der Marktpreis zum Zeitpunkt t geschrieben werden als

$$V_t = V_0 \sum_j w_j S_{t,j} = V_0 \mathbf{w} S_t^T, \quad (2.18)$$

wobei S_t^T der transponierte Vektor von S_t ist. Der Verlust eines Portfolios zum Zeitpunkt T wird dann geschrieben als

$$\begin{aligned} L_T &= V_0 \sum_j w_j (1 - \exp(-R_{(T,j)})) \\ &\approx V_0 \sum_j w_j R_{(T,j)} \\ &= V_0 \mathbf{w} R_{(T)} T, \end{aligned} \quad (2.19)$$

mit $R_{(T)} = (R_{(T,1)}, \dots, R_{(T,d)})$. Dies sind die Renditen der d Finanzwerte im Portfolio zum Zeitpunkt T . Die Renditen eines Portfolios werden bestimmt durch

$$R_t^* = \log V_t - \log V_{t-1}. \quad (2.20)$$

Mittels Formel 2.15 ist zu erkennen, dass ein fast linearer Zusammenhang zwischen den log-Renditen $R_{(T)}$ und dem Verlust L_T besteht. Dies vereinfacht die Schätzung des Risikos.

Im nächsten Abschnitt werden Risikomaße eingeführt. Sie sollen das Risiko über die Verlustverteilung L_t , $t \in \mathbb{N}$ bestimmen.

2.3.0.2 Risikomaße

Ein Risikomaß muss verschiedene Eigenschaften erfüllen, um in seiner Anwendung das Risiko angemessen wiederzugeben. Die wichtigste Eigenschaft für die Risikominimierung ist die Subadditivität. Durch diese wird gewährleistet, dass durch ein Portfolio das Risiko der Summe der einzelnen Positionen vermindert werden kann (siehe Markowitz [60]).

Definition 9 (Risikomaß, Embrechts [24], Artzner et. al. [4]). *Gegeben sei eine Menge V reeller Zufallsvariablen. Eine Funktion $\rho : V \rightarrow \mathbb{R}$ heißt Risikomaß²¹, wenn*

1. *Positivität:* $X \in V, X \geq 0 \Rightarrow \rho(X) \geq 0$,
2. *Subadditivität:* $X, Y, X + Y \in V \Rightarrow \rho(X + Y) \leq \rho(X) + \rho(Y)$,
3. *Positive Homogenität:* $X \in V, h > 0, hX \in V \Rightarrow \rho(hX) = h\rho(X)$, und
4. *Translationsinvarianz:* $X, X + a \in V, a \in \mathbb{R} \Rightarrow \rho(X + a) = \rho(X) + a$.

erfüllt sind.

Die Subadditivität ist eine Grundvoraussetzung für die Risikodiversifikation in einem Portfolio. Wenn ein Risikomaß die Subadditivität erfüllt, ist eine Diversifikation eines Portfolios gleichzusetzen mit der Minimierung des Risikomaßes. Die Subadditivität bildet somit den Ausgangspunkt zur Optimierung der Zusammensetzung eines Portfolios (siehe Embrechts et. al. [24], Abschnitt 3.4).

Nachdem das Risikomaß allgemein definiert wurde, werden nun Risikomaße vorgestellt, die im Finanzsektor angewendet werden. Das Risikomaß wird auf eine Verteilung gewandt, die oben eingeführte Gewinn- und Verlustverteilung.

²¹auch coherentes Risikomaß

2.3.0.3 Value-at-Risk

Der Value-at-Risk (VaR) ist eines der bekanntesten Risikomaße. Er ist nach Definition 9 des Risikomaßes nur für bestimmte Verteilungen ein Risikomaß. Die Subadditivität ist nur für die Familie der elliptischen Verteilungen²² erfüllt. Der Value-at-Risk wird definiert als

$$P\{L_T > VaR(T, q)\} = 1 - q, \quad (2.21)$$

wobei der Value-at-Risk Parameter $VaR(T, q)$ das q -Quantil der Verlust-Verteilung ist. Die Quantilfunktion einer Verteilung mit Verteilungsfunktion F ist definiert als

$$F^{-1}(q) = \inf \{t : F(t) \geq q\}, \quad q \in (0, 1). \quad (2.22)$$

Der Parameter q gibt die Wahrscheinlichkeit²³ an, dass der Verlust kleiner gleich dem $VaR(T, q)$ ist. Die Verteilungsfunktion F_T der T -Tages log-Renditen $R_{(T)} = \sum_{t \leq T} (-R_t)$ für einen einzelnen Finanzwert ist gegeben durch

$$F_T(x) = P\{R_{(T)} \leq x\}. \quad (2.23)$$

Dann kann der Value-at-Risk Parameter geschrieben werden als

$$\begin{aligned} VaR(T, q) &= V_0 (1 - \exp(-F_T^{-1}(1 - q))) \\ &\approx V_0 F_T^{-1}(1 - q), \end{aligned} \quad (2.24)$$

wobei V_0 der Marktpreis zum Zeitpunkt $t = 0$ ist. Es wird der lineare Zusammenhang zwischen Verlustverteilung und Renditen verwendet. Nimmt man diesen linearen Zusammenhang anstatt dem Exponentiellen, so ist der Fehler für einen kleinen Zeitraum vernachlässigbar²⁴. Der VaR eines Portfolios wird über die Verlustverteilung (siehe Formel 2.19) bestimmt, dabei ist $VaR(T, q)$ das q -Quantil dieser Verteilung, d. h. der Value-at-Risk gibt den Verlust an, der nur mit einer bestimmten Wahrscheinlichkeit $1 - q$ überschritten wird. Dabei wird aber keine Aussage darüber gemacht, wie groß der Verlust sein kann, wenn die Schranke überschritten wird. Diesen Mangel versuchen die nächsten Risikomaße zu umgehen. Sie geben den durchschnittlich zu erwartenden Verlust über einer Schranke an.

²²Bei den elliptischen Verteilung besteht der Kontourplot aus Ellipsen, welches ein wesentliches Merkmal dieser Verteilungen darstellt, vgl. [9].

²³z. B. $q = 0.99$ oder $q = 0.95$

²⁴Wenn die Zeitabstände der Renditen klein sind, z. B. Tageskurse. Wird nur von log-Renditen gesprochen, so sind dies log-Renditen berechnet für $T = 1$.

2.3.0.4 Tail Conditional Expectation

Sei L_T die Verlustverteilung, dann ist die Tail Conditional Expectation (TCE) definiert als

$$TCE^{(q)}(L_T) = E(L_T | L_T > VaR(T, q)), \quad (2.25)$$

mit q definiert wie beim VaR . Die TCE gibt den durchschnittlich zu erwartenden Verlust über der Schranke des $VaR(T, q)$ an, sie kann ebenfalls ausgedrückt werden durch²⁵

$$\begin{aligned} E(L_T | L_T > VaR(T, q)) &= \int_{VaR(T, q)}^{\infty} x dF^{[VaR(T, q)]}(x) \\ &= \int_{VaR(T, q)}^{\infty} x \frac{f(x)}{1 - F(VaR(T, q))} dx, \end{aligned} \quad (2.26)$$

wobei f Dichte von F und $F^{[u]}$ die Exzedenten-Verteilungsfunktion mit Schranke u ist. Sie wird definiert durch

Definition 10 (Excess Verteilungsfunktion). Sei F eine Verteilungsfunktion, dann ist die Exzess-Verteilungsfunktion $F^{(u)}$ definiert als

$$F^{(u)}(x) = P(X \leq u + x | X > u) = F^{[u]}(x + u) = \frac{F(x + u) - F(u)}{1 - F(u)}, \quad x \geq 0, \quad (2.27)$$

mit $F^{[u]}$ wird die Exzedenten-Verteilungsfunktion bezeichnet.

2.3.0.5 Expected Shortfall (Reiss und Thomas [73], Acerbi und Tasche [3])

Sei L_T die Zufallsvariable der Verlustverteilung²⁶. Für den festen Zeitraum T sei der Expected Shortfall²⁷ (ES) gegeben durch

$$\begin{aligned} ES^{(q)}(L_T) &= \frac{1}{1 - q} (E[L_T 1_{\{L_T \geq VaR(T, q)\}}] \\ &\quad - VaR(T, q) [P[L_T \geq VaR(T, q)] + q]). \end{aligned} \quad (2.28)$$

Der Expected Shortfall berechnet wie die Tail Conditional Expectation, den zu erwartenden Verlust über einer gegebenen Schranke.

²⁵siehe: Reiss und Thomas [73], Seite 323

²⁶kurz: P+L-Verteilung

²⁷Der Expected Shortfall wird auch als $CVaR^+$ bezeichnet.

2.3.0.6 Empirische Schätzer für die Risikomaße

In diesem Abschnitt sollen Schätzer für die Risikomaße vorgestellt werden. Die Angabe der Schätzer wird durch die Definition der Ordnungsstatistik²⁸ wesentlich vereinfacht. Seien $\{X_i\}$, $i = 1, \dots, n$ Zufallsvariablen auf einem Wahrscheinlichkeitsraum (Ω, \mathcal{A}, P) . Die der Größe nach geordneten X_i werden bezeichnet mit

$$X_{1:n} \leq X_{2:n} \leq \dots \leq X_{n:n}. \quad (2.29)$$

Dies ist die Ordnungsstatistik der X_i , $i = 1, \dots, n$, wobei das Minimum der Ordnungsstatistik

$$X_{1:n} = \min(X_1, \dots, X_n) \quad (2.30)$$

ist, und das Maximum

$$X_{n:n} = \max(X_1, \dots, X_n). \quad (2.31)$$

Dabei wird $X_{r:n}$ als r -te Ordnungsstatistik bezeichnet. Die empirische Verteilungsfunktion der X_i ist definiert durch

$$F_n(t) = \frac{1}{n} \sum_{i=1}^n 1_{(-\infty, t]}(X_i). \quad (2.32)$$

Die empirische Quantilfunktion der X_i ist gegeben durch

$$F_n^{-1}(q) = \begin{cases} X_{nq:n}, & \text{falls } nq \text{ ganzzahlig,} \\ X_{[nq]+1:n}, & \text{sonst,} \end{cases} \quad (2.33)$$

wobei $[nq]$ der ganzzahlige Teil von nq ist. Diese Quantilfunktion kann geschrieben werden als

$$F_n^{-1}(q) = X_{\langle nq \rangle : n} \quad (2.34)$$

mit $\langle nq \rangle = \min\{m : m \geq nq; m \in \mathbb{N}\}$. Seien nun r_1, \dots, r_n Realisation der Renditen eines Finanzwertes R_1, \dots, R_n , wobei die R_i identisch und unabhängig verteilt sind. Mit $z_1 \leq z_2 \leq \dots \leq z_n$ werden die der Größe nach sortierten Realisation der R_i , $i = 1, \dots, n$, bezeichnet. \mathbf{z} ist der Vektor der z_i , $i = 1, \dots, n$.

²⁸siehe Reiss [71]

Bemerkung 1 (Empirischer Schätzer des VaR, Reiss und Thomas [73]).

Der $VaR(T, q)$ ist das $1-q$ -Quantil der Verlustverteilung. Also kann der empirische $VaR(T, q)$ mit dem empirischen Quantil geschätzt werden. Durch Formel 2.24 kann die Verlustverteilung über die Renditen bestimmt werden. Es wird nun ohne Einschränkung angenommen, dass $V_0 = 1$ ist. Der empirische VaR ist definiert als

$$\begin{aligned} \hat{VaR}(T, q) &= V_0 \hat{F}_n^{-1}(q) \\ &= \hat{F}_n^{-1}(q) \\ &= z_{\langle nq \rangle} \end{aligned} \tag{2.35}$$

Der VaR gibt den Verlust als positiven Wert an. Es gibt auch noch andere Methoden um den VaR zu bestimmen (siehe Reiss und Thomas [73], Kapitel 13). In diesem Fall wurde keine Verteilungsannahme gemacht.

Die Bestimmung des empirischen VaR wird auch für die Schätzung der anderen Risikomaße benötigt. Es werden weiter die obigen Annahmen benutzt.

Bemerkung 2 (Empirischer Schätzer für TCE).

Der empirische Schätzer der TCE bildet die Summe aller Elemente, die größer der Schranke $z_{\langle nq \rangle}$ sind. $z_{\langle nq \rangle}$ ist der geschätzte VaR. Die Summe wird durch die Anzahl der Elemente über der Schranke geteilt.

$$\begin{aligned} T\hat{C}E_n(q) &= \frac{\sum_{i=1}^n z_i \mathbf{1}_{\{z_i \geq z_{\langle nq \rangle}\}}}{\sum_{i=1}^n \mathbf{1}_{\{z_i \geq z_{\langle nq \rangle}\}}} \\ &= (\text{Durchschnitt aller } z_i \geq z_{\langle nq \rangle}). \end{aligned} \tag{2.36}$$

Bemerkung 3 (Empirischer Schätzer für ES).

Der empirische Schätzer des ES teilt die Summe, nicht wie der TCE durch die Anzahl der Elemente, sondern benutzt statt dessen den Wert $n - \langle nq \rangle$.

$$\hat{E}S_n(q) = \frac{\sum_{i=1}^n z_i \mathbf{1}_{\{z_i \geq z_{\langle nq \rangle}\}}}{n - \langle nq \rangle} - z_{\langle nq \rangle} \tag{2.37}$$

Die Subadditivität des empirischen Schätzers kann leicht gezeigt werden. Seien $\mathbf{x} = \{x_i\}$, $i = 1, \dots, n$, und $\mathbf{y} = \{y_i\}$, $i = 1, \dots, n$, Realisationen. Mit z_i und w_i

werden die der Größe nach geordneten Realisationen bezeichnet ($\mathbf{z} = \{z_i\}$, $\mathbf{w} = \{w_i\}$, $i = 1, \dots, n$).

$$\begin{aligned}\hat{ES}_{z+w}(q) &= \frac{\sum_{i=1}^n (z_i + w_i) \mathbf{1}_{\{z_i + w_i \geq z_{\langle nq \rangle} + w_{\langle nq \rangle}\}}}{n - \langle nq \rangle} - (z_{\langle nq \rangle} + w_{\langle nq \rangle}) \\ &\leq \frac{\sum_{i=1}^n ((z_i) \mathbf{1}_{\{z_i \geq z_{\langle nq \rangle}\}} - z_{\langle nq \rangle}) + (w_i) \mathbf{1}_{\{w_i \geq w_{\langle nq \rangle}\}} - w_{\langle nq \rangle})}{n - \langle nq \rangle} \\ &= \hat{ES}_z(q) + \hat{ES}_w(q)\end{aligned}$$

Der empirische Schätzer des TCE ist nicht subadditiv (Acerbi und Tasche [2]).

Es kann gezeigt werden, dass (Acerbi und Tasche [2]) wenn X eine stetige Verteilung zu Grunde liegt, dass gilt

$$ES(q) = TCE(q), \quad q \in (0, 1). \quad (2.38)$$

2.4 Studentsche Modellierung von Finanzdaten

2.4.1 Univariate Modellierung

Im letzten Abschnitt wurden verschiedene Risikomaße vorgestellt. Damit aber ein Risikomaß angewandt werden kann, muss die Gewinn- und Verlustverteilung²⁹ bekannt sein. Ohne diese Verteilung kann das Risiko nicht berechnet werden.

Diese Verteilung muss aus den zur Verfügung stehenden Daten geschätzt werden, also aus den vergangenheitsbezogenen Aktienkursen oder Renditen von Finanzwerten. Es gibt zwei Möglichkeiten aus diesen historischen Daten die Verteilung zu bestimmen. Zum einem über die empirische Verteilung, zum anderen kann die Verteilung über ein Modell bestimmen werden. Hier soll der Schwerpunkt auf den zweiten Ansatz, der Modellbildung, gelegt werden.

In diesem Modell sollten möglichst viele Stylized Facts (siehe Tabelle 2.1) erfüllt sein, da diese die Eigenschaften der Finanzwerte beschreiben. Es wird davon ausgegangen, dass die Daten aus den relativen Kursänderungen³⁰ bestehen und univariat³¹ sind. Da die Daten nahezu unkorreliert³² sind, ist die Bestimmung der zu Grunde liegenden Verteilung relativ einfach³³. Es wird also angenommen, dass das Stylized Fact 2 erfüllt ist. Anwendung findet meist das Normalverteilungsmodell³⁴, welches aber, wie festgestellt wurde, das Risiko in den Flanken unterschätzt³⁵. Basierend auf dieser Einschränkung soll ein Modell mit der Studentschen Verteilung verwendet werden. Die Studentsche Verteilung³⁶ hat stärkere Flanken³⁷ und modelliert deshalb besser die Verteilung der Finanzwerte. Außerdem wird angenommen, dass zwischen dieser Verteilung und der Gewinn- und Verlustverteilung ein linearer Zusammenhang besteht. Dadurch kann die Berechnung des Risikos aus der Verteilung der Renditen erfolgen. Bei der Bestimmung dieser Verteilung gehen die Zeitinformationen verloren, deshalb wird der Stylized Fact 3 nicht erfüllt. Soll dieser Stylized Fact auch erfüllt werden, muss in dem Modell die Zeitinformation erhalten bleiben, d. h. eine Zeitreihe muss dem Modell

²⁹engl. Profit+Loss-Verteilung, oder nur Verlustverteilung

³⁰Renditen oder log-Renditen

³¹Es wird zunächst nur eine Aktie betrachtet.

³²Dies kann durch die empirische Berechnung der Autokorrelationsfunktion gezeigt werden.

³³Es müssen z. B. keine Modelle mit bedingten Verteilungen benutzt werden.

³⁴Es erfüllt Stylized Fact 5,6.

³⁵siehe 2.2.1, Seite 52

³⁶Sie erfüllt Stylized Fact 4,5,6.

³⁷engl. fat tails

zu Grunde liegen. Diese Zeitreihe sollte dann ebenfalls Volatilitätscluster haben. Dies führt später zum Modell der Studentschen Zeitreihe. Danach werden diese univariaten Modelle auf multivariate Modelle erweitert. Bei den multivariaten Modellen muss dann der Stylized Fact 7 berücksichtigt werden.

2.4.2 Univariater Fall

In diesem Abschnitt wird die univariate Studentsche Verteilung vorgestellt. Sie soll die Grundlage für die Analyse der Finanzdaten bilden. Zunächst werden univariate Modelle betrachtet, die später zu multivariaten Modelle erweitert werden.

Definition 11 (Univariate Dichte der Studentschen Verteilung). *Die Dichte der Studentschen Verteilung mit Gestaltsparemeter $\alpha > 0$, Lokationsparameter $\mu \in \mathbb{R}$ und Skalenparameter $\sigma > 0$ ist gegeben durch*

$$f_{\alpha,\mu,\sigma}^S(x) = \frac{\Gamma(\frac{\alpha+1}{2})}{\Gamma(\frac{\alpha}{2})\Gamma(\frac{1}{2})\sigma} \left(1 + \frac{(x - \mu)^2}{\alpha\sigma^2}\right)^{-(\alpha+1)/2}. \quad (2.39)$$

Die Studentsche Verteilung kann als bedingte³⁸ Normalverteilung bzgl. einem gamma-verteilten Skalenparameter angesehen werden. Die Gamma-Dichte ist definiert durch

Definition 12 (Dichte der Gamma-Verteilung). *Die Dichte der Gamma-Verteilung ist definiert als*

$$f_{r,s}^G(x) = \frac{s}{\Gamma(r)} (sx)^{r-1} \exp(-sx) \quad (2.40)$$

mit Gestaltsparemeter r und reziprokem Skalenparameter $s = 1/\sigma$. Die standard Gamma-Dichte hat folgende Form:

$$f_r^G(x) = \frac{1}{\Gamma(r)} x^{r-1} \exp(-x). \quad (2.41)$$

Mit $\Gamma(r)$ wird die Gamma-Funktion bezeichnet³⁹. Der Zusammenhang über die Mischung wird im nachfolgenden Abschnitt vorgestellt.

³⁸gemischte Normalverteilung

³⁹siehe Anhang 41, S. 179

Satz 1 (Dichte der Studentische Verteilung als Mischung). Die Zufallsvariable

$$Z = \frac{X}{\sqrt{2Y/\alpha}} = \frac{X}{\sqrt{Y/r}} \quad (2.42)$$

mit X standard-normalverteilt und Y gamma-verteilt mit Gestaltsparameter $r = \alpha/2$ und Skalenparameter $s = 1$ ist Studentisch-verteilt. Die Dichte der Studentischen Verteilung kann als Mischung einer Normalverteilung mit einer reziproken Gamma-Verteilung angesehen werden, d. h. die reziproke Varianz $1/\sigma$ der Normalverteilung $N_{0,1/\vartheta}$ ist gammaverteilt.

$$F_S(x) = \int \Phi_{0,1/\gamma}(x)g(\gamma)d\gamma, \quad (2.43)$$

wobei g die Dichte aus Lemma 1 ist. Dies ist die Verteilungsfunktion der Studentischen Verteilung und besitzt die Dichte f_α^S .

Beweis. Für den Beweis werden folgende Hilfssätze benötigt:

Lemma 1. Wenn Y eine Gamma-Dichte mit Gestaltsparameter $r > 0$ und Skalenparameter $s = 1$ besitzt, dann hat $Z = (\frac{Y}{r})^{1/2}$ die Dichte

$$f_Z(x) = 2rx f_{r,1}^G(rx^2).$$

Beweis des Lemma. Mit dem Transformationssatz für Dichten folgt:

Sei $T(y) = (\frac{y}{r})^{1/2}$ eine Abbildung, für diese gilt dann:

1. $T'(y) = \frac{1}{2r} \frac{1}{(y/r)^{1/2}}$,
2. $T^{-1}(x) = rx^2$,
3. $|T'(y)| \neq 0$ für alle $y > 0$.

Damit folgt

$$\begin{aligned} f_Z(x) &= \frac{1}{|T'(T^{-1}(x))|} f_{r,1}^G(T^{-1}(x)) \\ &= \frac{1}{\left| \frac{1}{2r} \left(\frac{1}{\sqrt{(rx^2)/r}} \right) \right|} f_{r,1}^G(rx^2) \\ &= 2rx f_{r,1}^G(rx^2). \end{aligned}$$

□

Außerdem wird noch folgender Hilfssatz benötigt:

Lemma 2 (Dichte von X/Y). Seien X und $Y > 0$ unabhängige Zufallsvariablen mit Dichten f und h . Dann hat $\frac{X}{Y}$ die Dichte

$$x \rightarrow \int \vartheta f(\vartheta x) h(\vartheta) d\vartheta. \quad (2.44)$$

Das heißt $\mathcal{L}\left(\frac{X}{Y}\right)$ ist die Mischung der Wahrscheinlichkeitsmaße mit Dichten $x \rightarrow \vartheta f(x\vartheta)$ bzgl. $\mathcal{L}(Y)$.

Beweis.

$$\begin{aligned} P\left\{\frac{X}{Y} \leq z\right\} &= P\{g(X, Y) \leq z\}, \text{ setze } g(x, y) = \frac{x}{y} \\ &= \mathcal{L}(X, Y)\{(x, y) : g(x, y) \leq z\} \\ &= \int 1_{(-\infty, z]} \left(\frac{x}{y}\right) f(x) h(y) dx dy \\ &= \int \left(\int 1_{(-\infty, z]}(x) y f(xy) dx \right) h(y) dy \\ &= \int_{-\infty}^z \left(\int y f(xy) h(y) dy \right) dx. \end{aligned} \quad (2.45)$$

□

Somit folgt mit Lemma 1 für die Mischung

$$\begin{aligned} f(x) &= \int_0^\infty \vartheta \varphi_{0,1}(\vartheta x) 2r\vartheta f_r^G(r\vartheta^2) d\vartheta \\ &= \int_0^\infty 2r\vartheta^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\vartheta^2 x^2} \frac{1}{\Gamma(r)} (r\vartheta^2)^{r-1} e^{-r\vartheta^2} d\vartheta \\ &= \frac{2}{\sqrt{2\pi}\Gamma(r)} \int_0^\infty e^{-\frac{1}{2}\vartheta^2 x^2} e^{-r\vartheta^2} r\vartheta^2 (r\vartheta^2)^{r-1} d\vartheta \\ &= \frac{2}{\sqrt{2\pi}\Gamma(r)} \int_0^\infty e^{-r\vartheta^2 \left(1 + \frac{x^2}{2r}\right)} (r\vartheta^2)^r d\vartheta. \end{aligned}$$

Durch Substitution von $\frac{dr\vartheta^2}{d\vartheta} = 2r\vartheta \Leftrightarrow d\vartheta = \frac{dr\vartheta^2}{2r\vartheta}$ mit den Grenzen $2r0 = 0$ und $2r\infty = \infty$ ⁴⁰, folgt:

$$\begin{aligned}
 f^{Misch}(x) &= \dots \\
 &= \frac{2}{\sqrt{2\pi}\Gamma(r)} \int_0^\infty e^{-r\vartheta^2\left(1+\frac{x^2}{2r}\right)} \frac{1}{2r\vartheta} (r\vartheta^2)^r dr\vartheta^2 \\
 &= \frac{1}{\sqrt{2\pi}\Gamma(r)} \int_0^\infty e^{-r\vartheta^2 a} \frac{1}{\sqrt{r}} (r\vartheta^2)^{r-\frac{1}{2}} dr\vartheta^2 \text{ mit } a = \left(1 + \frac{x^2}{2r}\right) \\
 &= \frac{1}{\sqrt{2\pi r}\Gamma(r)} \int_0^\infty e^{-ua} u^{r-\frac{1}{2}} du \text{ mit } u = r\vartheta^2 \\
 &= \frac{1}{\sqrt{2\pi r}\Gamma(r)} \frac{\Gamma\left((r - 1/2) + 1\right)}{a^{r-(1/2)+1}}, \\
 &\text{wegen } \int_0^\infty x^n e^{-ax} dx = \frac{\Gamma(n+1)}{a^{n+1}} \text{ f\"ur } a > 0, n > -1 \\
 &= \frac{\Gamma(r+1/2)}{\sqrt{2\pi r}\Gamma(r)} a^{-(r+1/2)} \\
 &= \frac{\Gamma(r+1/2)}{\sqrt{2\pi r}\Gamma(r)} \left(1 + \frac{x^2}{2r}\right)^{-(r+1/2)} \\
 &= \frac{\Gamma\left(\frac{\alpha+1}{2}\right)}{\Gamma(\alpha/2)\sqrt{\pi\alpha}} \left(1 + \frac{x^2}{\alpha}\right)^{-((\alpha+1)/2)} \\
 &= f_\alpha^S(x)
 \end{aligned}$$

□

2.4.2.1 Eigenschaften

Satz 2 (MDA der Studentischen Verteilung). Die Studentische Verteilung mit $\alpha > 0$ befindet sich im Maximum-Anziehungsbereich der Fréchet-Verteilung.

Beweis. Siehe Anhang B.1.1.1, S. 181. □

Lemma 3 (spezielle Eigenschaften). Sei X eine Student-verteilte Zufallsvariable. Es gelten dann folgende Eigenschaften ($\mathcal{L}(X) = t_{\alpha,\mu,\sigma}$):

- $E(X) = \mu$,

⁴⁰d. h. die Grenzen bleiben erhalten.

- $Var(X) = \frac{\alpha}{\alpha-2}\sigma^2$ für $\alpha > 2$,
- $\gamma(X) = 0$, d. h. X ist symmetrisch,
- $\delta(X) = 3 + \frac{6}{\alpha-4}$.

2.4.2.2 Schätzer

Für die univariate Studentsche Verteilung gibt es unterschiedliche Schätzer. Zunächst wird der Maximum-Likelihood-Schätzer⁴¹ betrachtet, danach wird ein weiterer Schätzer für α , ein Momentenschätzer, angegeben. Da die Finanzdaten einen Erwartungswert von $\mu = 0$ haben, wird für die Schätzung die zentrale Studentsche Verteilung benutzt.

Lemma 4 (ML-Schätzer der Studentschen Verteilung). *Im Modell $\{t_{\alpha,0,\sigma^2}^n : \alpha > 0, \sigma > 0\}$ ist der Maximum-Likelihood-Schätzer $(\hat{\alpha}, \hat{\sigma})$ für (α, σ) eine Lösung der beiden folgenden Gleichungen*

1.

$$\begin{aligned} & \frac{1}{2}n \ln(\alpha) + \frac{1}{2}n + n \ln(\sigma) + \frac{n}{2\alpha} + \frac{1}{2}n\Psi\left(\frac{\alpha+1}{2}\right) - \frac{1}{2}n\Psi\left(\frac{1}{2}\alpha\right) \\ & + \sum_{i=1}^n \left(-\frac{1}{2} \ln(\alpha\sigma^2 + x_i^2) - \frac{1}{2} \frac{\alpha\sigma^2}{\alpha\sigma^2 + x_i^2} - \frac{1}{2} \frac{\sigma^2}{\alpha\sigma^2 + x_i^2} \right) \\ & = 0 \end{aligned} \tag{2.46}$$

2.

$$\frac{n\alpha}{s} + \sum_{i=1}^n \left(-\frac{\alpha^2\sigma}{\alpha\sigma^2 + x_i^2} - \frac{\alpha\sigma}{\alpha\sigma^2 + x_i^2} \right) = 0 \tag{2.47}$$

Mit Ψ wird die Digamma-Funktion bezeichnet.

⁴¹Der Maximum-Likelihood-Schätzer wurde Johnson und Kotz [49] entnommen.

Beweis. Die beiden Gleichungen werden durch Ableiten nach dem entsprechenden Parameter der Log-Likelihood-Funktion und Null-Setzen gewonnen:

$$\begin{aligned} \log L(\alpha|x_1, \dots, x_n) &= \frac{1}{2}n\alpha \ln(\alpha) + n\alpha \ln(\sigma) + \frac{1}{2}n \ln(\alpha) - \frac{1}{2}n \ln(\pi) \\ &\quad + n \ln \left(\Gamma \left(\frac{\alpha + 1}{2} \right) \right) - n \ln \left(\Gamma \left(\frac{1}{2}\alpha \right) \right) \\ &\quad + \sum_{i=1}^n \left(-\frac{1}{2}\alpha \ln(\alpha\sigma^2 + x_i^2) - \frac{1}{2} \ln(\alpha\sigma^2 + x_i^2) \right). \end{aligned} \quad (2.48)$$

□

Der Schätzer kann nicht in geschlossener Form angegeben werden, sondern muss mit einem numerischen Verfahren, z. B. dem Newton-Verfahren, bestimmt werden. Der Momentenschätzer ist im folgenden Lemma dargestellt.

Lemma 5 (Momentenschätzer für den Gestaltparameter α). *Der Momentenschätzer ist für $\hat{\delta} > 3$ gegeben durch*

$$\hat{\alpha} = \frac{6}{\hat{\delta} - 3} + 4 \quad (2.49)$$

wobei $\hat{\delta}$ die empirische Kurtosis ist.

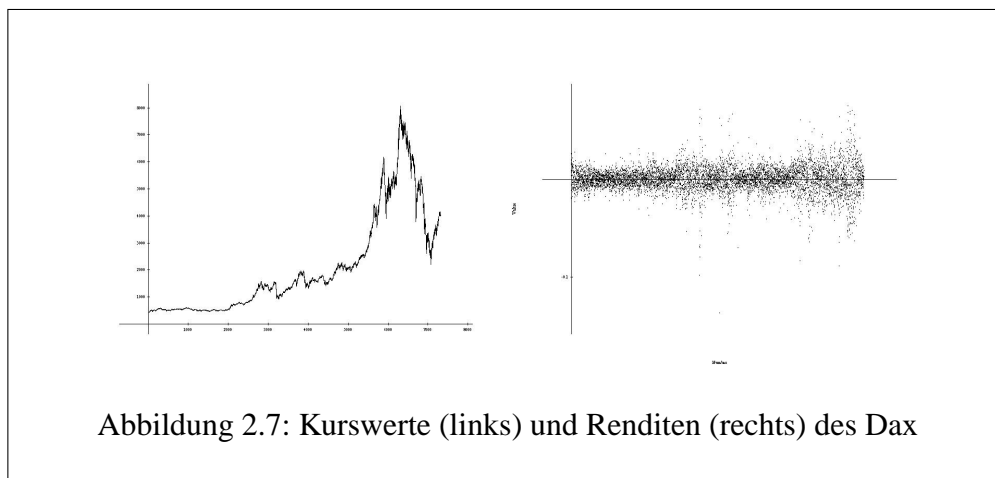
Beweis. Der Schätzer ergibt sich aus Lemma 3 und der empirischen Berechnung der Kurtosis. □

2.4.2.3 Datenanalyse

In diesem Abschnitt wird das univariate Studentsche Modell mit der Normalverteilungsannahme verglichen. Für die Modellierung soll die Studentsche Verteilung benutzt werden, sie löst die sonst übliche Normalverteilung ab. Die Modelle werden anhand von Dax-Daten betrachtet. Zu Grunde liegen die Index-Werte des Dax in einem Zeitraum vom 03.01.1975 bis zum 01.03.2004⁴². Diese absoluten Werte werden zunächst in Renditen⁴³ transformiert, bevor mit geeigneten Schätzern die Parameter der beiden Modelle geschätzt werden. Es werden zunächst jedoch die empirischen Daten näher betrachtet. In Abbildung 2.7 wurden

⁴²Stichprobenumfang des Datensatzes: $n = 10651$.

⁴³relative Kursänderungen



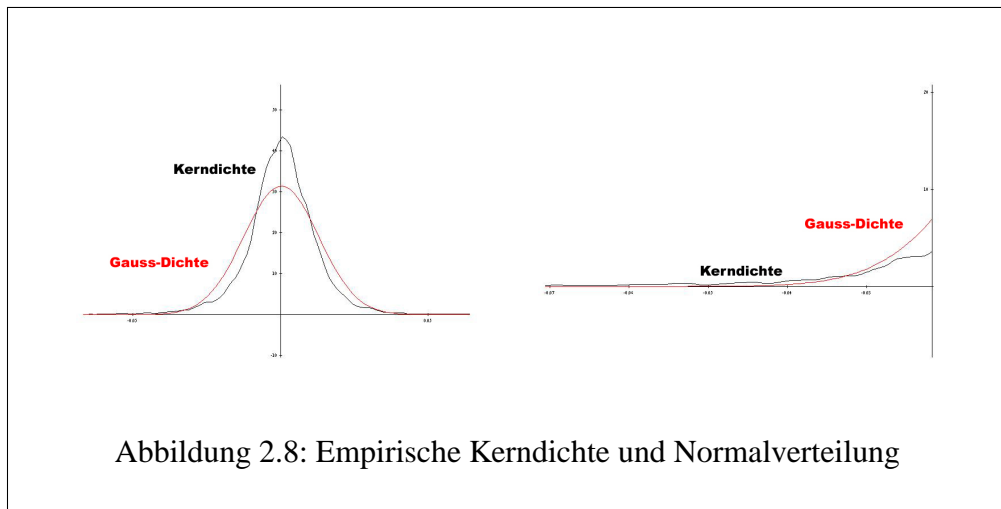
Parameter	Schätzwert
Normalverteilung μ :	0.000312848
Normalverteilung σ :	0.0126934
Studentsche Verteilung α :	3.47764
Studentsche Verteilung σ :	0.00850071

Tabelle 2.2: Geschätzte Parameter für die beiden Modelle

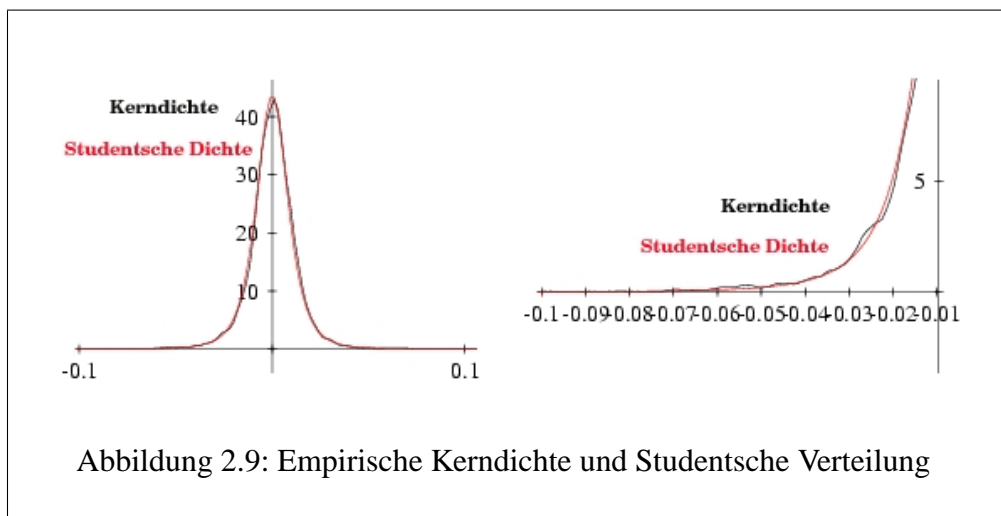
die Kurswerte und Renditen in Abhängigkeit zur Zeit geplottet. Betrachtet man die Autokorrelationsfunktionen (siehe Abbildung 2.3 , S. 50), so sind die Renditen nahezu unkorreliert, die Kurse dagegen sind korreliert. Dies wurde in den Stylized Fact 1 und Stylized Fact 2 formuliert (siehe Tabelle 2.1, Seite 56).

Ausgehend von den Renditen werden mit dem Maximum–Likelihood–Schätzer für die Normalverteilung und die Studentischen Verteilung die Parameter der Modelle geschätzt (siehe Tabelle 2.2).

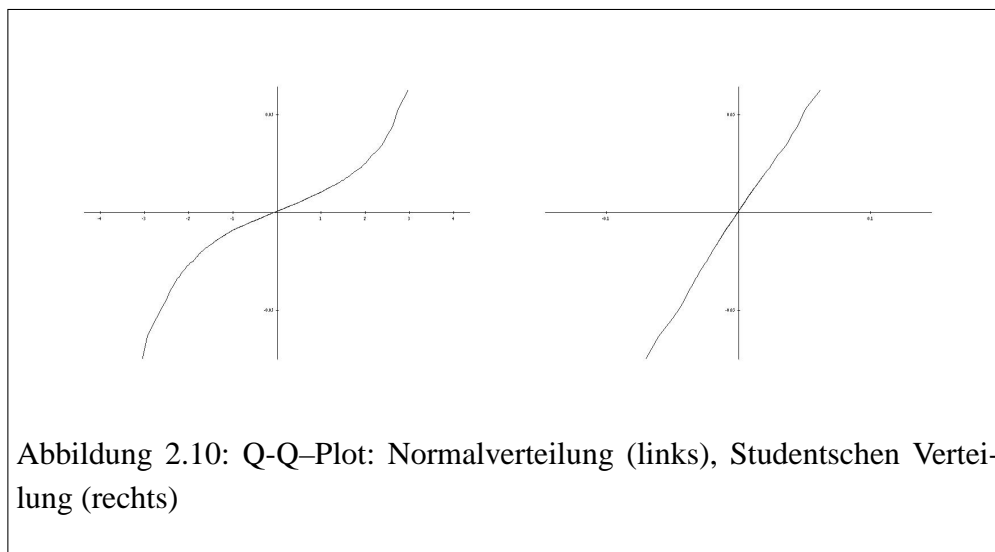
In der nächsten Abbildung wird die empirische Kerndichte der Dax–Renditen mit den Dichten der Modelle verglichen, dabei werden in die Modelle die geschätzten Parameter eingesetzt. Es kann festgestellt werden, dass die Normalverteilungsannahme für die Dax–Renditen nicht zutreffend ist. Die Normalverteilung hat schwächere Flanken (siehe Abbildung 2.8 (rechts)) und eine gestauchte Mitte (siehe Abbildung 2.8 (links)) gegenüber der empirischen Dichte. Das Normalvertei-



lungsmodell erfüllt Stylized Fact 4 nicht, wie hier auch empirisch festzustellen ist. Das Studentsche Modell hingegen passt sich besser an die empirische Dichte an, auch die Flanken werden besser erfasst (siehe Abbildung 2.9). Die Aussage, dass



die Studentsche Verteilung besser der empirischen Verteilung als die Normalverteilung entspricht, wird auch durch deren Q-Q-Plots bestätigt (siehe Abbildung 2.10). Den Unterschied zwischen der empirischen Dichte, der Normalverteilungsdichte und der Studentsche Dichte wird in Abbildung 2.11 veranschaulicht.



2.4.3 Studentsche Zeitreihe

2.4.3.1 Modellbildung der Zeitreihe

Das Modell der Studentschen Verteilung, welches im letzten Abschnitt vorgestellt wurde, erfüllt mehr Stylized Facts der Finanzdaten als das Normalverteilungsmodell. Es ist dabei besser zur Risikomessung der Finanzdaten geeignet. Jedoch erfüllen beide Modelle nicht Stylized Fact 3, welcher besagt, dass bei Finanzdaten Volatilitätscluster auftreten. Die Volatilitätscluster sind zeitabhängig und können nur durch ein Modell berücksichtigt werden, welches diese Zeitabhängigkeit wiedergibt. Deshalb wird nun ein Modell einer Studentschen Zeitreihe vorgeschlagen, welches in diesem Abschnitt vorgestellt wird.

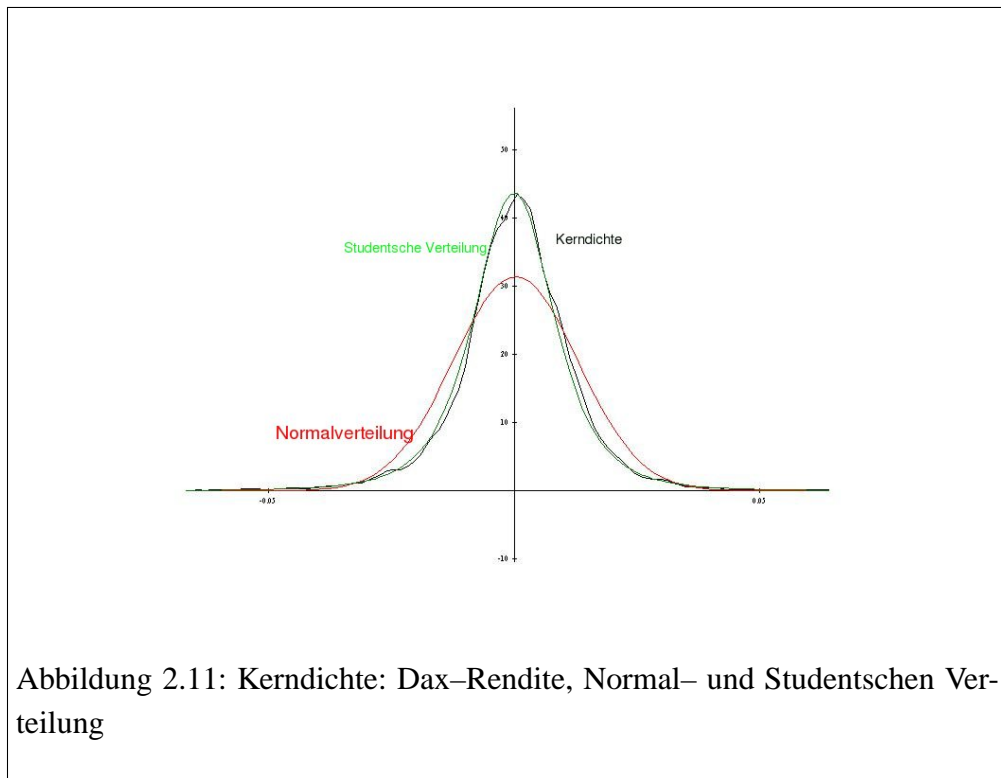
Das Modell der Studentschen Zeitreihe soll die log-Renditen eines Finanzwertes simulieren. Über Formel (2.3) und einem bekannten Kurswert S_0 ⁴⁴ können rekursiv die Kurse der Finanzwerte berechnet werden. Es gilt folgender Zusammenhang:

$$S_t = S_{t-1} \exp(R_t), t = 1, \dots$$

Die log-Renditen im Studentschen Zeitreihenmodell sollen die Form

$$R_t = \sigma_t \epsilon_t \tag{2.50}$$

⁴⁴Es wird angenommen, dass der Kurswert für den heutigen Tag bekannt ist.



haben. Die σ_t stellen eine stochastische Volatilitätsserie dar, die ϵ_t sind Zufallsvariablen zum Zeitpunkt t . Im Folgenden wird die stochastische Volatilitätsserie näher betrachtet. In Abschnitt 2.4.1 wurde gezeigt, dass die Studentische Verteilung als eine Mischung aus einer standardnormal-verteilten Zufallsvariable X und einer gamma-verteilten Zufallsvariable Y mit Parameter r dargestellt werden kann.

$$\frac{X_t}{\sqrt{Y_t/r}}$$

Dieser Ansatz wird auch in diesem Zeitreihenmodell verwendet. Die ϵ_t sollen in dem Zeitreihenmodell unabhängig identisch und standard-normalverteilt sein, wie X_t , demnach müssen die σ_t gamma-verteilt mit Parameter r , wie Y_t sein. Für gamma-verteilte Zufallsvariablen gilt, dass auch deren Summe wieder gamma-verteilt ist.

Bemerkung 4 (Summe von gamma-verteilten Zufallsvariablen). Sei Y_1, \dots, Y_n unabhängige gamma-verteilte Zufallsvariablen mit Parameter $r_i, i = 1, \dots, n$. Dann ist

$$W = Y_1 + \dots + Y_n$$

gamma-verteilt mit Parameter $r = r_1 + \dots + r_n$.

Es seien nun die ϵ_t eine Reihe von unabhängig identisch standard-normalverteilten Zufallsvariablen, die unabhängig von der gamma-verteilten Zufallsvariable σ_t sind für alle t . Die σ_t sollen folgenden Aufbau haben:

$$\sigma_t = \sigma \sqrt{\frac{r}{Y_t}}, \sigma > 0, r > 0, \quad (2.51)$$

wobei σ ein fester Skalenparameter ist und die Y_t standard gamma-verteilte Zufallsvariablen mit Parameter r sein sollen. Diese gamma-verteilte Zufallsvariable Y_t soll als Summe von gamma-verteilten Zufallsvariablen $\eta_{t,m}$ dargestellt werden. Wobei $\mathcal{L}(\eta_{t,m}) = \Gamma(r_m, 1)$ für alle t und m eine Reihe von identisch unabhängig gamma-verteilten Zufallsvariablen $\eta_{t,m}, t \in \mathbb{Z}, m = 1, \dots, q$ sei. Es gelte $r = \sum_{i=0}^{q-1} \sum_{m=i+1}^q r_m$ für $q \in \mathbb{N}$ und die Darstellung

$$Y_t = \sum_{i=0}^{q-1} \sum_{m=i+1}^q \eta_{t-i,m}. \quad (2.52)$$

q beschreibt den Zeithorizont,⁴⁵ über den die Summe der gamma-verteilten Zufallsvariablen gebildet wird. Falls der Parameter $r_m = 0$ ist, soll die entsprechende Zufallsvariable gleich null gesetzt werden.

Es wird vorausgesetzt, dass die Parameter r_m über eine Wahrscheinlichkeitsdichte k , welche auf dem Intervall $[-q, 0]$ streng monoton wachsend sein soll, berechnet werden können:

$$r_m = r s_m, \quad (2.53)$$

wobei s_m die Form

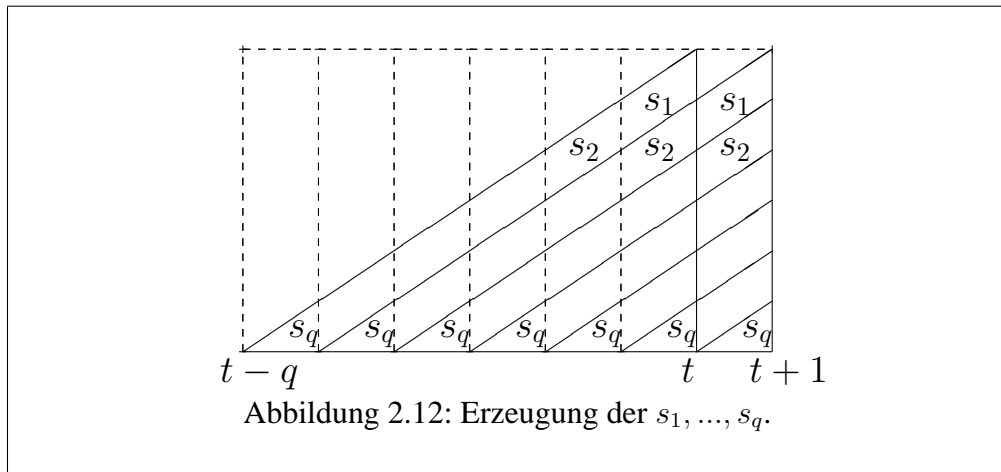
$$s_m = \int_{-1}^0 (k(x - (m - 1)) - k(x - m)) dx, \quad m = 1, \dots, q \quad (2.54)$$

⁴⁵ q drückt den Zeitraum aus, wie weit zur Berechnung Daten aus der Vergangenheit mit einbezogen werden.

besitzen. Für die Summe der s_m gilt

$$\sum_{i=0}^{q-1} \sum_{m=i+1}^q s_m = 1$$

Somit ist die Zufallsvariable Y_t gamma-verteilt mit Parameter r . Da die Y_t wegen der Summendarstellung zeitabhängig sind, wird in Abbildung 2.12 anschaulich dargestellt, wie der Zusammenhang zwischen der Zufallsvariable Y_t und Y_{t+1} über die s_m ist. Der Abbildung kann entnommen werden, welche $\eta_{t,m}$ aus der Summe



Y_t bei Übergang zur Zufallsvariable Y_{t+1} entfernt werden, und welche dafür neu in die Summe eingefügt werden. Die Berechnung der s_m kann ebenfalls durch die Verteilungsfunktion K der Dichte k dargestellt werden. Es gilt

$$s_m = (K(-(m-1)) - 2K(-m) + K(-(m+1))), \quad m = 1, \dots, q, \quad (2.55)$$

wegen

$$\begin{aligned} s_m &= \int_{-1}^0 (k(x - (m-1)) - k(x - m)) dx \\ &= K(x - (m-1)) - K(x - m) \Big|_{-1}^0 \\ &= K(-(m-1)) - K(-m) - K(-1 - (m-1)) + K(-1 - m) \\ &= K(-(m-1)) - 2K(-m) + K(-(m+1)). \end{aligned}$$

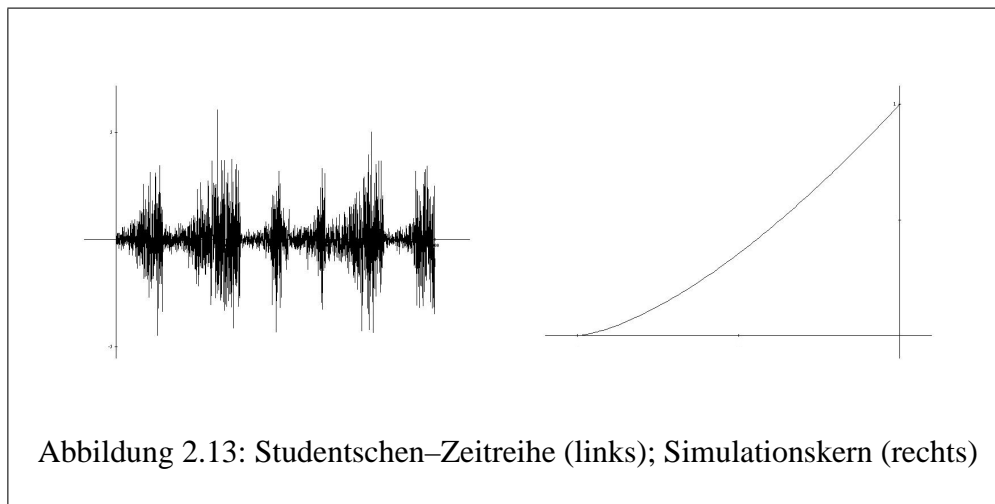
K muss wegen den Anforderungen, die an k^{46} gestellt wurden, streng konvex auf $[-q, 0]$ sein. K ist eine Verteilungsfunktion und kann als Kern auf $[-q, 0]$ betrachtet werden. Ein spezieller Kern für die Simulationen wäre der Kern

$$K(x) = B(x/q) \quad (2.56)$$

mit

$$B(x) = (1+x)^{(1+\beta)}, \quad -1 \leq x \leq 0, \beta > 0. \quad (2.57)$$

Für $q = 300$ und $\beta = 0.5$ sieht man in der Abbildung 2.13 (links), dass Volatilitätscluster auftreten. Die Form des Kerns kann aus Abbildung 2.13 (rechts) entnommen werden. Die Zufallsvariable Y_t kann in Vektorschreibweise ausgedrückt



werden. Es gilt

$$Y_t = \sum_{j=0}^{q-1} \psi_j \eta'_{t-j}, \quad (2.58)$$

mit

$$\eta_t = (\eta_{t,1}, \dots, \eta_{t,q}),$$

Vektoren mit gamma-verteilten Zufallsvariablen und

$$\psi_j = (\psi_{j,1}, \dots, \psi_{j,q})$$

⁴⁶ k sollte streng monoton wachsend sein.

Zählvektoren mit der Darstellung

$$\psi_0 = (1, \dots, 1), \psi_1 = (0, 1, \dots, 1), \dots, \psi_{q-1} = (0, \dots, 0, 1).$$

Mit η_{t-j}' wird der transponierte Vektor von η_{t-j} bezeichnet. Durch den Aufbau der Zählvektoren kann die Summe 2.58 als unendliche Reihe

$$Y_t = \sum_{j=0}^{\infty} \psi_j \eta_{t-j}', \quad (2.59)$$

mit $\eta_t = (\eta_{t,m})_{m=1}^{\infty}$ und $\psi_j = (\psi_{j,m})_{m=1}^{\infty}$ ⁴⁷ für $j = 0, 1, 2, \dots$, ausgedrückt werden. Die η_t sollen unabhängig und identisch verteilt sein. Wenn $\eta_{t,m}$ den Parameter r_m besitzt, dann muss

$$\sum_{j=0}^{\infty} \left(\sum_{m=1}^{\infty} \psi_{j,m} r_m \right) < \infty \quad (2.60)$$

gelten, und die Y_t sind gamma-verteilt mit Parameter

$$r = \sum_{m=1}^{\infty} \left(\sum_{j=0}^{\infty} \psi_{j,m} \right) r_m. \quad (2.61)$$

Der Zählvektor soll folgenden Aufbau haben

$$\psi_{j,m} = \begin{cases} 0 & m \leq j \\ 1 & m > j \end{cases}. \quad (2.62)$$

Es gilt dann

$$\sum_{j=0}^{\infty} \psi_{j,m} = m$$

und

$$r = \sum_{m=0}^{\infty} m r_m, \quad r_m = 0 \text{ für } m > q. \quad (2.63)$$

⁴⁷ $\psi_{j,m} = 1$ oder $\psi_{j,m} = 0$

Lemma 6 (Autokovarianzfunktion der Y_t). Die Zeitreihe Y_t ist streng stationär mit Autokovarianzfunktion

$$\text{Cov}(Y_0, Y_h) = rK(-h), \quad h = 0, 1, \dots \quad (2.64)$$

Beweis.

$$\begin{aligned} \text{Cov}(Y_0, Y_h) &= E((Y_0 - E(Y_0))(Y_h - E(Y_h))) \\ &= E(Y_0 Y_h - E(Y_0) Y_h - Y_0 E(Y_h) + E(Y_0) E(Y_h)) \\ &= E(Y_0 Y_h) - E(Y_0) E(Y_h) \\ &= E(Y_0 Y_h) - E(Y_0)^2 \end{aligned} \quad (2.65)$$

Der Erwartungswert von Y_t ist r für alle $t \in \mathbb{N}$, da Y_t standard-gamma-verteilt mit Parameter r ist. Für den Erwartungswert von $Y_0 Y_h$ ist die Berechnung etwas komplexer. Jede der Zufallsvariablen Y_0 und Y_h wird in eine Summe aufgespalten. Es soll gelten

$$Y_0 = X_{0,h} + X_{1,h} \quad \text{und} \quad Y_h = X_{0,h} + X_{2,h}, \quad (2.66)$$

mit folgender Eigenschaft für $X_{0,h}$:

$$X_{0,h} = \sum_{j=h}^{\infty} \psi_j \eta'_{h-j} = \sum_{j=0}^{\infty} \psi_{j+h} \eta'_{-j}. \quad (2.67)$$

Durch Subtraktion von 2.66 mit 2.67, folgt für $X_{1,h}$ und $X_{2,h}$

$$\begin{aligned} X_{1,h} &= \sum_{j=0}^{\infty} (\psi_j - \psi_{j+h}) \eta'_{-j}, \\ X_{2,h} &= \sum_{j=0}^{h-1} \psi_j \eta'_{h-j}. \end{aligned}$$

$X_{0,h}$, $X_{1,h}$ und $X_{2,h}$ sind unabhängige Zufallsvariablen. Die Zufallsvariablen Y_0 und Y_h wurden so aufgespalten, dass in beiden die Zufallsvariable $X_{0,h}$ enthalten ist. Sie repräsentiert den abhängigen Teil von Y_0 und Y_h , wobei die beiden Zufallsvariablen $X_{1,h}$ und $X_{2,h}$ den unabhängigen Teil von Y_0 und Y_h darstellen. Dieser ist jeweils nur in einer Zufallsvariable enthalten. $X_{0,h}$ sei standard-

gamma-verteilt mit Parameter $rK(-h)$, $X_{1,h}$ und $X_{2,h}$ haben den Parameter $r(1 - K(-h))$. Der Erwartungswert von Y_0Y_h wird berechnet wie folgt

$$\begin{aligned}
E(Y_0Y_h) &= E((X_{0,h} + X_{1,h})(X_{0,h} + X_{2,h})) \\
&= E(X_{0,h}^2 + X_{0,h}X_{1,h} + X_{0,h}X_{2,h} + X_{1,h}X_{2,h}) \\
&= E(X_{0,h}^2) + E(X_{0,h}X_{1,h}) + E(X_{0,h}X_{2,h}) + E(X_{1,h}X_{2,h}) \\
&= E(X_{0,h}^2) + E(X_{0,h})E(X_{1,h}) + E(X_{0,h})E(X_{2,h}) \\
&\quad + E(X_{1,h})E(X_{2,h}) \\
&= rK(-h) + r^2K(-h)^2 + 2rK(-h)(r(1 - K(-h))) \\
&\quad + (r(1 - K(-h)))^2 \\
&= rK(-h) + r^2K(-h)^2 + 2r^2K(-h) \\
&\quad - 2r^2K(-h)^2 + r^2 - 2r^2K(-h) + r^2K(-h)^2 \\
&= r^2 + rK(-h).
\end{aligned}$$

Somit folgt für die Kovarianz

$$Cov(Y_0Y_h) = r^2 + rK(-h) - r^2 = rK(-h).$$

Die Kovarianz von Y_0 und Y_h hängt nur von h und nicht vom Zeitpunkt t ab, deshalb ist die Zeitreihe R_t schwach stationär⁴⁸. \square

Im nächsten Abschnitt werden die Renditen nochmals näher betrachtet. Es sollen verschiedene Eigenschaften der Zeitreihe bestimmt werden, die Aufschluss über den Kern K und der Zeitabhängigkeit geben.

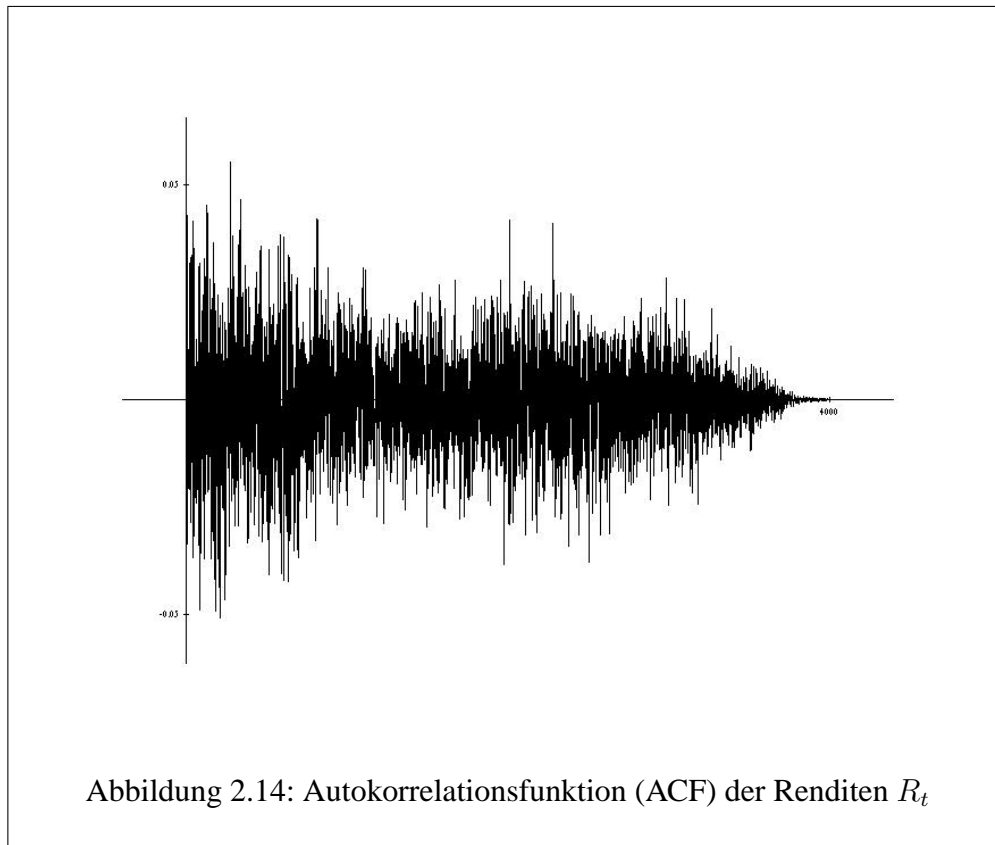
2.4.3.2 Untersuchung der absoluten Renditen

Für die Renditen R_t der Form $R_t = \sigma_t \epsilon_t$ gilt

$$\begin{aligned}
Cov(R_i, R_j) &= E((R_i - E(R_i))(R_j - E(R_j))) \\
&= E(R_iR_j - E(R_i)R_j - E(R_j)R_i + E(R_i)E(R_j)) \\
&= E(R_iR_j) - E(R_i)E(R_j) \\
&= E(\sigma_i \epsilon_i \sigma_j \epsilon_j) - E(\sigma_i \epsilon_i)E(\sigma_j \epsilon_j) \\
&= E(\sigma_i \epsilon_i \sigma_j)E(\epsilon_j) - E(\sigma_i)E(\epsilon_i)E(\sigma_j)E(\epsilon_j) \\
&= 0, \quad i \neq j.
\end{aligned}$$

⁴⁸Wegen der Konstruktion der Zeitreihe ist sie auch streng stationär.

Die Renditen R_t sind somit unkorreliert, d. h. aber auch, dass die Autokorrelationsfunktion der Renditen keine Aussage über die Zeitabhängigkeit liefern kann (siehe Abbildung 2.14⁴⁹). Anstatt den Renditen R_t werden jetzt die absoluten Ren-



reiten $|R_t|$ betrachtet. Die Betrachtung der absoluten Renditen wird durch die Autokorrelationsfunktion (siehe Abbildung 2.15) inspiriert⁵⁰.

Es wird eine Approximation der Autokorrelationsfunktion (ACF) für die Zeitreihe der absoluten Renditen $|R_t|$ für kleine h ($h \rightarrow 0$) angeben.

Satz 3 (Approximation der ACF über die $|R_t|$).

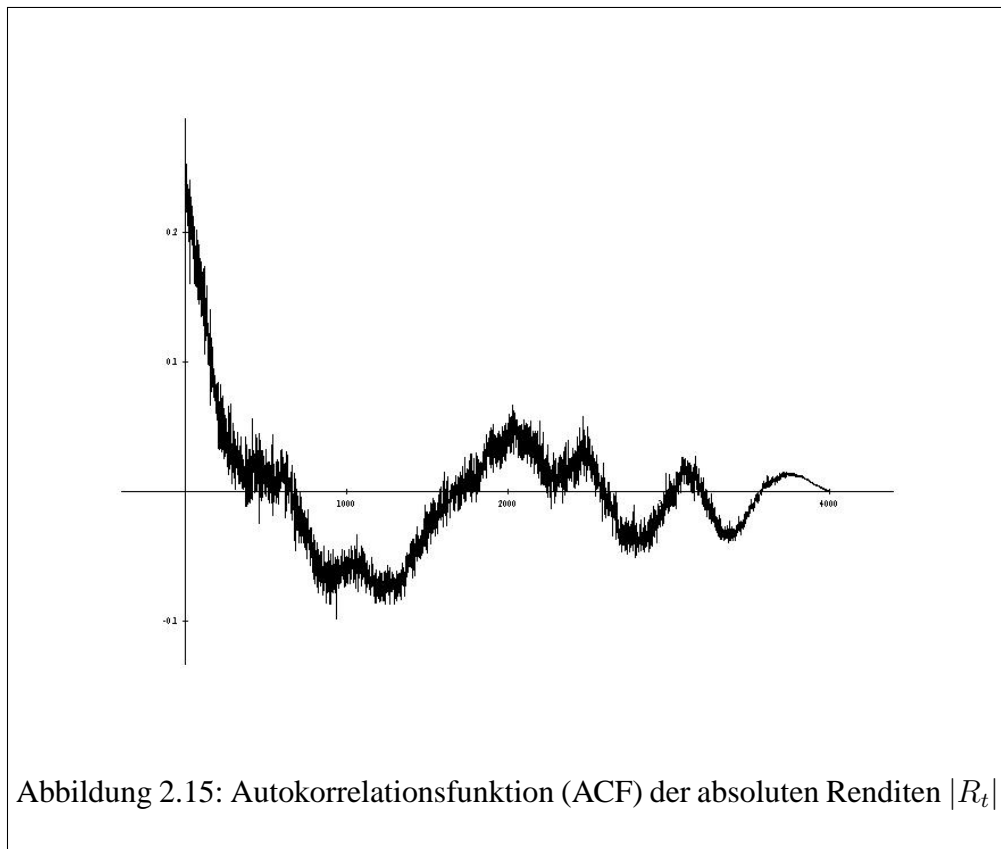
Für $h \rightarrow 0$ gelten folgende Approximationen

1. Falls $r > 3$, dann

$$ACF(h) = 1 - c(r)(1 - K(-h))(1 + o(1 - K(-h))), \quad (2.68)$$

⁴⁹Es wurde eine Autokorrelationsfunktion für einen Simulationsdatensatz mit einem Betakern geplottet.

⁵⁰Es liegt der Simulationsdatensatz aus Abbildung 2.14 zu Grunde.



mit

$$c(r) = \frac{r}{\left(E(Y_0^{-1}) - E(Y_0^{-1/2})^2 \right) (r-1)^2 (r-2)}. \quad (2.69)$$

2. Falls $r > 2$, dann

$$ACF(h) = 1 - c(r)(1 - K(-h))(1 + O(1 - K(-h))) \quad (2.70)$$

mit $c(r)$ wie oben.

Beweis. Für eine gamma-verteilte Zufallsvariable Y mit Parameter r gilt

$$E(Y^\beta) = \Gamma(r + \beta) / \Gamma(r), \quad r + \beta > 0. \quad (2.71)$$

Es gilt für die Kovarianz der absoluten Renditen

$$\begin{aligned}
Cov(|R_0|, |R_h|) &= E((|\epsilon_0\sigma_0| - E(|\epsilon_0\sigma_0|)) (|\epsilon_h\sigma_h| - E(|\epsilon_0\sigma_0|))) \\
&= E(|\epsilon_0\sigma_0\epsilon_h\sigma_h| - E(|\epsilon_0\sigma_0|)(|\epsilon_h\sigma_h|) \\
&\quad - E(|\epsilon_0\sigma_0|)(|\epsilon_0\sigma_0|) + E(|\epsilon_0\sigma_0|)^2) \\
&= E\left(|\epsilon_0|\sigma\sqrt{\frac{r}{Y_0}}|\epsilon_h|\sigma\sqrt{\frac{r}{Y_h}}\right) - E\left(|\epsilon_0|\sigma\sqrt{\frac{r}{Y_0}}\right)^2 \quad (2.72) \\
&= \sigma^2 r \left(E\left(\frac{\epsilon_0^2}{\sqrt{Y_0 Y_h}}\right) - E\left(\frac{\epsilon_0}{\sqrt{Y_0}}\right)^2 \right) \\
&= \sigma^2 r Cov(Y_0^{-1/2}, Y_h^{-1/2}).
\end{aligned}$$

Die Kovarianz von $Y_0^{-1/2}$ und $Y_h^{-1/2}$ wird jetzt berechnet. Dazu werden die Y_t genau wie bei der Berechnung der Kovarianz von Y_0 und Y_h als Summe betrachtet. Die einzelnen Summanden haben die gleiche Gestalt wie in 2.66. Es gilt

$$Cov\left(Y_0^{-1/2}, Y_h^{-1/2}\right) = E\left((Y_0 Y_h)^{-1/2}\right) - E\left(Y_0^{-1/2}\right)^2.$$

Der Erwartungswert von $Y_0^{-1/2}$ lässt sich leicht mit Formel 2.71 berechnen und damit auch dessen Quadrat

$$\left(E\left(Y_0^{-1/2}\right)\right)^2 = \left(\frac{\Gamma(r-1/2)}{\Gamma(r)}\right)^2, \quad r > 1/2. \quad (2.73)$$

Dieser Erwartungswert und dessen Quadrat wurde in Abhängigkeit von r in Abbildung (2.16) ausgegeben. Nun wird $E\left((Y_0, Y_h)^{-1/2}\right)$ betrachtet, dabei ist wieder

$$Y_0 = X_{0,h} + X_{1,h} \text{ und } Y_h = X_{0,h} + X_{2,h}. \quad (2.74)$$

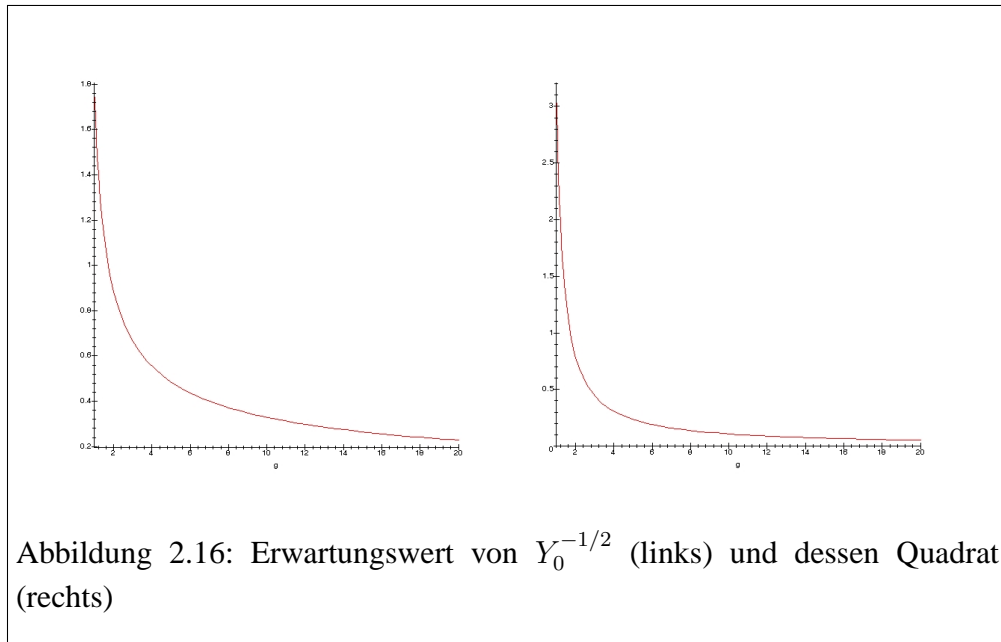
$X_{0,h}$ ist standard-gamma-verteilt mit Parameter $rK(-h)$ sein, $X_{1,h}$ und $X_{2,h}$ mit Parameter $r(1-K(-h))$. Für die Berechnung der Approximation wird die rekursive Definition der Gammaverteilung

$$\Gamma(\lambda + 1) = \lambda\Gamma(\lambda), \quad \lambda > 0 \quad (2.75)$$

benötigt, ebenso wie folgende Ungleichung⁵¹

$$1 - \frac{x+y}{2} \leq ((1+x)(1+y))^{-1/2} \leq 1 - \frac{x+y}{2} + \frac{3(x+y)^2}{8}. \quad (2.76)$$

⁵¹Details zur Ungleichung kann Anhang B.1.1.1 auf Seite 182 entnommen werden.



Es folgt somit

$$\begin{aligned}
E((Y_0 Y_h)^{-1/2}) &= E((X_{0,h} + X_{1,h})^{-1/2} (X_{0,h} + X_{2,h})^{-1/2}) \\
&= E\left(X_{0,h}^{-1} \left(\left(1 + \frac{X_{1,h}}{X_{0,h}}\right) \left(1 + \frac{X_{2,h}}{X_{0,h}}\right) \right)^{-1/2}\right) \\
&= E(Y_0^{-1}) - E(X_{1,h})(1 + O(E(X_{1,h})))E(Y_0^{-2}) \\
&\quad \text{mit } X_{0,h} \approx Y_0, h \rightarrow 0 \\
&= E(Y_0^{-1}) - E(X_{1,h})(1 + O(E(X_{1,h}))) \\
&\quad \frac{\Gamma(r-2)}{\Gamma(r)} \\
&= E(Y_0^{-1}) - E(X_{1,h})(1 + O(E(X_{1,h}))) \\
&\quad \frac{\Gamma(r-2)}{(r-1)(r-2)\Gamma(r-2)} \\
&= E(Y_0^{-1}) - E(X_{1,h})(1 + O(E(X_{1,h}))) \\
&\quad \frac{1}{(r-1)(r-2)}.
\end{aligned}$$

Damit folgt für die Kovarianz von $(Y_0 Y_h)^{-1/2}$

$$\begin{aligned}
Cov(Y_0^{-1/2}, Y_h^{-1/2}) &= E((Y_0 Y_h)^{-1/2}) - E\left(Y_0^{-1/2}\right)^2 \\
&= E(Y_0^{-1}) - \frac{E(X_{1,h})(1 + O(E(X_{1,h})))}{(r-1)(r-2)} - E\left(Y_0^{-1/2}\right)^2 \\
&= \frac{\Gamma(r-1)}{\Gamma(r)} - \frac{r(1-K(-h))(1+O(1-K(-h)))}{(r-1)(r-2)} \\
&\quad - \left(\frac{\Gamma(r-1/2)}{\Gamma(r)}\right)^2 \\
&= \frac{1}{r-1} - \frac{r(1-K(-h))(1+O(1-K(-h)))}{(r-1)(r-2)} \\
&\quad - \left(\frac{\Gamma(r-1/2)}{\Gamma(r)}\right)^2.
\end{aligned}$$

Die Autokorrelation für $h \rightarrow 0$ ist somit gegeben durch

$$\begin{aligned}
ACF(h) &= \frac{\sigma^2 r Cov(Y_0^{-1/2}, Y_h^{-1/2})}{\sqrt{Var(|R_0|)Var(|R_h|)}} \\
&= \frac{1}{r-2} Cov(Y_0^{-1/2}, Y_h^{-1/2}) \\
&= \frac{1}{r-2} \\
&\quad \cdot \left(E(Y_0^{-1}) - E(X_{1,h})(1 + O(E(X_{1,h})))E(Y_0^{-2}) - E(Y_0^{-1/2})^2\right) \\
&= \frac{1}{r-2} \\
&\quad \cdot \left(E(Y_0^{-1}) - r(1-K(-h))(1+O(1-K(-h)))E(Y_0^{-2}) - E(Y_0^{-1/2})^2\right) \\
&= \frac{1}{r-2} \\
&\quad \cdot \left(\left(E(Y_0^{-1}) - E(Y_0^{-1/2})^2\right) - r(1-K(-h))(1+O(1-K(-h)))E(Y_0^{-2})\right) \\
&= \frac{1}{r-2} \left(E(Y_0^{-1}) - E(Y_0^{-1/2})^2\right) \\
&\quad - \frac{r}{(r-1)(r-2)}(1-K(-h))(1+O(1-K(-h))) \\
&= 1 - \frac{1}{(r-1)^2(r-2)} \left(E(Y_0^{-1}) - E(Y_0^{-1/2})^2\right) \\
&\quad \cdot (1-K(-h))(1+O(1-K(-h))),
\end{aligned}$$

mit $Var(R_h) = \sigma \frac{r}{r-2}$, $r > 2$. Die Herleitung von $O()$ und $o()$ kann in Reiss und Spillmann [72] nachvollzogen werden. \square

Im nächsten Ansatz werden die quadratischen Renditen betrachtet.

Satz 4 (Approximation der ACF über die R_t^2). Für $h > 0$ gilt für die ACF der quadratischen Renditen

$$ACF(h) \geq \frac{(r-1)^2(r-2)}{2r-1} \cdot \left(\frac{1}{a_1 a_2} - \frac{2r(1-K(-h))}{a_1 a_2 a_3} - \frac{(r(1-K(-h)))^2}{a_1 a_2 a_3 a_4} - \frac{1}{(r-1)^2} \right), \quad (2.77)$$

mit $a_1 := rK(-h) - 1$, $a_2 := rK(-h) - 2$, $a_3 := rK(-h) - 3$ und $a_4 := rK(-h) - 4$ (Dabei ist zu beachten, dass r so gewählt werden muss, dass kein Nenner null wird.).

Beweis. Für die Kovarianz von R_0^2 und R_h^2 , $h > 0$, $h \in \mathbb{N}$, gilt

$$\begin{aligned} Cov(R_0^2, R_h^2) &= E \left([\epsilon_0^2 \sigma_0^2 - E(\epsilon_0^2 \sigma_0^2)] [\epsilon_h^2 \sigma_h^2 - E(\epsilon_h^2 \sigma_h^2)] \right) \\ &= E(\epsilon_0^2 \sigma_0^2 \epsilon_h^2 \sigma_h^2) - E(\epsilon_0^2 \sigma_0^2)^2 \\ &= E(\epsilon_0^2) E(\epsilon_h^2) E(\sigma_0^2 \sigma_h^2) - E(\epsilon_0^2)^2 E(\sigma_0^2)^2. \end{aligned} \quad (2.78)$$

Da $\epsilon_i \sim N_{0,1}$ unabhängig identisch sind, folgt, dass $\epsilon_i^2 \chi^2(1)$ -verteilt ist und $E(\epsilon_i^2) = 1$ sowie $Var(\epsilon_i^2) = 2$ ist. Mit diesen Resultaten gilt

$$\begin{aligned} Cov(R_0^2, R_h^2) &= E(\epsilon_0^2) E(\epsilon_h^2) E(\sigma_0^2 \sigma_h^2) - E(\epsilon_0^2)^2 E(\sigma_0^2)^2 \\ &= E(\sigma_0^2 \sigma_h^2) - E(\sigma_0^2)^2 \\ &= E \left(\left(\sigma \sqrt{\frac{r}{Y_0}} \right)^2 \left(\sigma \sqrt{\frac{r}{Y_h}} \right)^2 \right) - E \left(\left(\sigma \sqrt{\frac{r}{Y_0}} \right)^2 \right)^2 \\ &= \sigma^4 r^2 E(Y_0^{-1} Y_h^{-1}) - \sigma^4 r^2 E(Y_0^{-1})^2 \\ &= \sigma^4 r^2 Cov(Y_0^{-1}, Y_h^{-1}). \end{aligned} \quad (2.79)$$

Aus (2.71) folgt

$$\begin{aligned}
E(Y_0^{-1})^2 &= \left(\frac{\Gamma(r-1)}{\Gamma(r)} \right)^2, \quad r > 1 \\
&= \left(\frac{\Gamma(r-1)}{(r-1)\Gamma(r-1)} \right)^2 \\
&= \frac{1}{(r-1)^2}.
\end{aligned} \tag{2.80}$$

Jetzt wird der Erwartungswert $E(Y_0^{-1}Y_h^{-1})$ berechnet: Y_0 und Y_h haben die gleiche Darstellung wie in (2.74). Es wird die Ungleichung⁵²

$$1 - x - y - xy \leq \frac{1}{(1+x)(1+y)} \tag{2.81}$$

für $x, y \in [0, \infty)$ benötigt. Dann folgt für den Erwartungswert

$$\begin{aligned}
E(Y_0^{-1}Y_h^{-1}) &= E((X_{0,h} + X_{1,h})^{-1}(X_{0,h} + X_{2,h})^{-1}) \\
&= E\left(X_{0,h}^{-2} \left(\left[1 + \frac{X_{1,h}}{X_{0,h}}\right] \left[1 + \frac{X_{2,h}}{X_{0,h}}\right] \right)\right) \\
&\geq E\left(X_{0,h}^{-2} \left(1 - \frac{X_{1,h}}{X_{0,h}} - \frac{X_{2,h}}{X_{0,h}} - \frac{X_{1,h}X_{2,h}}{X_{0,h}X_{0,h}} \right)\right) \\
&= E(X_{0,h}^{-2}) - E(X_{0,h}^{-3})E(X_{1,h}) - E(X_{0,h}^{-3})E(X_{2,h}) \\
&\quad - E(X_{1,h})E(X_{2,h})E(X_{0,h}^{-4}) \\
&= \frac{\Gamma(rK(-h) - 2)}{\Gamma(rK(-h))} - \frac{\Gamma(rK(-h) - 3)}{\Gamma(rK(-h))}r(1 - K(-h)) \\
&\quad - \frac{\Gamma(rK(-h) - 3)}{\Gamma(rK(-h))}r(1 - K(-h)) \\
&\quad - \frac{\Gamma(rK(-h) - 4)}{\Gamma(rK(-h))}r(1 - K(-h))r(1 - K(-h)) \\
&= \frac{1}{(rK(-h) - 1)(rK(-h) - 2)} \\
&\quad - \frac{2r(1 - K(-h))}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)} \\
&\quad - \frac{((r(1 - K(-h)))^2}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)(rK(-h) - 4)}.
\end{aligned} \tag{2.82}$$

⁵²Der Beweis für die Ungleichung kann Anhang B.1.1.1 auf Seite 182 entnommen werden.

Die Autokorrelationsfunktion für die quadratischen Renditen R_t^2 ist gegeben durch

$$\begin{aligned} ACF(h) &= \frac{Cov(R_0^2, R_h^2)}{\sqrt{Var(R_0^2)Var(R_h^2)}} \\ &= \frac{\sigma^4 r^2 Cov(Y_0^{-1}, Y_h^{-1})}{\sqrt{Var(R_0^2)Var(R_h^2)}}. \end{aligned} \quad (2.83)$$

Für die Varianz von R_i^2 gilt

$$\begin{aligned} Var(R_i^2) &= Var(\sigma_i^2 \epsilon_i^2) \\ &= Var(\sigma_i^2)Var(\epsilon_i^2) + E(\sigma_i^2)^2 Var(\epsilon_i^2) + E(\epsilon_i^2)^2 Var(\sigma_i^2) \\ &= 2Var(\sigma_i^2) + 2E(\sigma_i^2)^2 + Var(\sigma_i^2) \\ &= \sigma^4 r^2 (3Var(Y_0^{-1}) + 2E(Y_0^{-1})^2), \end{aligned} \quad (2.84)$$

mit

$$E(\epsilon_i^2) = 1 \text{ und } Var(\epsilon_i^2) = 2, \text{ da } \epsilon_i^2 \sim \chi^2(1), \quad (2.85)$$

und

$$\begin{aligned} Var(\sigma_i^2) &= Var\left(\left(\sigma\sqrt{\frac{r}{Y_i}}\right)^2\right) \\ &= Var(\sigma^2 r Y_0^{-1}) \\ &= \sigma^4 r^2 Var(Y_0^{-1}), \end{aligned} \quad (2.86)$$

und

$$\begin{aligned} E(\sigma_i^2)^2 &= E\left(\left(\sigma\sqrt{\frac{r}{Y_0}}\right)^2\right)^2 \\ &= E(\sigma^2 r Y_0^{-1}) \\ &= \sigma^4 r^2 E(Y_0^{-1})^2. \end{aligned} \quad (2.87)$$

Für die Varianz und den Erwartungswert von Y_0^{-1} gilt

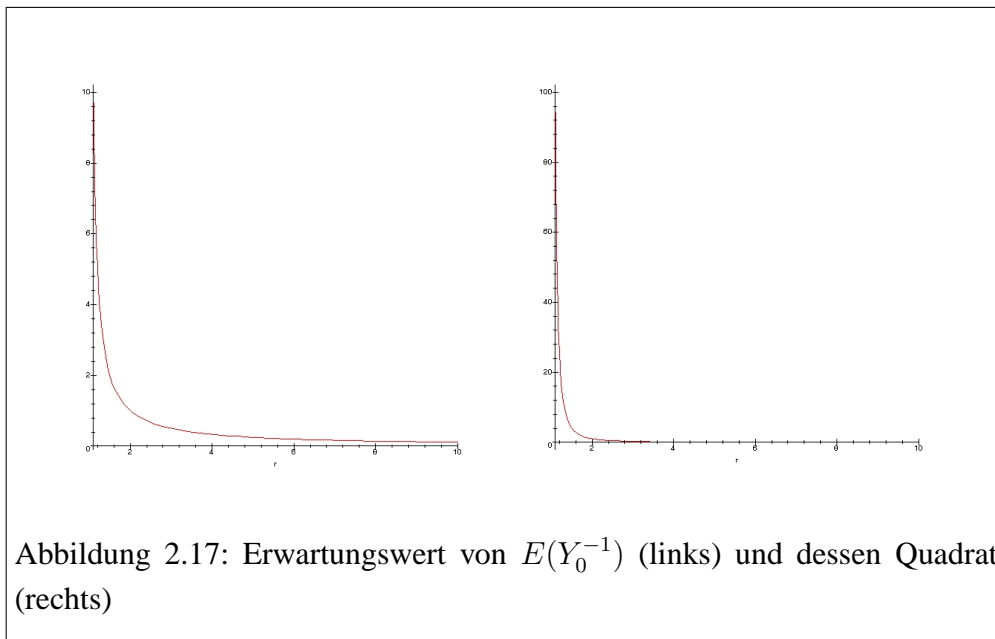
$$E(Y_0^{-1})^2 = \left(\frac{\Gamma(r-1)}{\Gamma(r)}\right)^2 = \frac{1}{(r-1)^2}, \quad (2.88)$$

$$\begin{aligned}
\text{Var}(Y_0^{-1}) &= E((Y_0^{-1})^2) - E(Y_0^{-1})^2 \\
&= E(Y_0^{-2}) - E(Y_0^{-1})^2 \\
&= \frac{\Gamma(r-2)}{\Gamma(r)} - \frac{1}{(r-1)^2} \\
&= \frac{1}{(r-1)(r-2)} - \frac{1}{(r-1)^2}.
\end{aligned}
\tag{2.89}$$

Für den Nenner der Autokorrelationsfunktion ergibt sich

$$\begin{aligned}
\sqrt{\text{Var}(R_0^2)\text{Var}(R_h^2)} &= \text{Var}(R_0^2) \\
&= \sigma^4 r^2 (3\text{Var}(Y_0^{-1}) + 2E(Y_0^{-1})^2) \\
&= \sigma^4 r^2 \left(\frac{3}{(r-1)(r-2)} - \frac{3}{(r-1)^2} + \frac{2}{(r-1)^2} \right) \\
&= \sigma^4 r^2 \left(\frac{3}{(r-1)(r-2)} - \frac{1}{(r-1)^2} \right).
\end{aligned}
\tag{2.90}$$

Damit folgt für die Autokorrelationsfunktion

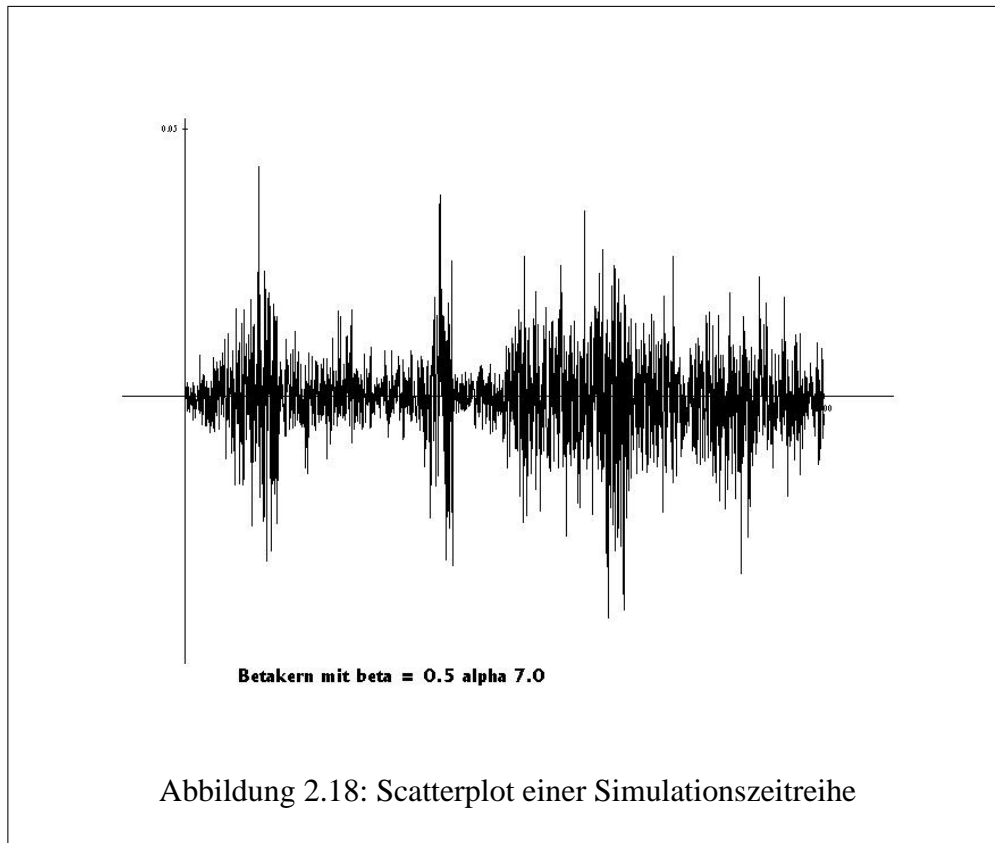


$$\begin{aligned}
ACF(h) &\geq \left(\frac{1}{(rK(-h) - 1)(rK(-h) - 2)} \right. \\
&\quad \left. - \frac{2r(1 - K(-h))}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)} \right. \\
&\quad \left. - \frac{(r(1 - K(-h)))^2}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)(rK(-h) - 4)} - \frac{1}{(r - 1)^2} \right) \\
&\quad \cdot \frac{1}{(3Var(Y_0)^{-1} + 2E(Y_0^{-1})^2)} \\
&= \left(\frac{1}{(rK(-h) - 1)(rK(-h) - 2)} \right. \\
&\quad \left. - \frac{2r(1 - K(-h))}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)} \right. \\
&\quad \left. - \frac{(r(1 - K(-h)))^2}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)(rK(-h) - 4)} - \frac{1}{(r - 1)^2} \right) \\
&\quad \cdot \frac{1}{\left(\frac{3}{(r-1)(r-2)} - \frac{1}{(r-1)^2} \right)} \\
&= \left(\frac{1}{(rK(-h) - 1)(rK(-h) - 2)} \right. \\
&\quad \left. - \frac{2r(1 - K(-h))}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)} \right. \\
&\quad \left. - \frac{(r(1 - K(-h)))^2}{(rK(-h) - 1)(rK(-h) - 2)(rK(-h) - 3)(rK(-h) - 4)} - \frac{1}{(r - 1)^2} \right) \\
&\quad \cdot \frac{(r - 1)^2(r - 2)}{2r - 1}.
\end{aligned} \tag{2.91}$$

□

2.4.3.3 Simulationsalgorithmus

In diesem Abschnitt wird der Algorithmus vorgestellt, der der Simulation der



Zeitreihe zu Grunde liegt. Es werden Aufbau und die wichtigsten Eigenschaften erläutert. Zunächst wird der grobe Ablauf des Algorithmus⁵³ angegeben:

1. Eingabeparameter

- $\alpha > 0$, setze $r = \alpha/2$
- $\sigma > 0$
- $q > 0$, $q \in \mathbb{N}$
- $T > q$, $T \in \mathbb{N}$
- Kern $K(x)$ ⁵⁴ und dessen Parameter

⁵³Ein C++ Listing kann im Anhang B.1 auf Seite 210 entnommen werden.

⁵⁴Auf die Kerne wird später im Detail eingegangen.

2. Berechnung der s_m mittels der Formel

$$s_m = (K(-(m-1)) - 2K(-m) + K(-(m+1))), \quad m = 1, \dots, q.$$

3. Generierung der gamma-verteilten Stichprobe (Zufallsvektor) $\eta_{i,j}$ mit den Parametern r_m , $m = 1, \dots, q$.

4. Berechnung der gamma-verteilten Zufallsvariable Y_t mit der Formel

$$Y_t = \sum_{i=0}^{q-1} \sum_{j=i}^q \eta_{t-i,j}.$$

5. Erzeugung der standard-normalverteilten Zufallsvariable ϵ_t .

6. Erzeugung der Studentischen Stichprobe zum Zeitpunkt t durch

$$Z_t = \epsilon_t \sigma \sqrt{\frac{r}{Y_t}}.$$

7. Verschiebung der gamma-verteilten Zufallsvariablen $\eta_{i,j}$ um einen Zeitpunkt nach hinten (siehe Abbildung 2.12, S. 76) und Erzeugung der gamma-verteilten Stichprobe $\eta_{t,m}$, $m = 1, \dots, q$, mit den Parametern s_m .

8. Wiederholung der Schritt 4 - 7 bis $t = T$, d. h. bis der Stichprobenumfang erreicht ist.

9. Rückgabe der Studentischen Zeitreihe: z_1, \dots, z_t der Z_1, \dots, Z_T .

Für die Simulationen wurden die drei folgenden Kerne verwendet.

- **Beta-Kern**

Der Beta-Kern wurde beim Modellaufbau der Studentischen Zeitreihe erwähnt.

Die Formel lautet

$$K^\beta(x) = (1+x)^{1+\beta}, \quad \beta > 0, \quad -1 \leq x \leq 0. \quad (2.92)$$

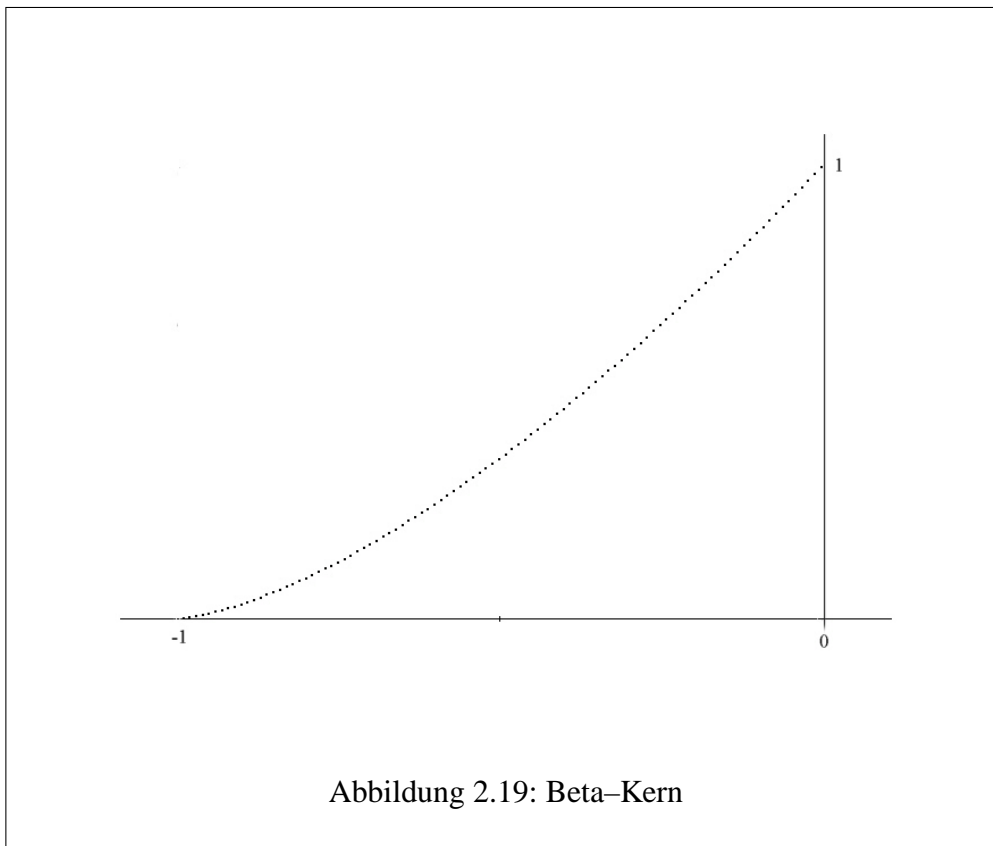
Für $\beta = 0.5$ zeigt Abbildung 2.19 diesen Kern.

- **Gauss-Kern**

Der Gauss-Kern ist wie folgt definiert:

$$K^{Gauss}(x) = \frac{\Phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{-\mu}{\sigma}\right)}, \quad \mu \in \mathbb{R}, \quad \sigma > 0, \quad -1 \leq x \leq 0. \quad (2.93)$$

Abbildung 2.20 zeigt den Kern für $\mu = 1$ und $\sigma = 2$.

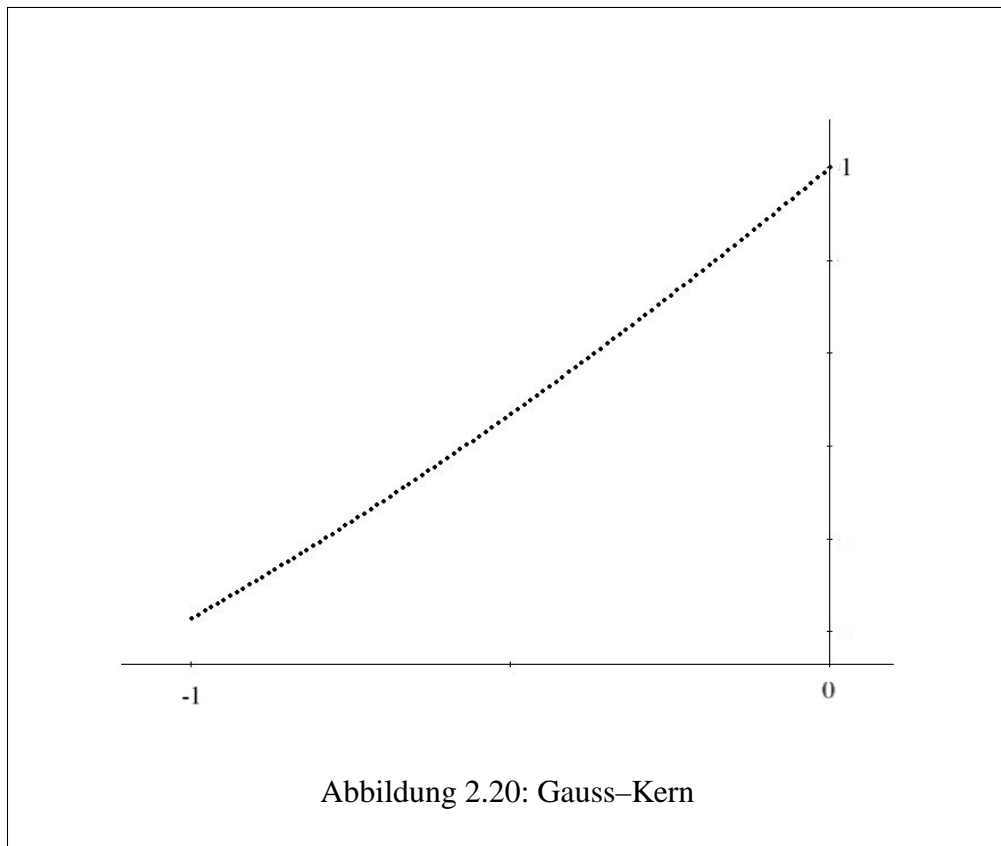


- **Sprung-Kern**

Dieser Kern wurde anhand der Daten des Dax bestimmt. In Satz 3 des letzten Abschnitts wurde der Zusammenhang zwischen der Autokorrelationsfunktion der absoluten Dax-Renditen $|R_t|$ und dem Kern $K(h)$ in der Nähe des Zeitpunktes null erklärt. Die Autokorrelationsfunktion der absoluten Dax-Renditen wurde ausgegeben, wobei ein Sprung vom Zeitpunkt null zum Zeitpunkt eins festgestellt werden (siehe Abbildung 2.21) konnte. Dieser Kern soll ebenfalls einen solchen Sprung in der Autokorrelationsfunktion der absoluten Werte der Stichprobe erzeugen. Für die Simulation wurde folgender Kern (siehe Abbildung 2.22), mit der Formel

$$K^S(x) = \begin{cases} 1 & x = 0 \\ 0.9(x + 1)^{\frac{q}{-1+q}} & -1 \leq x < 0 \end{cases}, q \in \mathbb{N}^+ \quad (2.94)$$

benutzt. Diesen Sprung in der Autokorrelationsfunktion der absoluten Renditen kann nicht nur beim Dax gefunden werden, sondern auch bei anderen



Aktien (siehe Abbildungen B.2, S. 216). Dies ist wiederum eine Eigenschaft die bei den meisten Finanzwerten zu sehen ist. Sie kann als weiterer Stylized Fact formuliert werden:

Durch diesen Stylized Fact kann keine Aussage über den Gewinn oder den Verlust bzgl. der Renditen getroffen werden. Der Stylized Fact zeigt, dass die Stärke der Volatilität schwach von der Volatilität der Vortage abhängt. Dies wird empirisch durch die Volatilitätscluster von Finanzdaten bestätigt.

Stylized Fact ACF:
Die Autokorrelationsfunktion der absoluten Renditen ist schwach positiv korreliert.

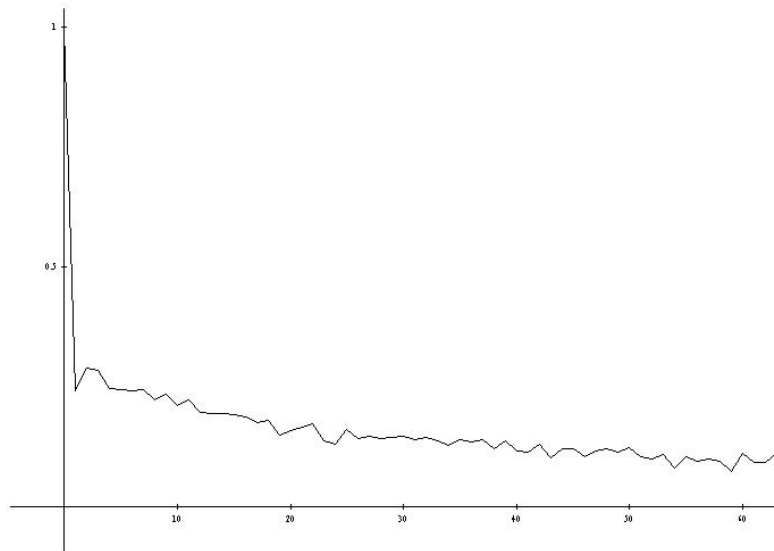


Abbildung 2.21: Autokorrelationsfunktion der absoluten Renditen des Dax

Dieser Stylized Fact tritt bei Stichproben ab einem Zeitraum von einem Jahr auf. Bei Stichproben mit kleinerem Umfang⁵⁵ kann dieser Stylized Fact nicht beobachtet werden, bzw. er ist nicht eindeutig erkennbar.

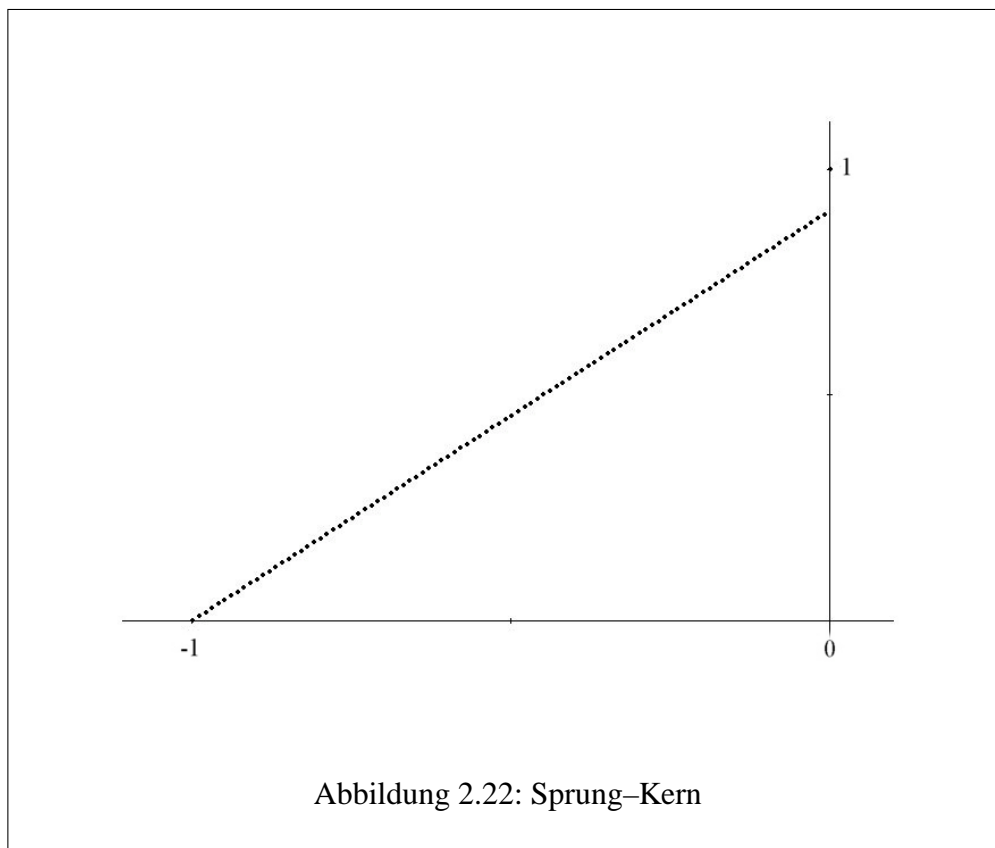
2.4.3.4 Schätzung der Parameter

Nachfolgend sollen die Parameter α und σ einer Simulation der Zeitreihe mit dem Maximum-Likelihood-Schätzer der Studentschen Verteilung geschätzt werden. Der Simulationsdatensatz hat einen Stichprobenumfang von $T = 20000$ und es wurde $\alpha = 5$ und $\sigma = 1$ gewählt. Als Kern wurde der Sprungkern gewählt. Als Abhängigkeitszeitraum q wurde immer der Wert aus der Simulation verwendet. Der Maximum-Likelihood-Schätzer der Studentschen Verteilung kann auf das Zeitreihenmodell nur eingeschränkt angewandt werden. Er überschätzt den Para-

⁵⁵Stichprobenumfang kleiner als 300 Tage

q (Zeitabhängigkeit)	$\hat{\alpha}$	$\hat{\sigma}$
1	10.317	1.41676
2	11.8585	10.1125
5	15.4258	0.787437
10	11.1309	0.59936
20	9.85844	0.457299
50	8.38446	0.290317
80	7.22912	0.227366
100	7.38732	0.21.2875
120	8.27058	0.195494
140	10.0868	0.183728
160	7.31295	0.165856
180	7.52875	0.156389
200	7.51809	0.148617
300	6.80896	0.122185
400	7.25702	0.10723

Tabelle 2.3: Schätzung von α und σ der Zeitreihe mit MLE–Student



meter α unabhängig der gewählten Zeitabhängigkeit (siehe Tabelle 2.3). Für das Zeitreihenmodell sollten neue oder modifizierte Schätzer bestimmt werden. Auf die Entwicklung von neuen Schätzern wird in dieser Arbeit nicht eingegangen.

2.4.4 Dichte der multivariaten Studentischen Verteilung

Wird anstatt eines Finanzwertes ein Portfolio betrachtet, muss ein multivariates Modell zur Risikomessung zu Grunde gelegt werden. In den hier vorgestellten multivariaten Modellen kommt wiederum die Studentische Verteilung⁵⁶ zum Einsatz. Bei der Risikoberechnung muss die Gewichtung der einzelnen Aktien im Portfolio berücksichtigt werden. Dies führte zur Einführung der Gewinn – und Verlustverteilung (siehe Abschnitt 2.3.0.1).

⁵⁶Es werden entweder eine multivariate Studentischen Verteilung oder mindestens univariate Studentische Randverteilungen benutzt.

Im nächsten Abschnitt wird eine multivariate Studentsche Verteilung vorgestellt. Außerdem werden mögliche Schätzer für die Parameter angegeben. Abschließend wird ein Modell, welches auf Copulas basiert, vorgestellt.

Zuvor jedoch wird der Zusammenhang zwischen Risikomaß, Abhängigkeitsmaß und Risikominimierung erklärt.

2.4.5 Risikomaß, Abhängigkeitsmaß und Risikominimierung

Mittels eines Risikomaßes kann eine Aussage über das Auftreten von Verlusten, meist über einer Schranke, getroffen werden. Im univariaten Fall ist die zu Grunde liegende Verlustverteilung eindimensional und kann leicht bestimmt werden. Im multivariaten Fall sieht die Bestimmung der Verlustverteilung schon etwas komplexer aus. Die Verlustverteilungen für die Randverteilungen sind wie im univariaten Fall leicht zu bestimmen, es kommt aber jetzt noch die Abhängigkeit der Randverteilungen zueinander hinzu. In einem multivariaten Modell zur Risikoberechnung ist also die Berechnung der Abhängigkeit sehr wichtig. Markowitz zeigt in seiner Portfolio-Theorie, dass mit der Berechnung der Abhängigkeit der einzelnen Finanzwerte zueinander das Risiko verringert werden kann. Er sagt, dass das Risiko eines Portfolios, welches durch die Varianzen ausgedrückt wird, stets kleiner oder gleich dem Risiko der Summe der Einzelrisiken⁵⁷ ist.

Soll das Risiko eines Portfolios minimiert werden, gibt es zwei mögliche Ansatzpunkte, um bestehende Modelle zu verbessern. Die erste Möglichkeit setzt bei der Modellierung des Risikos an. Durch eine bessere Anpassung der Risikoverteilung kann das Risiko genauer geschätzt werden. Findet man eine Verteilung zur Modellierung der Finanzdaten, die besser an der Realität liegt, so kann trivialerweise auch das Risiko besser geschätzt werden. Dem zweiten Ansatzpunkt liegt die Berechnung der Abhängigkeitsstruktur zu Grunde, kann diese genauer bestimmt werden, so kann das Portfolio an das Ergebnis angepasst werden. Nach Markowitz wird das Risiko des Portfolios dadurch verringert. Durch diese beiden mögliche Ansatzpunkte können bessere Modelle entwickelt werden.

Bei der Berechnung des Risikos in multivariaten Modellen wird durch das Schätzen der Parameter der Risikoverteilung auch automatisch immer die Abhängigkeit mitgeschätzt. Dies führt meist zu komplexen Berechnungen und hat zur Folge, dass bei Austausch eines Finanzwertes im Portfolio alle Parameter der

⁵⁷Das Risiko wird aus den Varianzen der Randverteilungen berechnet.

Risikoverteilung neu geschätzt werden müssen. Die Abhängigkeit muss immer durch ein Risikomaß berücksichtigt werden. Es wird meist durch einen oder mehrere eigene Parameter, wie z. B. die Kovarianzmatrix der Normalverteilung, repräsentiert. In Abschnitt 2.4.7 wird ein Modell mit Copulas vorgestellt. Mit diesem Modell kann das Risiko in zwei Schritten berechnet werden.

Im Folgenden werden verschiedene Abhängigkeitsmaße vorgestellt, und deren Eigenschaften erklärt.

2.4.5.1 Pearson's ρ

Der lineare Korrelationskoeffizient, auch Pearson's ρ genannt, beschreibt die lineare Abhängigkeit zweier Zufallsvariablen zueinander.

Definition 13 (Linearer Korrelationskoeffizient, Pearson's ρ). Sei $(X, Y)^T$ ein Vektor mit zwei Zufallsvariablen, für die gilt $Var(X) < \infty$ und $Var(Y) < \infty$, dann ist Pearson's ρ definiert als

$$\rho = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}. \quad (2.95)$$

Wichtig für die Schätzung des linearen Korrelationskoeffizienten ist dessen empirische Berechnung.

Definition 14 (Empirische Berechnung von Pearson's ρ). Seien x_i Realisationen der Zufallsvariable X und y_i Realisationen der Zufallsvariable Y mit $i = 1, \dots, n$, dann wird ρ empirisch berechnet durch

$$\hat{\rho} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n \sqrt{\bar{s}_1^2 \bar{s}_2^2}}, \quad (2.96)$$

wobei

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i,$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

die Stichprobenmittelwerte von x_i und y_i sind, und

$$\bar{s}_1^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2,$$

$$\bar{s}_2^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

die Stichprobenvarianzen von x_i und y_i sind. In der Stichprobenvarianz muss, falls ein erwartungstreuer Schätzer gewünscht wird, der Faktor $\frac{1}{n}$ durch $\frac{1}{n-1}$ ersetzt werden.

Ein weiteres Abhängigkeitsmaß ist Spearman's ρ . Es beschreibt ebenso die lineare Abhängigkeit zwischen zwei Zufallsvariablen.

2.4.5.2 Spearman's ρ

Definition 15 (Spearman's ρ ⁵⁸). Sei (X, Y) ein Vektor zweier Zufallsvariablen, und (\tilde{X}, \tilde{Y}) und (\check{X}, \check{Y}) unabhängige Kopien des Zufallsvektors, dann ist Spearman's ρ definiert als

$$\begin{aligned} \rho_S(X, Y) = & 3 \left(P \left\{ (X - \tilde{X})(Y - \check{Y}) > 0 \right\} \right) \\ & - 3 \left(P \left\{ (X - \tilde{X})(Y - \check{Y}) < 0 \right\} \right). \end{aligned} \quad (2.97)$$

Spearman's ρ betrachtet nicht die numerischen Werte der Zufallsvariablen wie Pearson's ρ dies tut, sondern deren Ränge.

Definition 16 (Empirische Berechnung von Spearman's ρ). Spearman's ρ lässt sich mit der Formel

$$\hat{\rho}_S = 1 - \frac{6 \sum D^2}{N(N^2 - 1)} \quad (2.98)$$

empirisch berechnen. Dabei ist D die Differenz der Ränge bezüglich X und Y und $N(N^2 - 1)$ gibt die Anzahl der Paarmöglichkeiten an.

Es gilt für Pearson's und Spearman's ρ

$$-1 \leq \rho \leq 1 \text{ und } -1 \leq \rho_S \leq 1. \quad (2.99)$$

2.4.5.3 Kendall's τ

Kendall's τ betrachtet ebenso wie Spearman's ρ_S die Ränge der Zufallsvariablen. Das Abhängigkeitsmaß zählt das Auftreten von gleichem und entgegengesetztem Auftreten der Vorzeichen der Zufallsvariablen.

⁵⁸engl. Spearman Rank Correlation

Definition 17 (Kendall's τ). Sei (X, Y) ein Vektor zweier Zufallsvariablen und (\tilde{X}, \tilde{Y}) eine unabhängige Kopie des Zufallsvektor, dann ist Kendall's τ folgendermaßen definiert

$$\begin{aligned} \tau(X, Y) = & P \left\{ (X - \tilde{X})(Y - \tilde{Y}) > 0 \right\} \\ & - P \left\{ (X - \tilde{X})(Y - \tilde{Y}) < 0 \right\}. \end{aligned} \quad (2.100)$$

Lemma 7 (Eigenschaften).

- $\tau(X, Y) \in [-1, 1]$.
- *Perfekte Abhängigkeit* für $\tau(X, Y) = \pm 1$.
- *Unabhängigkeit* für $\tau(X, Y) = 0$.
- $\tau(X, Y)$ bleibt unverändert bei streng monotonen Transformationen, d. h. ist f streng monoton, dann gilt

$$\tau(f(X), f(Y)) = \tau(X, Y).$$

- Für elliptische Verteilungen läßt sich Kendall's τ mit der Beziehung

$$\tau = \frac{2}{\pi} \arcsin(\rho)$$

berechnen⁵⁹.

Kendall's τ hat gegenüber Pearson's ρ und Spearman's ρ_S die wichtige Eigenschaft, dass es invariant gegenüber monotonen Transformationen ist. Es bietet sich an dieses Abhängigkeitsmaß zu benutzen, weil es wichtige Eigenschaften besitzt und sich außerdem leicht berechnen lässt. Bei der Verwendung von Copulas läßt sich Kendall's τ ebenso leicht berechnen, was die nächsten Sätze zeigen werden. Diese Sätze werden der Einführung von Copulas (siehe Abschnitt 2.4.7, Seite 119 oder [65]) vorweggenommen, da sie die Anwendbarkeit dieser Abhängigkeitsmaße zeigen.

⁵⁹Siehe dazu Lindskog, McNeil und Schmock [55].

Satz 5. Seien $(X, Y)^T$ und $(\tilde{X}, \tilde{Y})^T$ zwei unabhängige Zufallsvektoren stetiger Zufallsvariablen mit der gemeinsamen Verteilung H und \tilde{H} und Randverteilungen F (für X und \tilde{X}) und G (für Y und \tilde{Y}). Seien C und \tilde{C} die Copulas von $(X, Y)^T$ und $(\tilde{X}, \tilde{Y})^T$, so dass gilt $H(x, y) = C(F(x), G(y))$ und $\tilde{H}(x, y) = \tilde{C}(F(x), G(y))$. Sei Q definiert durch

$$Q = P \left\{ (X - (\tilde{X}))(Y - (\tilde{Y})) > 0 \right\} - P \left\{ (X - (\tilde{X}))(Y - (\tilde{Y})) < 0 \right\}. \quad (2.101)$$

Dann gilt für Q im Zusammenhang mit den Copulas

$$Q = Q(C, \tilde{C}) = 4 \int_{[0,1]^2} \tilde{C}(u, v) dC(u, v) - 1. \quad (2.102)$$

Beweis. Siehe Embrechts et al. [22], Seite 11. □

Der Unterschied zwischen dem letzten Satz und dem nächsten Lemma sind die Randverteilungen. In Satz 5 konnten beliebige Randverteilungen gewählt werden, im nachfolgenden Lemma wird nur die Copula betrachtet, d. h. die Randverteilungen sind gleichverteilt auf $[0, 1]$.

Lemma 8 (Berechnung von Kendall's τ bei Copulas⁶⁰). Sei $(X, Y)^T$ ein Zufallsvektor mit zwei stetigen Zufallsvariablen X und Y mit einer Copula C . Dann gilt für Kendall's τ von $(X, Y)^T$

$$\tau(X, Y) = Q(C, C) = 4 \int_{[0,1]^2} C(u, v) dC(u, v) - 1, \quad (2.103)$$

mit $Q(C, C)$ wie oben. U und V seien gleichverteilt auf $[0, 1]$ und es gilt dann $\tau(X, Y) = 4E(C(U, V)) - 1$.

Das nächste Lemma zeigt wie Kendall's τ empirisch berechnet werden kann. Diese Berechnung kann auf beliebige Daten angewandt werden, auch auf Copula-Daten⁶¹.

Definition 18 (Empirische Berechnung von Kendall's τ). Seien x_i die Realisationen des Zufallsvektor $(X, Y)^T$ für $i = 1, \dots, n$. Sei τ_{pos} die relative Häufigkeit der positiven Produkte von $(x_i < x_j)(y_i < y_j) > 0$, d. h.

$$\tau_{pos} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (1_{(x_i < x_j, y_i < y_j)} + 1_{(x_i > x_j, y_i > y_j)}),$$

⁶⁰Embrechts et. al. [22]

⁶¹Das heißt auf einen bivariaten bzw. multivariaten Datensatz mit Randverteilungen auf $[0, 1]$.

und entsprechend

$$\tau_{neg} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (1_{(x_i < x_j, y_i > y_j)} + 1_{(x_i > x_j, y_i < y_j)}).$$

Kendall's τ wird dann empirisch berechnet durch

$$\hat{\tau} = \tau_{pos} - \tau_{neg}. \quad (2.104)$$

Im nächsten Lemma⁶² werden einige Eigenschaften von Kendall's τ und Spearman's ρ_S angegeben.

Lemma 9. *Seien X und Y Zufallsvariablen mit den stetigen Verteilungsfunktionen F_1 und F_2 , gemeinsamer Verteilung F und Copula C . Dann gilt folgendes:*

1. $\rho_S(X, Y) = \rho_S(Y, X)$, $\tau(X, Y) = \tau(Y, X)$.
2. Wenn X und Y unabhängig sind, dann ist $\rho_S(X, Y) = \tau(X, Y) = 0$.
3. $-1 \leq \rho_S(X, Y), \tau(X, Y) \leq +1$.
4. $\rho_S(X, Y) = 12 \int_0^1 \int_0^1 \{C(x, y) - xy\} dx dy$.
5. $\tau(X, Y) = 4 \int_0^1 \int_0^1 C(u, v) dC(u, v) - 1$.
6. Für $T : \mathbb{R} \rightarrow \mathbb{R}$ streng monoton auf dem Bildbereich von X , sind ρ_S und τ invariant gegenüber dieser monotonen Transformation. Es gilt $\rho_S(X, Y) = \rho_S(T(X), Y)$ und $\tau(X, Y) = \tau(T(X), Y)$.
7. $\rho_S(X, Y) = \tau(X, Y) = 1 \Leftrightarrow C = C_u \Leftrightarrow Y = T(X)$, wenn T wachsend ist.
8. $\rho_S(X, Y) = \tau(X, Y) = -1 \Leftrightarrow C = C_l \Leftrightarrow Y = T(X)$, wenn T fallend ist.

Es gilt $C_l = \max\{x_1 + \dots + x_n + 1 - n, 0\}$ und $C_u = \min\{x_1, \dots, x_n\}$.

Beweis. Siehe Embrechts et. al. [24]. □

⁶²siehe Embrechts et. al. [24], Seite 16.

Die bisher vorgestellten Abhängigkeitsmaße beschreiben die Abhängigkeit zweier Zufallsvariablen für die komplette Verteilung. Sehr oft ist es aber der Fall, dass nur die Abhängigkeit in den Flanken von Bedeutung ist. Die folgenden Abhängigkeitsmaße betrachten diese Abhängigkeit, dabei ist immer eine Schranke festzulegen, welche den Beginn der Flanke angibt. Für die empirische Berechnung der Abhängigkeit ist diese Schranke sehr wichtig, da in den Flanken sich je nach Datensatz nur wenige Punkte befinden können.

2.4.5.4 Tail-Dependence χ

Ausgangspunkt für das Abhängigkeitsmaß χ bildet ein bivariater Zufallsvektor (X, Y) .

Definition 19 (χ für beliebige Randverteilungen).

$$\chi = \lim_{z \rightarrow z^*} P(X > z | Y > z),$$

wobei z gegen den rechten Eckpunkt z^* der Randverteilung konvergiert.

Es gibt noch eine zweite Definition von χ über die Copula. U und V seien die Randverteilungen der Copula.

Definition 20 (χ für Copulas, Coles, Heffernan und Tawn [19]).

$$\chi(u) = P(U > u | V > u), \quad u \in [0, 1]. \quad (2.105)$$

Für $u \rightarrow 1$ ist dann χ definiert durch

$$\chi = \lim_{u \rightarrow 1} \chi(u). \quad (2.106)$$

Bemerkung 5. Die Tail-Dependence-Funktion kann mit einer Copula wie folgt ausgedrückt werden:

$$\begin{aligned} \chi(u) &= P(U > u | V > u), \quad u \in [0, 1] \\ &= \frac{P(U > u, V > u)}{P(U > u)} \\ &= \frac{1 - 2u + C(u, u)}{1 - u} \\ &= 2 - \frac{1 - C(u, u)}{1 - u} \\ &\sim 2 - \frac{\log C(u, u)}{\log u}. \end{aligned}$$

$\chi(u)$ besitzt somit folgende Schranken

$$\frac{2 - \log(2u - 1)}{\log(u)} \leq \chi(u) \leq 1. \quad (2.107)$$

Bemerkung 6 (Schätzer für χ , Poon, Rockinger und Tawn [68]).

- **Empirischer Schätzer:** Seien x_i und y_i die Realisationen des Zufallsvektors $(X, Y)^T$ mit $i = 1, \dots, n$. Dann lässt sich χ für eine Schranke $u \in (0, 1]$ berechnen durch

$$\hat{\chi} = \frac{\sum_{i=1}^n 1_{(u, \infty)}(x_i) 1_{(u, \infty)}(y_i)}{\sum_{i=1}^n 1_{(u, \infty)}(y_i)}. \quad (2.108)$$

- **Hill Schätzer:** Seien x_i und y_i die Realisationen des Zufallsvektors (X, Y) mit $i = 1, \dots, n$ und $x_i, y_i \in [0, 1]$. Dann seien s_i und t_i die mit $F(h) = \exp(-1/h)$ auf Fréchet–Randverteilungen transformierten Realisationen des Zufallsvektors (S, T) . Z sei die Zufallsvariable mit der Bedingung $Z = \min(S, T)$, d. h. die Realisation der Zufallsvariable Z sind $z_i = \min(s_i, t_i)$, $i = 1, \dots, n$. Somit kann der Hill–Schätzer⁶³ angewandt werden und es gilt:

$$\hat{\xi} = \frac{1}{n_u} \sum_{i=1}^{n_u} \log \left(\frac{z_i}{u} \right),$$

Dieses Abhängigkeitsmaß liefert bei empirischen Daten keine Aussage über die Abhängigkeit, falls $\chi = 0$. Daher wurde von Coles, Heffernan und Tawn [19] eine Abhängigkeitsmaß $\bar{\chi}$ eingeführt, welches helfen soll diesen Nachteil zu umgehen.

2.4.5.5 Tail–Dependence $\bar{\chi}$

Definition 21 (Tail–Dependence–Funktion). Für $u \in [0, 1)$ ist die Tail–Dependence–Funktion $\bar{\chi}$ durch

$$\bar{\chi}(u) = \frac{2 \log P(U > u)}{\log P(U > u, V > u)} - 1 \quad (2.109)$$

Dies führt zur Definition des Grenzwertes von $\bar{\chi}$

⁶³Der Hill–Schätzer ist nur bedingt anwendbar und hat seine Schwächen.

Definition 22 ($\bar{\chi}$ Tail–Dependence).

$$\bar{\chi} = \lim_{u \rightarrow 1} \bar{\chi}(u) \quad (2.110)$$

Für $\bar{\chi}$ gilt dann $-1 < \bar{\chi} \leq 1$.

Die Tail–Dependence–Funktion $\bar{\chi}$ kann mit einer Copula C und der Survival Copula \bar{C} geschrieben als

$$\begin{aligned} \bar{\chi}(u) &= \frac{2 \log P(U > u)}{\log P(U > u, V > u)} - 1 \\ &= \frac{2 \log(1 - u)}{\log \bar{C}(u, u)} - 1, \quad 0 \leq u \leq 1 \\ &= \frac{2 \log(1 - u)}{\log(1 - 2u + C(u, u))} - 1, \quad 0 \leq u \leq 1. \end{aligned} \quad (2.111)$$

Die Abhängigkeit einer bivariaten Verteilung kann folgendermaßen berechnet werden:

- ($\chi > 0, \bar{\chi} = 1$) Asymptotische Abhängigkeit der Randverteilungen.
- ($\chi = 0, \bar{\chi} < 1$) Asymptotische Unabhängigkeit der Randverteilungen, $\bar{\chi}$ liefert eine Information über die Stärke der Abhängigkeit, außerhalb der Flanken.

Bemerkung 7 (Schätzer für $\bar{\chi}$).

- **Empirischer Schätzer:** Seien x_i und y_i die Realisationen des Zufallsvektors $(X, Y)^T$ mit $i = 1, \dots, n$. Dann lässt sich $\bar{\chi}$ für eine Schranke $u \in (0, 1]$ folgendermaßen berechnen:

$$\hat{\bar{\chi}}(u) = \frac{2 \log \left(\sum_{i=1}^n 1_{(u, \infty)}(y_i) \right)}{\log \left(\sum_{i=1}^n 1_{(u, \infty)}(x_i) 1_{(u, \infty)}(y_i) \right)} - 1. \quad (2.112)$$

- **Hill Schätzer:** Es gelten die Bedingungen von oben. Ebenso sei wieder $Z = \min(S, T)$ und z_i die dazugehörige Stichprobe⁶⁴. Der Hill–Schätzer für $\bar{\chi}$ lässt sich dann berechnen mit:

⁶⁴ S, T sind wieder auf Fréchet–Randverteilungen transformiert (siehe Bemerkung 6).

$$\hat{\chi}(u) = \frac{2}{n_u} \left(\sum_{i=1}^{n_u} \log \left(\frac{z_j}{u} \right) \right) - 1,$$

Bemerkung 8 (Tail–Dependence bei Archimedischen Copulas). Es seien $\phi'(1)$ und $\phi'(0)$ die einseitigen Ableitungen von ϕ . ϕ ist der Erzeuger der Archimedischen Copula⁶⁵. Dann

- implizieren $\phi'(1) = 0$ und $\chi_U = 2 - (\phi^{-1} \circ 2\phi)'(1)$ die obere Tail–Dependence,
- aus $\phi'(1) < 0$ folgt ebenso die obere Tail–Dependence,
- wenn $\phi'(0) > -\infty$ und ϕ nicht strikt⁶⁶ ist, folgt die untere Tail–Dependence,
- die untere Tail–Dependence impliziert $\phi'(0) = -\infty$, ϕ ist strikt, und $\chi_L = (\phi^{-1} \circ 2\phi)'(0)$,

wobei $\chi_L(u) = P(U < u | V < u)$ und $\chi_U(u) = P(U > u | V > u)$ ist.

Beweis. Siehe Schmidt [79]. □

Diese beiden Abhängigkeitsmaße χ und $\bar{\chi}$ beschreiben das gemeinsame Auftreten von Extrema in beiden Zufallsvariablen. Sie sind besonders gut geeignet für den Einsatz im Stress–Testing. Sie hängen beide von einer Schranke u ab. Die Ergebnisse werden durch diese Schranke stark beeinflusst, da sie angibt, in welchem Bereich nach Daten gesucht wird, mit denen geschätzt werden soll.

2.4.6 Studentische Dichte mit einem Gestaltsparameter α

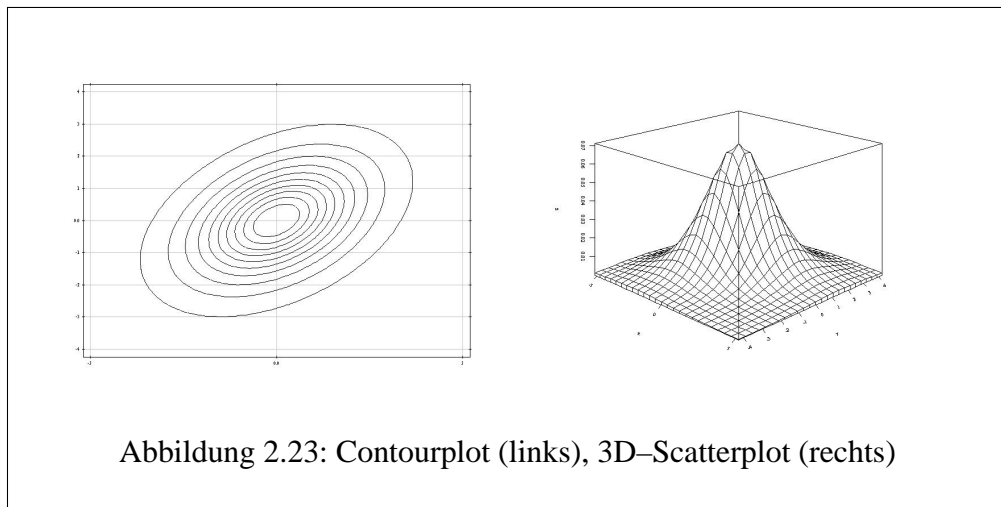
Im den vorherigen Abschnitten wurden nur univariate Modelle zur Modellierung von Finanzdaten dargestellt. Diese Modelle beziehen sich somit nur auf einen Finanzwert oder auf einen Index, wie zum Beispiel den Dax. Der Index ist aber nur dann aussagekräftig für ein Portfolio, wenn er die gleiche Zusammensetzung wie dieses hat. Um das Risiko eines Portfolios mit beliebiger Zusammensetzung bestimmen zu können, wird ein multivariates Modell benötigt. Dem multivariaten Modell liegt eine multivariate Verteilung zu Grunde⁶⁷. Diese Verteilung⁶⁸ soll in den nächsten Abschnitten eine Studentische Verteilung sein.

⁶⁵vgl. Abschnitt 2.4.14.3, S. 142

⁶⁶ ϕ wird strikt genannt, wenn gilt: $\phi(0) = \infty$.

⁶⁷zumindestens eine bivariate Verteilung

⁶⁸Diese Verteilung soll in einem linearen Zusammenhang zur P+L–Verteilung stehen.



Definition 23 (Multivariate Studentsche Dichte, Johnson und Kotz [50]).

$$f_{\alpha, \mu, \Sigma}^{\mathcal{S}^d} = \frac{\Gamma(\frac{1}{2}(\alpha + d))}{(\alpha\pi)^{d/2} \Gamma(\frac{1}{2}\alpha) |\Sigma|^{1/2}} \left(1 + \frac{1}{\alpha} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right)^{-\frac{1}{2}(\alpha + d)}, \quad (2.113)$$

wobei $\frac{\alpha}{\alpha-2} \Sigma$ für $\alpha > 2$ die Kovarianzmatrix dieser Verteilung und $|\Sigma|$ die Determinante von Σ ist.

2.4.6.1 Schätzer

In diesem Abschnitt wird der EM-Schätzer⁶⁹ für multivariate Studentsche Verteilung⁷⁰ (p-dimensional) darstellt. Diese wird bezeichnet mit

$$t_p(\mu, \Sigma, \alpha),$$

und kann ebenso wie im univariaten Fall als Mischung einer Normalverteilung mit einer Gamma-Verteilung angesehen werden. Das heißt, wenn $\mathcal{L}(W) = t_p(\mu, \Sigma, \alpha)$, gilt

$$P(W \in \cdot | U = u) = N(\mu, \Sigma/u). \quad (2.114)$$

⁶⁹Der EM-Schätzer ist ein Maximum-Likelihood Schätzer, und bezieht sich auf den EM-Algorithmus.

⁷⁰t-Verteilung oder Student t-Verteilung

Die zu dem Gewicht u zugehörige Zufallsvariable U ist verteilt nach

$$\Gamma\left(\frac{1}{2}\alpha, \frac{1}{2}\alpha\right) \quad (2.115)$$

Die Gamma-Dichte ist dann gegeben durch

$$f_{r,s}^G(u) = \frac{s^r u^{r-1}}{\Gamma(r)} \exp(-su) 1_{[0,\infty)}(u), \quad r, s > 0. \quad (2.116)$$

Durch Integration nach u kann die Dichte von W berechnet werden, also

$$f(w; \mu, \Sigma, \alpha) = \frac{\Gamma\left(\frac{\alpha+p}{2}\right) |\Sigma|^{-1/2}}{(\pi\alpha)^{p/2} \Gamma(\alpha/2) |\Sigma|^{1/2} \{1 + \delta(w, \mu; \Sigma)/\alpha\}^{\frac{1}{2}(\alpha+p)}}, \quad (2.117)$$

wobei

$$\delta(w, \mu; \Sigma) = (w - \mu)^T \Sigma^{-1} (w - \mu). \quad (2.118)$$

Σ^{-1} bezeichnet die Inverse der Kovarianzmatrix. Es wird zunächst von dem Fall ausgegangen, dass α bekannt ist. Wir nehmen nun an, dass W fehlende Daten besitzt, die nicht beobachtbar sind (siehe Anhang B.1.2.1, S. 184). Die beobachtbaren Daten bezeichnen wir mit w_1, \dots, w_n . Sie sind multivariat Student-sch-verteilt. Es sei nun

$$y = (w_1, \dots, w_n)^T. \quad (2.119)$$

Man bekommt die folgende log-Likelihood-Funktion für die beobachtbaren Daten y

$$\begin{aligned} \log L(\psi) &= \sum_{j=1}^n \log f(w_j; \alpha, \mu, \Sigma) \\ &= -\frac{1}{2} np \log(\pi\alpha) + n \left(\log \Gamma\left(\frac{\alpha+p}{2}\right) - \log \Gamma\left(\frac{\alpha}{2}\right) \right) \\ &\quad - \frac{1}{2} n \log |\Sigma| + \frac{1}{2} n (\alpha + p) \log \alpha \\ &\quad - \frac{1}{2} (\alpha + p) \sum_{j=1}^n \log \left(\alpha + (w_j - \mu)^T \Sigma^{-1} (w_j - \mu) \right) \end{aligned} \quad (2.120)$$

Diese log-Likelihood-Funktion liefert keine geschlossene Form für den Maximum-Likelihood-Schätzer. Es müsste z. B. das Newton-Raphson Verfahren zum Bestimmen einer Lösung verwendet werden, was aber wegen des komplexen Aufbaus der log-Likelihood-Funktion sehr aufwendig wäre. Der EM-Algorithmus

liefert einen einfacheren Algorithmus zur Berechnung des Schätzers. Wir nehmen nun an, dass die beobachtbaren Daten y nicht vollständig sind. Der vollständige Datenvektor x sei gegeben durch

$$x = (y^T, z^T)^T, \quad (2.121)$$

wobei $z = (u_1, \dots, u_n)^T$ die fehlenden Daten⁷¹ bezeichne, die definiert sind durch

$$P(W_j \in \cdot | U_j = u_j) = N(\mu_j, \sigma_j/u_j), \quad (2.122)$$

paarweise unabhängig für $j = 1, \dots, n$ und es gelte

$$U_1, \dots, U_n \sim \Gamma\left(\frac{1}{2}\alpha, \frac{1}{2}\alpha\right). \quad (2.123)$$

Es wird in diesem Fall angenommen, dass die Gewichte u_i nicht beobachtbar sind. Da die Studentsche Verteilung als bedingte Verteilung dargestellt werden kann, kann die Likelihood-Funktion in zwei Faktoren aufgespalten werden, die bedingte Dichte W gegeben z und der Dichte von z . Die log-Likelihood-Funktion⁷² der vollständigen Daten ist dann gegeben durch

$$\log L_c(\psi) = \log L_{1c}(\psi) + a(z), \quad (2.124)$$

mit

$$\begin{aligned} \log_{1c}(\psi) = & -\frac{1}{2}np \log(2\pi) - \frac{1}{2}n \log |\Sigma| \\ & + \frac{1}{2}p \sum_{i=1}^n \log(u_i) - \frac{1}{2} \sum_{i=1}^n u_i (w_i - \mu)^T \Sigma^{-1} (w_i - \mu) \end{aligned} \quad (2.125)$$

und

$$a(z) = -n \log \Gamma\left(\frac{1}{2}\alpha\right) + \frac{1}{2}n\alpha \log\left(\frac{1}{2}\alpha\right) + \frac{1}{2}\alpha \sum_{i=1}^n (\log u_i - u_i) - \sum_{i=1}^n \log u_i \quad (2.126)$$

Der E-Schritt⁷³ der $(k+1)$ -ten Iteration ist dann die Berechnung von

$$Q(\psi; \psi^{(k+1)}) = E(\log L_c(\psi^k)) \quad (2.127)$$

⁷¹ N bezeichnet hier die univariate Normalverteilung.

⁷²Für sie gilt ebenso die Aufspaltung wie bei der Likelihood-Funktion.

⁷³Expectation Step: siehe Anhang B.1.2.1, Seite 184

Die ist der bedingte Erwartungswert der log–Likelihood–Funktion $\log L_c(\psi)$ des kompletten Datensatzes. In diesem Fall muss nur der erste Term $\log L_{1c}(\psi)$ betrachtet werden, da im zweiten Term keine unbekannt Parameter enthalten sind. Der E–Schritt gestaltet sich relativ einfach, da alle u_i nur linear voneinander abhängen. Es wird jedes u_i durch die bedingte Erwartung gegeben w_i ersetzt. Da die u_i gamma–verteilt sind, muss die bedingte Verteilung von U gegeben $W = w$ berechnet⁷⁴ werden,

$$\mathcal{L}(U \in \cdot | W = w) = \Gamma(m_1, m_2) \quad (2.128)$$

mit

$$m_1 = \frac{1}{2}(\alpha + p) \quad (2.129)$$

und

$$m_2 = \frac{1}{2} \{ \alpha + \delta(w, \mu; \Sigma) \}. \quad (2.130)$$

Es gilt nun für die bedingte Erwartung⁷⁵

$$E(U | W = w) = \frac{\alpha + p}{\alpha + \delta(w, \mu; \Sigma)}. \quad (2.131)$$

Die $u_i^{(k)}$ können also bestimmt werden durch

$$u_i^{(k)} = \frac{\alpha + p}{\alpha + \delta(w, \mu^{(k)}; \Sigma)^{(k)}}. \quad (2.132)$$

Hiermit ist die Bestimmung des E–Schritts abgeschlossen. Jetzt muss nur noch der M–Schritt⁷⁶ berechnet werden. Der M–Schritt der n unabhängigen Beobachtungen w_1, \dots, w_n ist die Berechnung des gewichteten Mittelwertsvektors und der gewichteten Kovarianzmatrix⁷⁷. Diese sehen wie folgt aus:

$$\mu^{(k+1)} = \sum_{i=1}^n y_i u_i^{(k)} / \sum_{i=1}^n u_i^{(k)} \text{ gewichtetes Stichprobenmittel,} \quad (2.133)$$

und

⁷⁴ $w \sim N(\mu, \Sigma/u)$

⁷⁵Man behalte in Erinnerung, dass $U|w$ gamma–verteilt ist.

⁷⁶Maximisation–Step: siehe Anhang B.1.2.1, Seite 184.

⁷⁷ $\Sigma/u_1, \dots, \Sigma/u_n$

$$\Sigma^{(k+1)} = \frac{1}{n} \sum_{i=1}^n u_i^{(k)} (w_i - \mu^{(k+1)})(w_i - \mu^{(k+1)})^T \text{ Stichprobenkovarianzmatrix.} \quad (2.134)$$

Der M–Schritt sieht so einfach aus, da die w_i gegebenen u_i normalverteilt sind. Hiermit ist der EM–Algorithmus für die multivariate Studentsche Verteilung mit bekanntem α hergeleitet worden. Jetzt wird davon ausgegangen, dass auch α unbekannt ist. In diesem Fall sieht der E–Schritt folgendermaßen aus. Für die $(k+1)$ –te Iteration muss

$$E_{\psi^{(k)}}(\log U_j | W_j = w_j), \quad j = 1, \dots, n, \quad (2.135)$$

berechnet werden. Dazu wird folgendes Resultat benötigt, wenn $R \sim \Gamma(r, s)$ ist, dann gilt für den Erwartungswert von $\log R$

$$E(\log R) = \psi(r) - \log s, \quad (2.136)$$

mit

$$\psi(s) = \frac{\partial \log \Gamma(s)}{\partial s} = (\partial \Gamma(s) / \partial s) / \Gamma(s). \quad (2.137)$$

Dies ist die Digamma–Funktion. Mit diesem Resultat sieht die bedingte Erwartung U_j gegeben $W_j = w_j$ wie folgt aus:

$$\begin{aligned} E_{\psi^{(k)}}(\log U_i | W_i = w_i) &= \psi \left(\frac{\alpha^k + p}{2} \right) - \log \left[\frac{1}{2} \left\{ \alpha^{(k)} + \delta(w_i, \mu_i^{(k)}; \Sigma^{(k)}) \right\} \right] \\ &= \log u_i^{(k)} + \left\{ \psi \left(\frac{\alpha^k + p}{2} \right) - \log \left(\frac{\alpha^k + p}{2} \right) \right\}, \end{aligned} \quad (2.138)$$

mit

$$u_i^{(k)} = E_{\psi^{(k)}}(U_i | W_i = w_i) = \frac{\alpha^{(k)} + p}{\alpha^{(k)} + \delta(w_i, \mu_i^{(k)}; \Sigma^{(k)})}, \quad j = 1, \dots, n. \quad (2.139)$$

$Q(\psi; \psi^{(k)})$ ist nun

$$\begin{aligned}
Q(\psi; \psi^{(k)}) &= -n \log \Gamma\left(\frac{1}{2}\alpha\right) + \frac{1}{2}n\alpha \log\left(\frac{1}{2}\alpha\right) \\
&\quad + \frac{1}{2}n\alpha \left\{ \frac{1}{n} \sum_{i=1}^n (\log u_i^{(k)} - u_i^{(k)}) + \psi\left(\frac{\alpha^{(k)} + p}{2}\right) - \log\left(\frac{\alpha^{(k)} + p}{2}\right) \right\}
\end{aligned}
\tag{2.140}$$

Abschließend muss der M-Schritt berechnet werden. Dieser sieht für μ und Σ genauso, wie der M-Schritt der $(k+1)$ -ten Iteration im Fall mit bekannten α , aus. Demnach muss nur noch der M-Schritt für α angegeben werden: Es muss die Ableitung

$$\frac{\partial Q(\psi; \psi^{(k)})}{\partial \alpha} = 0
\tag{2.141}$$

berechnet werden. Die Lösung der Gleichung

$$\begin{aligned}
& -\psi\left(\frac{1}{2}\alpha\right) + \log\left(\frac{1}{2}\alpha\right) + 1 \\
& + \frac{1}{n} \sum_{i=1}^n (\log u_i^{(k)} - u_i^{(k)}) + \psi\left(\frac{\alpha^{(k)} + p}{2}\right) - \log\left(\frac{\alpha^{(k)} + p}{2}\right) = 0
\end{aligned}
\tag{2.142}$$

liefert den Schätzer für den Parameter α im $(k+1)$ -ten Iterationsschritt ($\alpha^{(k)}$).

2.4.6.2 Datenanalyse

In diesem Abschnitt wird der vorgestellte EM-Algorithmus auf einen bivariaten Datensatz angewandt. Der Datensatz besteht aus den Renditen der Aktien Commerzbank und Degussa vom 08.02.1999 bis 01.03.2004. Beide Aktien sind im Dax enthalten. Zunächst wird jeder Datensatz mit dem Maximum-Likelihood-Schätzer der univariaten Studentschen Verteilung geschätzt. Dies liefert folgendes Resultat für die Verteilung der Renditen: Die beiden α 's der Randverteilun-

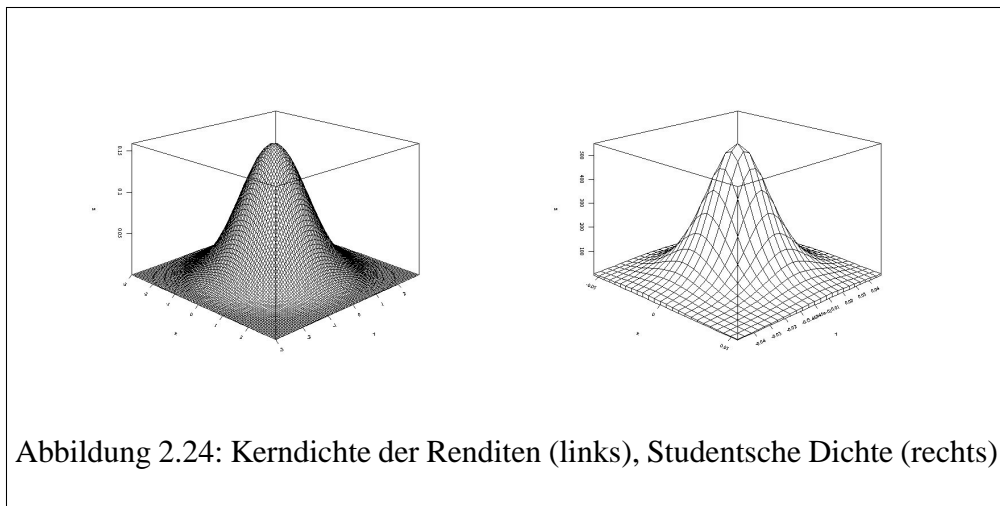
	Commerzbank	Degussa
α	3.06933	5.32537
σ	0.0169691	0.0169233

Tabelle 2.4: Geschätzte Parameter für die Commerzbank- und Degussa-Aktie

gen unterscheiden sich stark voneinander (siehe Tabelle 2.4). Jetzt wird der EM-Schätzer auf den bivariaten Datensatz der Aktienrenditen mit folgendem Ergebnis angewandt:

$$\alpha = 4.09068, \Sigma = \begin{pmatrix} 0.000323729 & 7.34756e - 005 \\ 7.34756e - 005 & 0.000275413 \end{pmatrix}$$

Die bivariate Kerndichte der Aktienrenditen (siehe Abbildung 2.24 (links)) im Vergleich zur der bivariaten Studentschen Dichte (siehe Abbildung 2.24 (rechts)) mit den geschätzten Parametern sieht folgendermaßen aus: Mit diesen geschätz-



ten Parametern wurde ein Datensatz mit Stichprobenumfang $n = 5000$ simuliert. Dieser hat die geschätzten Parameter,

$$\alpha = 4.13939 \text{ und } \Sigma = \begin{pmatrix} 0.000323128 & 7.7291e - 005 \\ 7.7291e - 005 & 0.000271587 \end{pmatrix}.$$

Für diesen Datensatz sieht die Kerndichte wie folgt (siehe Abb. 2.25) aus, die Randverteilungen haben die geschätzten Parameter

1. Randverteilung

$$\alpha = 3.98782 (0.91849, -1.33745)$$

$$\sigma = 0.0178983 (0.0009292, 0.000975),$$

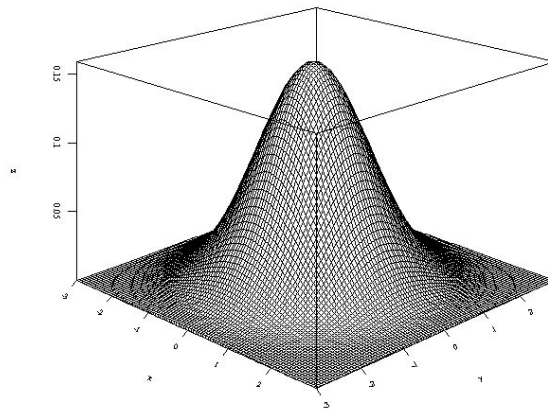


Abbildung 2.25: Kerndichte des Studentischen Datensatzes

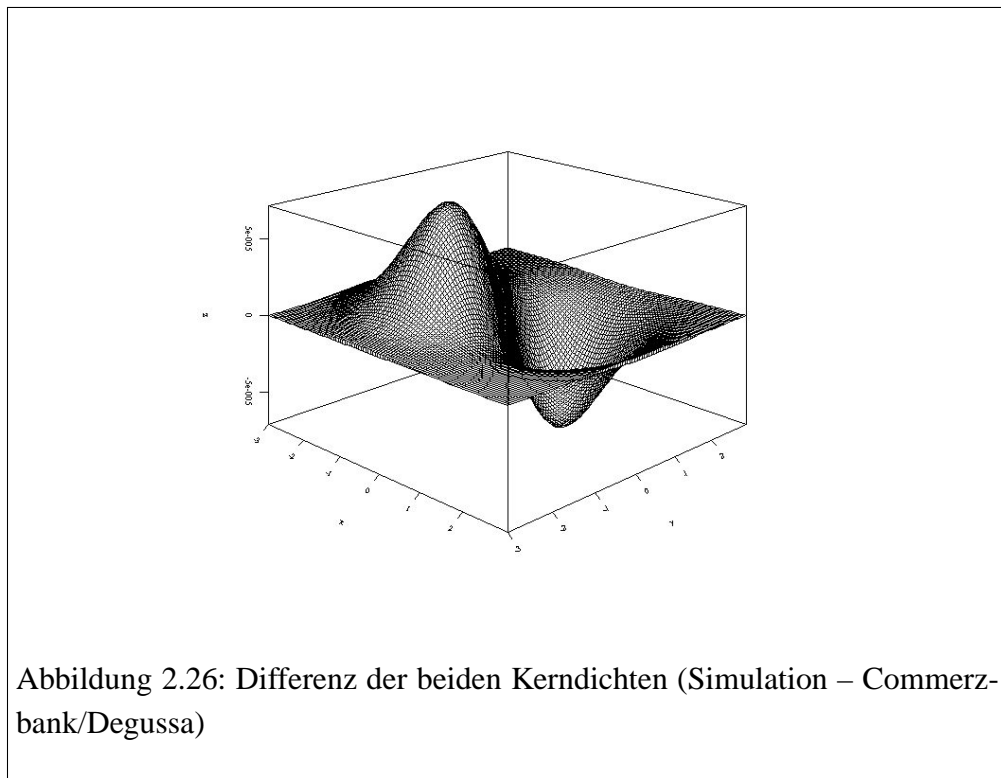
2. Randverteilung

$$\alpha = 4.33104 (1.26171, -0.99423)$$

$$\sigma = 0.0166577 (-0.0003114, -0.0002656).$$

In den Klammern stehen die Differenzen⁷⁸ zu den geschätzten Parametern der Commerzbank-Aktie (erster Wert) und Degussa-Aktie (zweiter Wert). Abbildung 2.26 gibt die Differenz zwischen der Kerndichte der Renditen und der Kerndichte des Studentischen Datensatzes wieder. Der EM-Algorithmus unter- und überschätzt in bestimmten Bereichen die Dichte der Finanzdaten. In diesem Beispiel wurden α 's in den Randverteilungen gewählt, die sich stark unterscheiden. Im nächsten Beispiel wurden zwei Aktien aus dem Dax untersucht, die ein ähnliches α besitzen. Es sind die Aktien von Linde und Henkel über den Zeitraum 17.8.1992 bis 01.03.2004. Im Vergleich dazu wurde rechts die Kerndichte eines

⁷⁸Parameter des Datensatzes minus Parameter der Aktie



	Henkel	Linde
α	3.34878	3.2948
σ	0.0123844	0.0124574

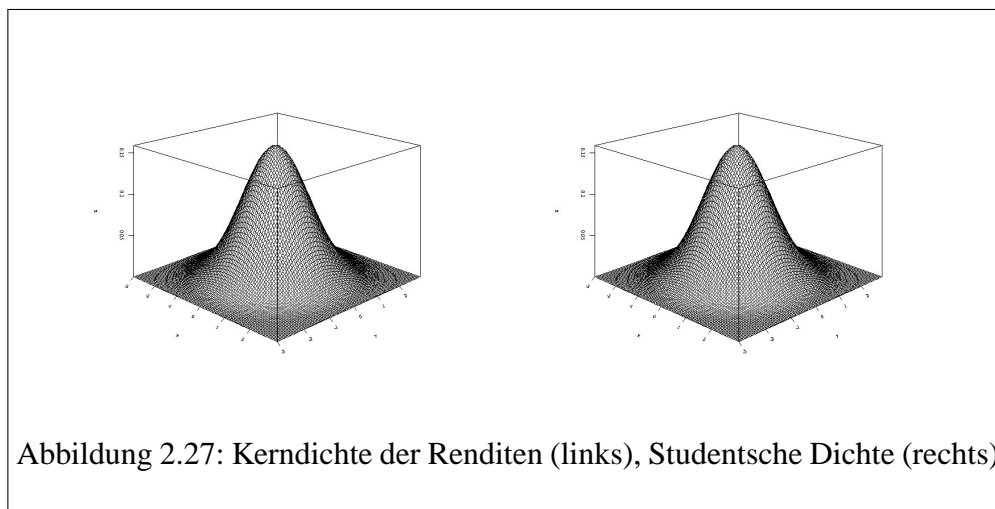
Tabelle 2.5: Geschätzte Parameter für die Henkel– bzw. Linde–Aktien

simulierten bivariaten Studentischen–Datensatz⁷⁹ mit den geschätzten Parametern des EM–Algorithmus ausgegeben (siehe Abb. 2.27).

$$\alpha = 3.11605, \Sigma = \begin{pmatrix} 0.000151314 & 4.6172e - 005 \\ 4.6172e - 005 & 0.000152063 \end{pmatrix}$$

Die Randverteilungen des simulierten Datensatzes haben folgende geschätzten Parameter: (In der Klammer wurde die Differenz zu dem geschätzten Parameter der Henkel– und Linde–Aktie angegeben.)

⁷⁹Stichprobenanzahl = 5000



1. Randverteilung

$$\alpha = 3.0687 \text{ } (-0.28008, -0.2261)$$

$$\sigma = 0.0121781 \text{ } (-0.0002793, -0.0002063),$$

2. Randverteilung

$$\alpha = 3.13606 \text{ } (-0.21272, -0.15874)$$

$$\sigma = 0.012388 \text{ } (3.6e - 06, -6.94e - 05).$$

Es wird der Fehler zwischen der Kerndichte der simulierten Daten und der Finanzdaten geplottet (siehe Abb. 2.28). Der simulierte Datensatz unter- und überschätzt auch hier die Kerndichte der Aktien. Im Fall von ähnlichen α 's ist der Fehler⁸⁰ nicht so groß wie bei verschiedenen α 's⁸¹. In Abbildung 2.29 wurde der Fehler der Aktien mit verschiedene α ⁸² minus dem Fehler der Aktien mit ähnlichem α ⁸³ geplottet. In dieser Abbildung hat der Fehler der Kerndichte mit ähnlichem α keine Aussagekraft. Um die Finanzdaten besser anpassen zu können, müsste ein Modell betrachtet werden, welches verschiedene α 's in den Randverteilungen zulässt. Eine solche multivariate Studentsche Verteilung hat einen sehr komplexen Aufbau. Für geringe Dimensionen ist die Angabe einer Dichte noch möglich. Das

⁸⁰maximal 1e-005

⁸¹Der Fehler beträgt hier maximal 5e-005.

⁸²Commerzbank und Degussa

⁸³Henkel und Linde

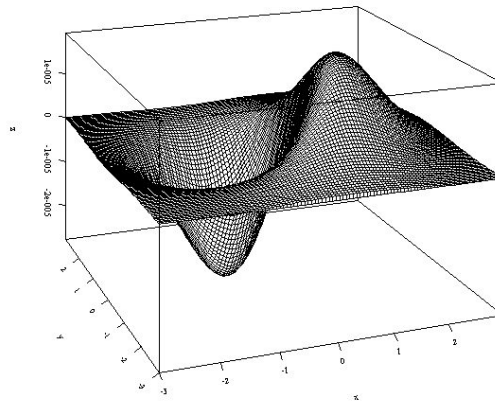
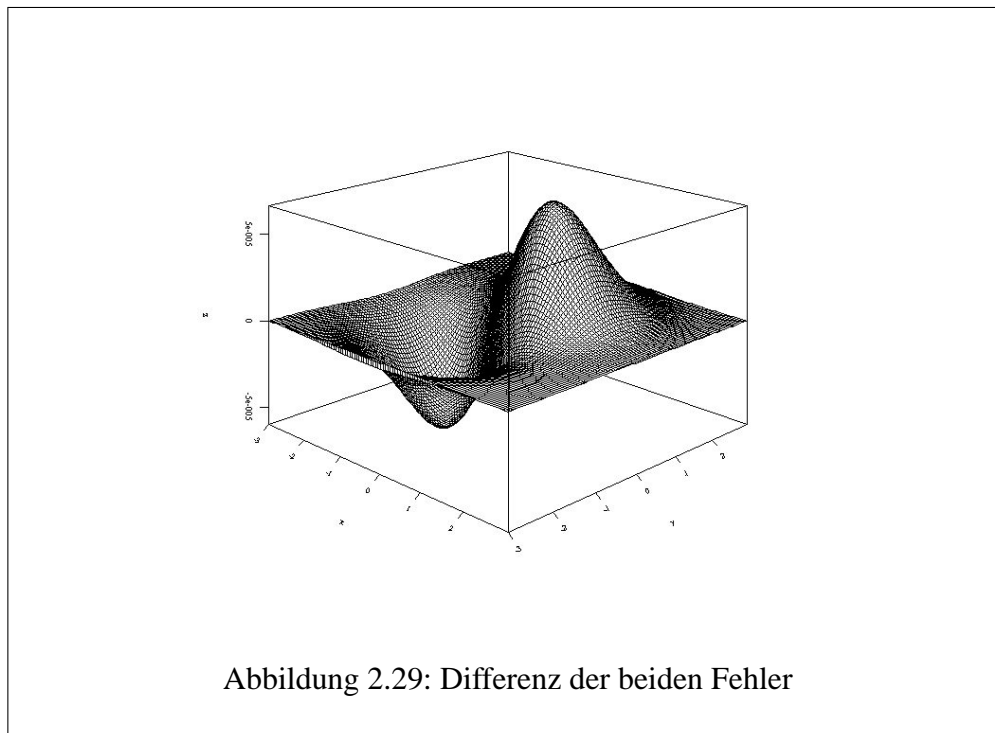


Abbildung 2.28: Differenz der beiden Kerndichten (Simulation – Henkel / Linde)

Schätzen dieser Verteilung ist zu komplex, um in der Praxis zur Anwendung zu kommen. Folglich muss ein anderer Ansatz gesucht werden.

Bei der Anwendung des EM-Schätzer müsste dessen Konvergenzverhalten⁸⁴ mit anderen Maximum-Likelihood-Schätzern verglichen werden. Dieser kann nicht in geschlossener Form angegeben werden, d. h. die Berechnungen gestalten sich komplex. Außerdem ist man wieder an die vorgegebenen Randverteilungen gebunden. Damit verschiedene Randverteilungen einer multivariaten Verteilung berücksichtigt werden, empfiehlt es sich die multivariate Modellierung von Verteilungen über Copulas vorzunehmen. In den nächsten Abschnitten werden Copulas im Anwendungsgebiet der Finanzmathematik vorgestellt. Es werden Schätzer für die Copulas angegeben und Tests für die Bestimmung der richtigen Copula.

⁸⁴z.B bei kleinen Stichprobenumfängen.



2.4.7 Copulas

2.4.7.1 Copulas in der Finanzmathematik

Copulas sind ein relativ neues Instrument zur Risikoberechnung im Finanzsektor. Meist soll das Risiko eines Portfolios berechnet und minimiert werden. Bei der Risikominimierung können zum einen Aktienverhältnisse⁸⁵ geändert werden oder es wird eine Aktie durch eine andere ersetzt. In diesem Fall bieten Copulas einen erheblichen Vorteil. Durch die Aufspaltung der Risikoberechnung in zwei Schritte können meist die Ergebnisse aus einem Schritt wiederverwendet werden. Diese Schritte sehen wie folgt aus:

1. Bestimmung der Parameter der Randverteilungen⁸⁶
2. Bestimmung der Abhängigkeit der Aktien⁸⁷ untereinander über Copulas.

⁸⁵Anzahl der Aktienanteile im Portfolio

⁸⁶Jede Aktie wird einzeln betrachtet.

⁸⁷Abhängigkeit der beiden Randverteilungen

Beim Austausch einer Aktie muss die Verteilung der neuen Aktie und deren Abhängigkeit über die Copula berechnet werden. Die Randverteilungen⁸⁸ können dabei beliebig sein. Es werden keine Randverteilungen durch Verwendung des Copula-Modells fest vorgegeben. Durch den Einsatz von Copulas kann das Modell sehr flexibel gestaltet werden, d. h. die Randverteilungen und Abhängigkeitsstruktur können beliebig gewählt werden. In multivariaten Modellen ohne Copulas⁸⁹ ist man an eine feste Randverteilung gebunden. In den nächsten Abschnitten werden Copulas definiert und deren Eigenschaften vorgestellt.

2.4.7.2 Einführung in Copulas

Die Copula C ist eine multivariate Verteilungsfunktion mit $C : [0, 1]^n \rightarrow [0, 1]$, wobei $n \geq 2, n \in \mathbb{N}$ ist. Wenn man bei einer multivariaten Verteilung nur die Randverteilungen kennt, kann man diese durch die Verteilungsfunktion der Randverteilung auf den Wertebereich der Copula transformieren. Werden diese dann in die Copula eingesetzt, erhält man eine multivariate Verteilung mit einer bestimmten Abhängigkeit. Diese Abhängigkeit wird durch die Copula spezifiziert. Mittels des Theorems von Sklar wird gezeigt, dass zu jeder multivariaten Verteilungsfunktion eine Copula existiert, so dass die multivariate Verteilungsfunktion sich aus den Randverteilungen und der Copula zusammensetzt. Es wird jetzt erstmal der zweidimensionale Fall betrachtet.

Definition 24 (zweidimensionale Sub-Copula, Nelsen [65]). *Eine zweidimensionale Sub-Copula ist eine Funktion C' mit folgenden Eigenschaften:*

1. *Der Grundraum von C' ist $S_1 \times S_2$ wobei S_1 und S_2 Teilmengen von $\mathbf{I} = [0, 1]$ sind.*
2. *C' ist beschränkt und zweifach wachsend⁹⁰.*
3. *Für jedes $u \in S_1$ und $v \in S_2$ gelte $C'(u, 1) = u$ und $C'(1, v) = v$.*

Definition 25 (H-Volumen, Nelsen[65]). *Seien S_1 und S_2 nicht offene Teilmengen von $\bar{\mathbb{R}}$, und sei H eine Funktion auf dem Grundraum $S_1 \times S_2$. Es sei $B =$*

⁸⁸Verteilungen der Aktien

⁸⁹z. B. multivariate Studentsche Verteilung

⁹⁰siehe Definition 26

$[x_1, x_2] \times [y_1, y_2]$ eine Rechteckmenge, wobei alle Eckpunkte im Grundraum von H liegen. Dann ist das H -Volumen gegeben durch

$$V_H(B) = H(x_2, y_2) - H(x_2, y_1) - H(x_1, y_2) + H(x_1, y_1). \quad (2.143)$$

Definition 26 (zwei-fach wachsend, Nelsen [65]). Eine zweifach reellwertige Funktion H ist zweifach wachsend, wenn $V_H(B) \geq 0$ für alle Rechteckmengen B , für die die Eckpunkte im Grundraum von H liegen.

Definition 27 (zweidimensionale Copula, Nelsen [65]). Eine zweidimensionale Copula C ist eine zweidimensionale Sub-Copula mit Grundraum \mathbf{I}^2 und Bildraum \mathbf{I} , für die folgende Eigenschaften gelten ($C : \mathbf{I}^2 \rightarrow \mathbf{I}$):

1. Für alle $u, v \in \mathbf{I}$ gilt $C(u, 0) = 0 = C(0, v)$.
2. Für alle $u_1, u_2, v_1, v_2 \in \mathbf{I}$ mit $u_1 \leq u_2$ und $v_1 \leq v_2$ gilt $C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0$.

Definition 28 (Survival Copula \bar{C} , Nelsen[65]). Sei H eine bivariate Verteilung mit den Randverteilungen X, Y ⁹¹, dann ist die Survivorfunktion von H gegeben durch $\bar{H}(x, y) = P[X > x, Y > y]$. Die Survivorfunktion \bar{H} kann mit der Copula C bzgl. H so ausgedrückt werden

$$\begin{aligned} \bar{H}(x, y) &= 1 - F(x) - G(y) + H(x, y) \\ &= \bar{F}(x) + \bar{G}(y) - 1 + C(F(x), G(y)) \\ &= \bar{F}(x) + \bar{G}(y) - 1 + C(1 - \bar{F}(x), 1 - \bar{G}(y)). \end{aligned}$$

Dann wird die Survival-Copula definiert durch

$$\bar{C}(u, v) = u + v - 1 + C(1 - u, 1 - v). \quad (2.144)$$

Es gilt somit

$$\bar{H}(x, y) = \bar{C}(\bar{F}(x), \bar{G}(y)).$$

Satz 6 (Sklar's Theorem, Nelsen [65]). Sei H eine bivariate Verteilungsfunktion mit den Randverteilungen F und G . Dann existiert eine Copula C , so dass für alle $x, y \in \mathbb{R}$ gilt,

$$H(x, y) = C(F(x), G(y)). \quad (2.145)$$

Wenn F und G stetig sind, dann ist die Copula C eindeutig.

⁹¹ X habe die Verteilungsfunktion F , Y die Verteilungsfunktion G .

Lemma 10 (Umkehrung Sklar's Theorem, Nelsen [65]). Sei H eine gemeinsame Verteilung mit den Randverteilungen F und G und seien F^{-1} und G^{-1} deren Quantilfunktionen und C' eine passende Sub-Copula und $u, v \in \mathbf{I}$, dann gilt

$$C'(u, v) = H(F^{-1}(u), G^{-1}(v)). \quad (2.146)$$

Wenn F und G stetig sind, dann gilt dieses Resultat auch für Copulas. Dies stellt daher eine Methode zur Erzeugung von Copulas dar.

Copulas wurden zunächst zweidimensional eingeführt. Sie auf den multivariaten Fall zu erweitern, gestaltet sich relativ einfach (siehe Nelsen [65]). Für die Betrachtung von Abhängigkeitsstrukturen reicht zunächst der zweidimensionale Fall aus. Die meisten Lemmata und Sätze werden aber, wenn sinnvoll, schon multivariat für $n \geq 2$ angegeben.

Lemma 11 (Invarianz gegenüber monoton wachsenden Transformationen, Embrechts et. al.[24]). Seien X_1, \dots, X_n reellwertige Zufallsvariablen mit der Copula C und $x_i, i = 1, \dots, n$ Realisationen der X_i . Dann gilt, wenn $T_1(x_1), \dots, T_n(x_n)$ streng monoton wachsende reellwertige Funktionen sind, dass die Zufallsvariablen $Y_1 = T_1(X_1), \dots, Y_n = T_n(X_n)$ mit Realisationen $y_i = T_1(x_1), \dots, y_n = T_n(x_n)$ die selbe Copula C besitzen, wie die Zufallsvariablen X_i .

Beweis. Habe $(U_1, \dots, U_n)^T$ die Verteilungsfunktion C^{92} und u_1, \dots, u_n deren Realisationen mit $u_i \in [0, 1]$. Dann gilt

$$\begin{aligned} C(F_{T_1(X_1)}(x_1), \dots, F_{T_n(X_n)}(x_n)) &= P[u_1 \leq F_{T_1(X_1)}(x_1), \dots, u_n \leq F_{T_n(X_n)}(x_n)] \\ &= P[F_{T_1(X_1)}^{-1}(u_1) \leq x_1, \dots, F_{T_n(X_n)}^{-1}(u_n) \leq x_n] \\ &= P[T_1 \circ F_{X_1}^{-1}(u_1) \leq x_1, \dots, T_n \circ F_{X_n}^{-1}(u_n) \leq x_n] \\ &= P[T_1(X_1) \leq x_1, \dots, T_n(X_n) \leq x_n]. \end{aligned}$$

□

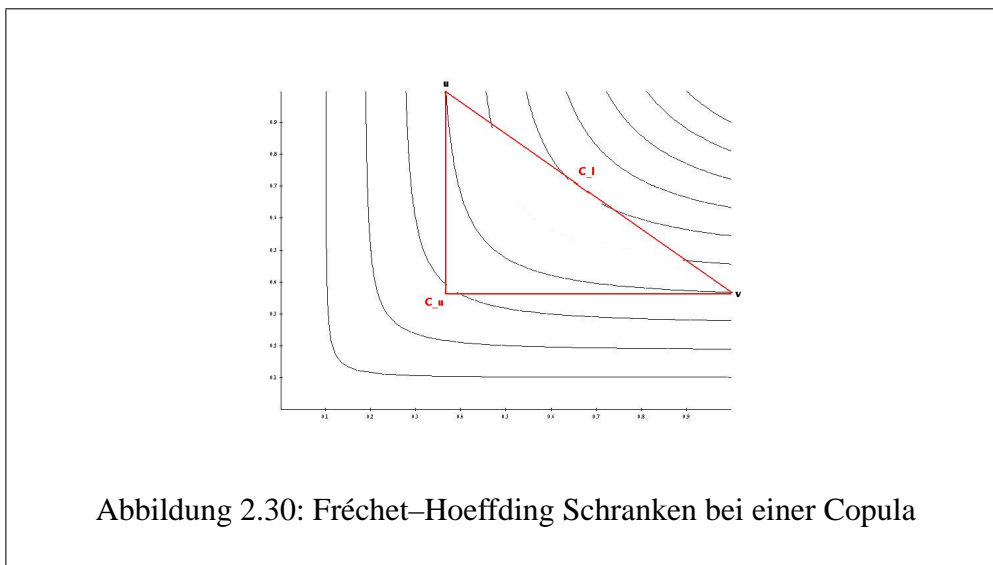
⁹²Falls F_{X_i} stetig ist, setze man $U_i = F_{X_i}(X_i)$.

Lemma 12 (Fréchet–Hoeffding Schranken, Nelsen [65]). Jede Copula C kann durch die Fréchet–Hoeffding–Schranken eingegrenzt werden. Sei

$C_l(x_1, \dots, x_n) = \max\{x_1 + \dots + x_n - n + 1, 0\}$ die untere Schranke und $C_u(x_1, \dots, x_n) = \min\{x_1, \dots, x_n\}$ die obere Schranke, dann gilt

$$C_l(x_1, \dots, x_n) \leq C(x_1, \dots, x_n) \leq C_u(x_1, \dots, x_n). \quad (2.147)$$

Für den Fall $n = 2$ sind C_l und C_u Copulas. Durch diese Schranken kann die Form der Copula festgelegt werden.



Definition 29 (Nelsen [65]). Eine Copula C_1 ist kleiner als eine Copula C_2 (oder C_2 ist größer als C_1). Dies wird bezeichnet mit

$$C_1 \prec C_2 \text{ (oder } C_2 \succ C_1), \quad (2.148)$$

wenn

$$\forall (u_1, \dots, u_N) \in \mathbf{I}^N, C_1(u_1, \dots, u_N) \leq C_2(u_1, \dots, u_N) \quad (2.149)$$

oder $C_1(u_1, \dots, u_N) \geq C_2(u_1, \dots, u_N)$.

Lemma 13 (Zusammenhang Copula und Copuladichte, Nelsen[65]). Jede Copula kann man schreiben als

$$C(u, v) = A_C(u, v) + S_C(u, v), \quad (2.150)$$

wobei

$$A_C(u, v) = \int_0^u \int_0^v \frac{\partial^2}{\partial s \partial t} C(s, t) dt ds \quad (2.151)$$

und

$$S_C(u, v) = C(u, v) - A_C(u, v) \quad (2.152)$$

ist.

Falls $S_C(u, v) = 0$ und $C(u, v) \equiv A_C(u, v)$ ist, dann besitzt die Copula eine Copuladichte und C ist absolut stetig. Auf die Copuladichte wird in einem folgenden Abschnitt noch weiter eingegangen. Wenn $S_C(u, v) \equiv C(u, v)$, d. h. $\frac{\partial^2 C(u, v)}{\partial u \partial v} = 0$, dann wird C als singularär bezeichnet. Über diese Darstellung können Schätzer für die Parameter einer Copula mittels der Maximum-Likelihood-Methode unter Verwendung der Copula-Dichte bestimmt werden.

2.4.8 Empirische Copula (Deheuvels Copula)

Definition 30 (Empirische Copula, Nelsen [65]). Sei $\{(x_i, y_i)\}_{i=1}^T$ mit $(x_i, y_i) \in [0, 1] \times [0, 1]$ eine bivariate Stichprobe vom Umfang T . Die empirische Copula \hat{C}_T ist gegeben durch

$$\hat{C}_T \left(\frac{i}{T}, \frac{j}{T} \right) = \frac{\text{Anzahl der Paare } (x, y) \text{ der Stichprobe mit } x \leq x_{(i)} \vee \leq y_{(j)}}{T}, \quad (2.153)$$

wobei $x_{(i)}$ und $y_{(j)}$ die Ordnungsstatistik⁹³ für $1 \leq i, j \leq T$ sind.

Es folgt eine weitere Definition der empirischen Copula, jetzt für den Fall $n \geq 2$.

Definition 31 (Empirische Copula, Bouye et. a. [11]). Sei $\mathcal{X} = \{(x_1^t, \dots, x_N^t)\}_{t=1}^T$ mit $(x_1^i, \dots, x_N^i) \in [0, 1]^N$ der Stichprobendatensatz der Copula. Dann wird mit $x_n^{(t)}$ die Ordnungsstatistik der Daten für $1 \leq t_1, \dots, t_N \leq T$ bezeichnet. Die empirische Copula ist gegeben durch

$$\hat{C} \left(\frac{t_1}{T}, \dots, \frac{t_n}{T}, \dots, \frac{t_N}{T} \right) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{[x_1^t \leq x_1^{(t_1)}, \dots, x_n^t \leq x_n^{(t_n)}, \dots, x_N^t \leq x_N^{(t_N)}]}. \quad (2.154)$$

Definition 32 (Empirische Copulahäufigkeit, Nelsen [65]). Die empirische Copulahäufigkeit \hat{c}_T ist gegeben durch

$$\hat{c}_T \left(\frac{i}{T}, \frac{j}{T} \right) = \begin{cases} 1/T, & \text{wenn } x_{(i)}, y_{(j)} \text{ Elemente der Stichprobe sind,} \\ 0, & \text{sonst.} \end{cases} \quad (2.155)$$

Falls doppelte Punkte auftreten ist $1/T$ durch $\frac{1}{T} \sum_{j=1}^n \mathbf{1}_{(x_{(i)}=x_j, y_{(i)}=y_j)}$ zu ersetzen, und der Stichprobenumfang sei n .

Es besteht folgender Zusammenhang zwischen \hat{C}_T und \hat{c}_T :

$$\hat{C}_T \left(\frac{i}{T}, \frac{j}{T} \right) = \sum_{p=1}^i \sum_{q=1}^j \hat{c}_n \left(\frac{p}{T}, \frac{q}{T} \right)$$

⁹³Diese Vorgehensweise vereinfacht die Berechnung der empirischen Copula, was besonders bei der Programmierung von Vorteil ist.

und

$$\begin{aligned}\hat{c}_n\left(\frac{i}{T}, \frac{j}{T}\right) &= \hat{C}_n\left(\frac{i}{T}, \frac{j}{T}\right) - \hat{C}_n\left(\frac{i-1}{T}, \frac{j}{T}\right) \\ &\quad - \hat{C}_n\left(\frac{i}{T}, \frac{j-1}{T}\right) + \hat{C}_n\left(\frac{i-1}{T}, \frac{j-1}{T}\right).\end{aligned}$$

Im nächsten Satz werden die empirischen Berechnungen von Spearman's ρ_S und Kendall's τ .

Satz 7 (Abhängigkeitsmaße für die Empirische Copula, Nelsen [65]). Sei \hat{C}_T die empirische Copula und \hat{c}_T die empirische Copulahäufigkeit für die Realisationen $\{(x_i, y_i)\}_{i=1}^T$. Die empirische Version von Spearman's ρ_S wird berechnet durch

$$\begin{aligned}\hat{\rho}_S &= \frac{12}{T^2 - 1} \\ &\quad \sum_{i=1}^T \sum_{j=1}^T \left[\hat{C}_T\left(\frac{i}{T}, \frac{j}{T}\right) - \frac{i}{T} \cdot \frac{j}{T} \right],\end{aligned}\tag{2.156}$$

Kendall's τ durch

$$\hat{\tau} = \frac{2n}{T-1} \sum_{i=2}^T \sum_{j=2}^T \sum_{p=1}^{i-1} \sum_{q=1}^{j-1} \left[\hat{c}_T\left(\frac{i}{T}, \frac{j}{T}\right) \hat{c}_T\left(\frac{p}{T}, \frac{q}{T}\right) - \hat{c}_T\left(\frac{i}{T}, \frac{q}{T}\right) \hat{c}_T\left(\frac{p}{T}, \frac{j}{T}\right) \right].\tag{2.157}$$

Beweis. Siehe Nelsen [65], Seite 178. □

Mit $[]$ wird die Gaussklammer bezeichnet.

Nun wird die empirische Tail-Dependence-Funktion vorgestellt.

Satz 8 (Empirische Tail-Dependence-Funktion, Durrleman, Nikeghbali und Roncalli [21]).

$$\hat{\chi}\left(\frac{t}{T}\right) = 2 - \frac{1 - \hat{C}_T\left(\frac{t}{T}, \frac{t}{T}\right)}{1 - \frac{t}{T}}\tag{2.158}$$

2.4.9 Copuladichte

Da eine Copula eine Verteilungsfunktion von $[0, 1]^n$, $n \geq 2$, $n \in \mathbb{N}$ auf $[0, 1]$ darstellt, soll jetzt durch Ableiten eine Copuladichte berechnet werden. Dabei muss berücksichtigt werden, dass die Copula C nicht singular ist, sondern $C(u, v) \equiv A_C(u, v)$ gilt. Mit I wird das Einheitsintervall $[0, 1]$ bezeichnet.

Satz 9 (Copuladichte, Patton [67]).

Sei C eine Copula mit $H(x, y) = C(F(x), G(y))$, $x, y \in \bar{\mathbb{R}}$. Außerdem seien X, Y zwei Zufallsvariablen für deren gemeinsame Verteilung gilt: $H = C(F, G)$. H sei die gemeinsame Verteilungsfunktion mit den Randverteilungen F und G , für die gilt $U = F(X)$ und $V = G(Y)$. Sind F und G stetige und streng monoton wachsende Funktionen, dann gilt nach dem Lemma von Sklar: Es gibt eine Copula C , für die gilt, $C(u, v) = H(F^{-1}(u), G^{-1}(v))$, $u, v \in \mathbf{I}$, wobei F^{-1} und G^{-1} die Quantilfunktionen (Inversen) von F und G sind. Es gilt $x = F^{-1}(u)$ und $y = G^{-1}(v)$, sowie $\frac{\partial F^{-1}(u)}{\partial u} = \frac{1}{f(F^{-1}(u))}$ und $\frac{\partial G^{-1}(v)}{\partial v} = \frac{1}{g(G^{-1}(v))}$. Dann gilt für die Ableitung von $C(u, v) = H(F^{-1}(u), G^{-1}(v))$

$$c(u, v) = \frac{h(F^{-1}(u), G^{-1}(v))}{f(F^{-1}(u))g(G^{-1}(v))}. \quad (2.159)$$

Beweis. Mit den obigen Annahmen folgt mit der Ableitung nach beiden Variablen:

$$\begin{aligned} \frac{\partial C(u, v)}{\partial u \partial v} &= \frac{\partial H(F^{-1}(u), G^{-1}(v))}{\partial u \partial v} \\ &= h(F^{-1}(u), G^{-1}(v)) \left| \begin{array}{cc} \frac{\partial F^{-1}(u)}{\partial G^{-1}(v)} & \frac{\partial F^{-1}(u)}{\partial v} \\ \frac{\partial u}{\partial G^{-1}(v)} & \frac{\partial v}{\partial v} \end{array} \right| \\ &= h(F^{-1}(u), G^{-1}(v)) \frac{\partial F^{-1}(u)}{\partial u} \frac{\partial G^{-1}(v)}{\partial v} \\ &= \frac{h(F^{-1}(u), G^{-1}(v))}{f(F^{-1}(u))g(G^{-1}(v))} \\ &= c(u, v). \end{aligned}$$

□

Lemma 14 (Eigenschaften der Copuladichte, Bouye et. al. [11]). Sei c die Dichte der Copula C , dann ist sie für den multivariaten Fall definiert durch

$$c(u_1, \dots, u_n) = \frac{\partial C(u_1, \dots, u_n)}{\partial u_1 \cdots \partial u_n}. \quad (2.160)$$

Sei nun \mathbf{F} die multivariate Verteilungsfunktion mit Dichte \mathbf{f} zur Copula C mit den Randverteilungen F_i und deren Dichten f_i mit $i = 1, \dots, n$. Dann gilt folgender Zusammenhang

$$\mathbf{f}(x_1, \dots, x_n) = c(F_1(x_1), \dots, F_n(x_n)) \prod_{i=1}^n f_i(x_i). \quad (2.161)$$

2.4.10 Empirische Copulahäufigkeit

In diesem Abschnitt soll eine allgemeine Form der Dichte einer empirischen Copula angegeben werden, in Definition 32 wurde schon eine Definition für den zweidimensionalen Fall geliefert. Leicht kann die Copuladichte auf den allgemeinen Fall $N \in \mathbb{N}, N \geq 2$ erweitert werden.

Definition 33 (Empirische Copulahäufigkeit bzw. Empirische Copuladichte, Bouye et. al. [11]). Die Dichte zu einer empirischen N – dimensionalen Copula \hat{C} ist gegeben durch

$$\hat{c} \left(\frac{t_1}{T}, \dots, \frac{t_n}{T}, \dots, \frac{t_N}{T} \right) = \sum_{i_1=1}^2 \dots \sum_{i_N=1}^2 (-1)^{i_1+\dots+i_N} \hat{C} \left(\frac{t_1 - i_1 + 1}{T}, \dots, \frac{t_N - i_N + 1}{T} \right), \quad (2.162)$$

wobei die gleichen Voraussetzungen wie in Definition 31 gelten sollen.

2.4.11 Schätzer für Copulas

In diesem Abschnitt sollen die Parameter der einzelnen Copulas⁹⁴, die bisher vorgestellt wurden, geschätzt werden. Da die Copula eine Verteilungsfunktion ist, soll, wenn diese eine Dichte besitzt⁹⁵, der Schätzer mit der Maximum–Likelihood–Methode bestimmt werden. Desweiteren werden noch ausgewählte andere Schätzer vorgestellt.

2.4.11.1 Maximum–Likelihood–Methode

Sei S der Grundraum und \mathcal{B} eine σ -Algebra. (S, \mathcal{B}) sei ein meßbarer Raum. Q_ϑ sei ein Familie von Wahrscheinlichkeitsmaßen auf \mathcal{B} mit $\vartheta \in \Theta$. Θ ist der Parameterraum.

Definition 34 (Likelihood-Funktion und Maximum-Likelihood-Schätzer).

Q_ϑ habe die μ -Dichte f_ϑ für $\vartheta \in \Theta$, wobei $\mu|_{\mathcal{B}}$ ein Maß ist:

- $L(\vartheta|\underline{x}) = \prod_{i=1}^n f_\vartheta(x_i)$ heißt Likelihood-Funktion zum Stichprobenumfang n ,

⁹⁴Die Definition der Copulas sind dem Anhang B.1.3 zu entnehmen.

⁹⁵Die Copula darf nicht singular sein.

- $\hat{\vartheta} : S^n \rightarrow \Theta$ heißt *Maximum-Likelihood-Schätzer zum Stichprobenumfang n falls*

$$\forall \underline{x} \in S^n : L(\hat{\vartheta}_n(\underline{x})|\underline{x}) = \max_{\vartheta \in \Theta} L(\vartheta|\underline{x})$$

Kann der Maximum-Likelihoodschätzer nicht über Ableiten der Likelihood-Funktion bestimmt werden, müssen andere numerische Verfahren zur Bestimmung des Maximums herangezogen werden.

Lemma 15 (MLE der Gauss–Copula). *Der Maximum–Likelihood–Schätzer für die Gauss–Copula ist gegeben durch*

$$\hat{\rho}_{ML} = \frac{1}{T} \sum_{t=1}^T \zeta_t T \zeta_t \quad (2.163)$$

mit $\zeta_t = (\Phi^{-1}(u_1^t), \dots, \Phi^{-1}(u_N^t))$ und Stichprobenumfang T .

Beweis. Die log–Likelihood–Funktion hat die Form

$$\ln(L)(\rho) = -\frac{T}{2} \ln |\rho| - \frac{1}{2} \sum_{t=1}^T \zeta_t T (\Sigma^{-1} - \mathcal{I}) \zeta_t,$$

\mathcal{I} bezeichnet die Einheitsmatrix. Der Beweis kann Magnus und Neudecker [56] entnommen werden. \square

Lemma 16 (MLE der Clayton–Copula). *Der Maximum–Likelihood–Schätzer für δ der Clayton–Copula ($n = 2$) ist ein Lösung von*

$$\begin{aligned} 0 = & \frac{n}{1 + \delta} \\ & + \sum_{i=1}^n \left(\frac{\ln \left(\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1 \right)}{\delta^2} - \frac{-\frac{\ln(u_i)}{u_i^\delta} - \frac{\ln(v_i)}{v_i^\delta}}{d \left(\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1 \right)} \right) \\ & + \sum_{i=1}^n \left(-2 \frac{-\frac{\ln(u_i)}{u_i^\delta} - \frac{\ln(v_i)}{v_i^\delta}}{\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1} - \ln(u_i) - \ln(v_i) \right). \end{aligned} \quad (2.164)$$

δ muss mit numerischen Verfahren berechnet werden, was bei großen Stichproben aufwendig sein kann.

Beweis. Die Likelihood-Funktion mit der Copuladichte (siehe Abschnitt B.1.3, S. 194) sieht folgendermaßen aus:

$$L(\delta; (u_1, v_1), \dots, (u_n, v_n)) = \prod_{i=1}^n (u_i^{-\delta} + v_i^{-\delta} - 1)^{\left(-\frac{1+2\delta}{\delta}\right)} (u_i v_i)^{-1-\delta} (1 - \delta).$$

Daraus ergibt sich die log-Likelihood-Funktion

$$\begin{aligned} \ln L(\delta; (u_i, v_i)) &= n \ln(1 - \delta) \\ &+ \sum_{i=1}^n \left(-\frac{\ln\left(\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1\right)}{\delta} - 2 \ln\left(\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1\right) \right) \\ &- \sum_{i=1}^n ((1 + \delta)(\ln(u_i) + \ln(v_i))). \end{aligned}$$

Ableiten nach δ liefert den ML-Schätzer für die Clayton-Copula

$$\begin{aligned} \frac{\partial \ln L(\delta; (u_i, v_i))}{\partial \delta} &= \frac{n}{1 - \delta} \\ &+ \sum_{i=1}^n \left(\frac{\ln\left(\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1\right)}{\delta^2} - \frac{-\frac{\ln(u_i)}{u_i^\delta} - \frac{\ln(v_i)}{v_i^\delta}}{d\left(\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1\right)} \right) \\ &+ \sum_{i=1}^n \left(-2 \frac{-\frac{\ln(u_i)}{u_i^\delta} - \frac{\ln(v_i)}{v_i^\delta}}{\frac{1}{u_i^\delta} + \frac{1}{v_i^\delta} - 1} - \ln(u_i) - \ln(v_i) \right). \end{aligned}$$

□

Lemma 17 (MLE der Student-Copula, Bouye et. al. [11]). *Der Maximum-Likelihood-Schätzer kann nicht in geschlossener Form, sondern nur durch die log-Likelihood-Funktion angegeben werden:*

$$\begin{aligned} \log L(\rho, \nu) &\propto -\frac{T}{2} \ln |\rho| - \left(\frac{\nu + N}{2}\right) \sum_{t=1}^T \ln \left(1 + \frac{1}{\nu} \zeta_t^t \rho^{-1} \zeta_t\right) \\ &+ \frac{\nu + 1}{2} \sum_{t=1}^T \sum_{n=1}^N \ln \left(1 + \frac{\zeta_n^2}{\nu}\right), \end{aligned} \tag{2.165}$$

mit $\zeta_t = (\Phi^{-1}(u_1^t), \dots, \Phi^{-1}(u_N^t))$ und Stichprobenumfang T . Für große N gestaltet sich die numerische Berechnung der Parameter sehr komplex. Um die Berechnung zu vereinfachen, werden noch andere Schätzmethoden⁹⁶ angegeben.

⁹⁶siehe IFM und CML

Beweis. Der Ausgangspunkt des Beweises ist die Dichte der Student–Copula

$$\mathbf{c}(u_1, \dots, u_N; \rho, \nu) = |\rho|^{-\frac{1}{2}} \frac{\Gamma\left(\frac{\nu+N}{2}\right) [\Gamma\left(\frac{\nu}{2}\right)]^N}{[\Gamma\left(\frac{\nu+1}{2}\right)]^N \Gamma\left(\frac{\nu}{2}\right)} \frac{(1 + \frac{1}{\nu} \zeta^T \rho^{-1} \zeta)^{-\frac{\nu+N}{2}}}{\prod_{n=1}^N \left(1 + \frac{\zeta_n^2}{\nu}\right)^{-\frac{\nu+1}{2}}}.$$

Mit Hilfe von ihr wird dann der Maximum–Likelihood–Schätzer berechnet. \square

Es wurden bisher nur Schätzer für die Copulas betrachtet. Die Randverteilungen wurden hier noch nicht berücksichtigt. Bei den nächsten Schätzern werden die Randverteilungen in die Schätzung mit einbezogen.

2.4.12 Weitere Schätzer

2.4.12.1 IFM–Methode

Bei der IFM⁹⁷ Methode werden die Parameter der Randverteilungen unabhängig von denen der Copula geschätzt, d. h. zunächst werden die Parameter der Randverteilung geschätzt und dann in die Copula übernommen, womit die restlichen Parameter der Copula geschätzt werden. Der Algorithmus (siehe Romano [77]) sieht wie folgt aus:

1. Schätzen der Parameter ϑ_i , $i = 1, \dots, n$, der Randverteilungen F_i mit dem Maximum–Likelihood–Schätzer:

$$\hat{\vartheta}_i = \arg \max \sum_{t=1}^T \ln f_i(x_i^t; \vartheta_i),$$

dabei sei f_i die Dichte von F_i sein.

2. Schätzen des Parameters α der Copula mit den vorhandenen geschätzten Parametern ϑ aus Schritt 1:

$$\hat{\alpha} = \arg \max l^c(\alpha) = \arg \max \sum_{t=1}^T \ln c\left(F_1(x_1^t; \hat{\vartheta}_1), \dots, F_n(x_n^t; \hat{\vartheta}_n); \alpha\right),$$

wobei l^c der Maximum–Likelihood–Schätzer der Copula ist.

⁹⁷Method of Inference Functions of Margins

2.4.12.2 CML–Methode

Die CML⁹⁸ Methode (siehe Romano [77]) unterscheidet sich von der IFM–Methode dadurch, dass in diesem Fall nicht die Parameter der Randverteilung geschätzt werden. Die Randverteilungen werden mit der empirischen Verteilungsfunktion in das Einheitsintervall transformiert. Damit können die Parameter der Copula dann geschätzt werden.

1. Transformieren des Datensatzes (x_1^t, \dots, x_n^t) , $t = 1, \dots, T$ mit der empirischen Verteilungsfunktion \hat{F} . Diese werden mit $(\hat{u}_1^t, \dots, \hat{u}_n^t)$ bezeichnet und die u_i liegen im Intervall $[0, 1]$.
2. Schätzen der Parameter der Copula mit

$$\hat{\alpha} = \arg \max \sum_{t=1}^T \ln c(\hat{u}_1^t, \dots, \hat{u}_n^t; \alpha). \quad (2.166)$$

2.4.12.3 Beispiele für die IFM– und CML–Methode

Für die Gauss–Copula kann der Schätzer des Parameters ρ mit der IFM– und CML–Methode wie folgt berechnet werden:

$$\hat{\rho}_{IFM/CML} = \frac{1}{T} \sum_{t=1}^T \zeta_t^T \zeta_t, \quad (2.167)$$

wobei $\zeta_t = (\Phi^{-1}(u_1^t), \dots, \Phi^{-1}(u_n^t))$. Es gilt für die CML–Methode, dass die u_i^t mit der empirischen Verteilungsfunktion berechnet werden. In der IFM–Methode werden die x_i mit der Verteilungsfunktion⁹⁹ der Normalverteilung in die $u_i \in [0, 1]$ transformiert.

Für den Schätzer des Parameters ρ der Student–Copula kann folgende Rekursion zur Berechnung angegeben werden:

1. Startschätzer ist der Schätzer der Gauss Copula für den Parameter ρ . α wird aus den Randverteilungen geschätzt.
2. $\hat{\rho}_{m+1} = \frac{1}{T} \left(\frac{\alpha+n}{\alpha} \right) \sum_{t=1}^T \frac{\zeta_t^T \zeta_t}{1 + \frac{1}{\alpha} \zeta_t^T \hat{\rho}_m^{-1} \zeta_t}$, $m = 1, 2, \dots$,
wobei $\zeta_t = (t_\alpha^{-1}(u_1^t), \dots, t_\alpha^{-1}(u_n^t))$.

⁹⁸Canonical Maximum Likelihood Method

⁹⁹Die Parameter der Verteilungsfunktion können mit der Maximum–Likelihood–Methode geschätzt werden.

3. Schritt 2 wird solange wiederholt, bis die Folge $\hat{\rho}_m$ konvergiert.

Marshall und Zeevi geben in ihrer Arbeit [61] einen anderen Algorithmus zur Schätzung des Parameters ρ der Student–Copula.

1. Transformiere die Daten (x_1^t, \dots, x_n^t) , $t = 1, \dots, T$ auf das Einheitsintervall $[0, 1]$ mit den empirischen Verteilungsfunktionen der Randverteilungen. Die Daten werden dann mit $(\hat{u}_1^t, \dots, \hat{u}_n^t)$ bezeichnet.
2. Schätze \hat{R} mit dem Schätzer für Kendall's τ :

$$\hat{R} = \sin\left(\frac{\pi}{2} \hat{\tau}_{i,j}\right), \quad i, j = 1, \dots, n.$$

3. $\hat{\alpha}$ wird durch numerisches Lösen der Maximum–Likelihood Gleichung bestimmt:

$$\hat{\alpha} = \arg \max_{\alpha \in (2, \infty]} \left[\sum_{t=1}^T \log \left(c(u_1^t, \dots, u_n^t; \alpha, \hat{R}) \right) \right], \quad (2.168)$$

wobei

$$c(u_1, \dots, u_n; \alpha, \hat{R}) = \frac{\Gamma((\alpha + n)/2) \Gamma(\alpha/2)^{n-1} (1 + \mathbf{y}^T \hat{R}^{-1} \mathbf{y})^{-(\alpha+n)/2}}{\sqrt{\hat{R}} \Gamma((\alpha + 1)/2)^n \prod_{i=1}^n (1 + y_i^2/\alpha)^{-(\alpha+1)/2}} \quad (2.169)$$

und

$$\mathbf{y} = (y_1, \dots, y_n) = (t_\alpha^{-1}(u_1), \dots, t_\alpha^{-1}(u_n)). \quad (2.170)$$

2.4.13 Erzeugen von Realisationen einer Copula

In diesem Abschnitt werden Verfahren vorgestellt, die Stichproben für eine Copula erzeugen. Im ersten Abschnitt wird ein allgemeines Verfahren zur Erzeugung vorgestellt, später werden spezielle Simulationsverfahren beschrieben.

2.4.13.1 Allgemeine Vorgehensweise

Sollen zufällige Daten simuliert werden denen eine spezielle Copula zu Grunde liegt, so geschieht dies nach dem folgendem Algorithmus, Bouye et. al. [11]:

1. Erzeuge N unabhängig auf $[0, 1]$ gleichverteilte Stichproben v_1, \dots, v_N .

2. Setze $u_1 = v_1$.

3. Sei $C(u_m; u_1, \dots, u_{m-1}) = C_{m|1, \dots, m-1}(u_1, \dots, u_m)$, $m = 2, \dots, n$, wobei

$$\begin{aligned} C_{m|1, \dots, m-1}(u_1, \dots, u_m) &= P\{U_m \leq u_m | (U_1, \dots, U_{m-1} = (u_1, \dots, u_{m-1}))\} \\ &= \frac{\partial^{m-1}_{(u_1, \dots, u_{m-1})} C(u_1, \dots, u_m, 1, \dots, 1)}{\partial^{m-1}_{(u_1, \dots, u_{m-1})} C(u_1, \dots, u_{m-1}, 1, \dots, 1)}. \end{aligned}$$

4. Setze $u_m = C^{-1}(v_m; u_1, \dots, u_{m-1})$, $m = 2, \dots, n$.

Die u_i , $i = 1, \dots, n$, sind erzeugt nach der Copula C . Dieses Verfahren wird jetzt für die Erzeugung einer Stichprobe der Frank–Copula, Romano [77], angewandt. Sei die Frank–Copula wie in Abschnitt 62, Seite 200. Dann sieht der Algorithmus folgendermaßen aus:

1. $v_1, v_2 \sim U[0, 1]$ iid. .

2. Setze $u_1 = v_1$.

3. Sei $C(u_2; u_1) = C_{2|1}(u_1, u_2)$. Setze $u_2 = C^{-1}(v_2; u_1)$.

4. Der Vektor (u_1, u_2) ist erzeugt worden nach der Copula C .

Für die bedingte Copula gilt

$$C_{2|1}(u_1, u_2) = \frac{(\exp(-\alpha u_1) - 1) \exp(-\alpha u_1)}{(\exp(-\alpha) - 1) + (\exp(-\alpha u_1) - 1) (\exp(-\alpha u_2) - 1)}.$$

und die Inverse Copula ist gegeben durch

$$\begin{aligned} C^{-1}(u; u_1) &= \{u_2 : C_{2|1}(u_1, u_2) = u\} \\ &= -\frac{1}{\alpha} \ln \left(1 + \frac{u (\exp(-\alpha) - 1)}{u + (1 - u) \exp(-\alpha u_2)} \right). \end{aligned}$$

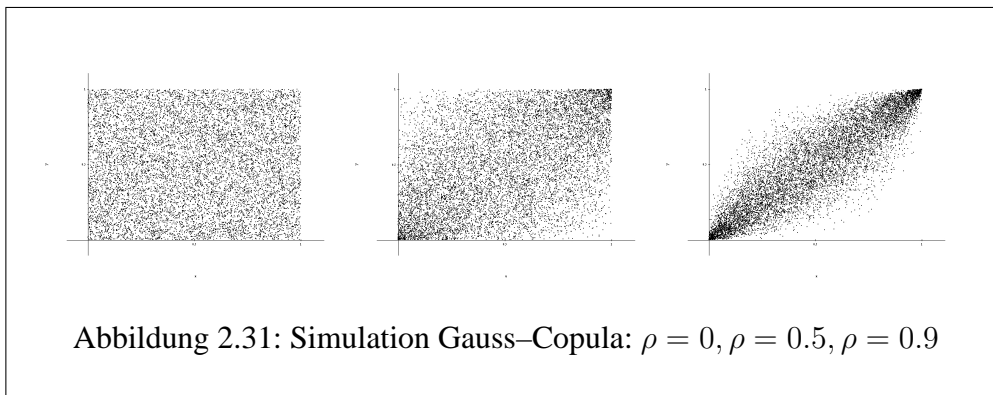
2.4.13.2 Spezielle Algorithmen zur Stichprobenerzeugung

1. **Simulation: Gauss–Copula**, Romano [77].

Sei R die Kovarianzmatrix der zu Grunde liegenden Verteilung.

- Berechnung der Cholesky–Zerlegung A für die Matrix R . Es gilt $AA^T = R$.

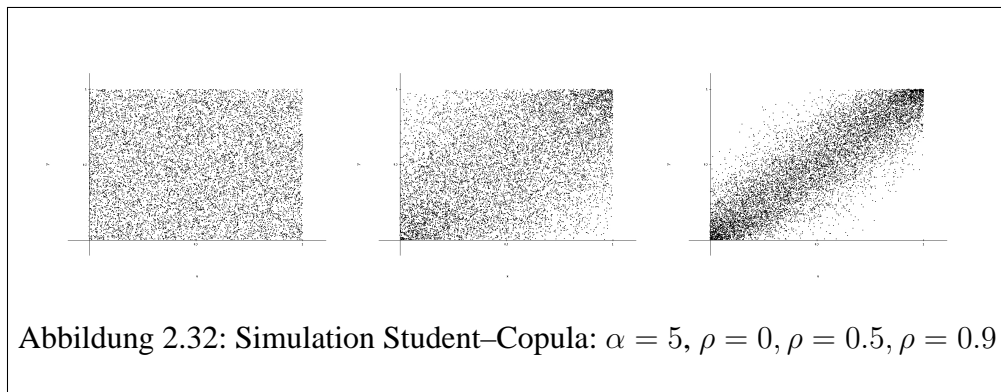
- Erzeugung von n unabhängigen standard normalverteilten Stichproben $z = (z_1, \dots, z_n)^T$.
- Berechne $x = Az$.
- Transformiere die x_i mit der Verteilungsfunktion der Standard–Normalverteilung auf $[0, 1]$, $u_i = \Phi(x_i)$, $i = 1, \dots, n$.
- Der Vektor (u_1, \dots, u_n) ist dann eine zufällige Stichprobe der Gauss–Copula mit Kovarianz–Matrix R .



2. Simulation: Student–Copula

Sei R wiederum die Kovarianz–Matrix und α der Gestaltsparameter. Dann wird die Stichprobe der Student–Copula wie folgt erzeugt:

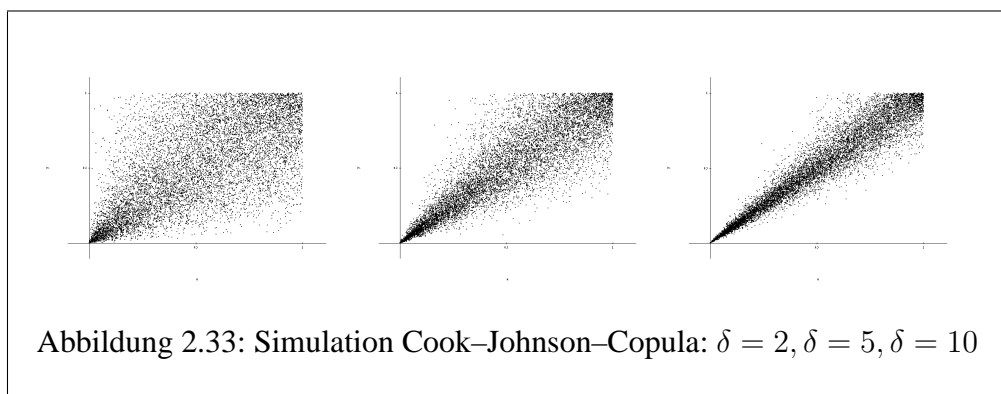
- Berechnung der Cholesky–Zerlegung A für die Matrix R .
Es gilt $AA^T = R$.
- Erzeugung von n unabhängigen standard normalverteilten Stichproben $z = (z_1, \dots, z_n)^T$.
- Berechne $y = Az$.
- Erzeuge eine gammaverteilte Zufallsvariable s unabhängig von z .
- Setze $x = \frac{\sqrt{\alpha}}{\sqrt{s}}y$.
- Transformiere die Komponenten auf $[0, 1]$ mit $u_i = t_\alpha(x_i)$, $i = 1, \dots, n$.
- (u_1, \dots, u_n) ist die gewünschte Stichprobe der Student–Copula.



3. Simulation: Cook-Johnson-Copula

Sei α der Parameter der Cook-Johnson-Copula. Der Algorithmus zur Erzeugung einer Stichprobe ist dann gegeben durch

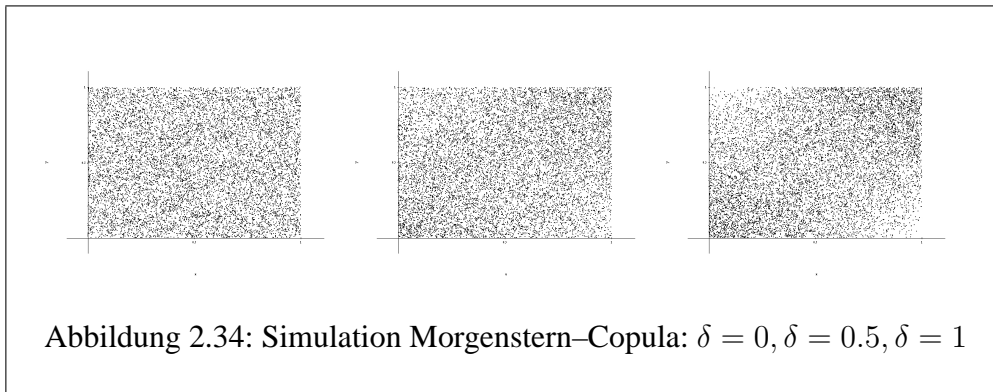
- Erzeuge n standard exponential verteilte Zufallsvariablen (y_1, \dots, y_n) .
- Erzeuge z gammaverteilt mit $\Gamma(1/\alpha, 1)$ unabhängig von den $y_i, i = 1, \dots, n$.
- Setze $u_j = (1 + y_j/z)^{-1/\alpha}, j = 1, \dots, n$.
- Die u_j sind nach der Cook-Johnson-Copula mit Parameter α verteilt.



4. Simulation: Farlie–Gumbel–Morgenstern–Copula

Sei α der Parameter der Farlie–Gumbel–Morgenstern Copula. Dann kann eine Stichprobe folgendermaßen erzeugt werden:

- Erzeuge $v_1, v_2 \sim U[0, 1]$ iid.
- Setze $u_1 = v_1$.
- Berechne
 $A = \alpha(2u_1 - 1) - 1$ und $B = [1 - \alpha(2u_1 - 1)]^2 + 4\alpha v_2(2u_1 - 1)$.
- Setze $u_2 = 2v_2 / (\sqrt{B} - A)$.
- u_1, u_2 ist durch die Farlie–Gumbel–Morgenstern–Copula erzeugt worden.



5. Simulation: Gumbel–Copula

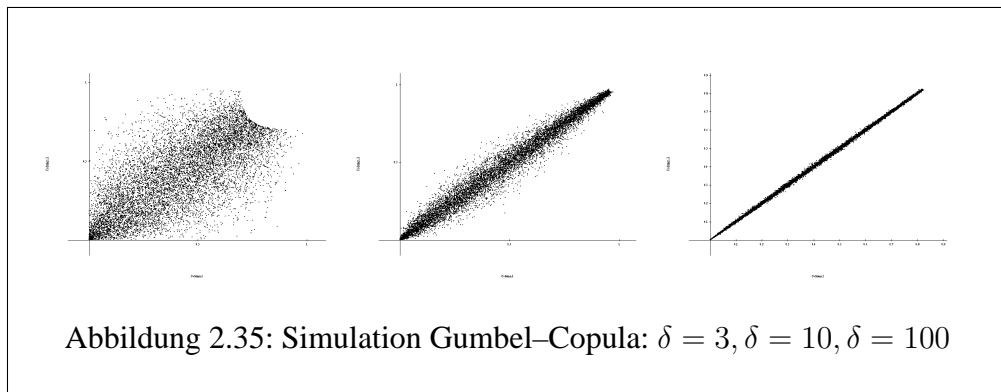
Sei $1 \leq \theta \leq \infty$, dann wird die Stichprobe nach dem folgendem Algorithmus¹⁰⁰ erzeugt:

- Erzeuge $s, q \sim U[0, 1]$ iid.
- Setze $t = K^{-1}(q)$ mit $K(t) = t - \frac{\phi(t)}{\phi'(t)}$, wobei $\phi(t)$ der Generator¹⁰¹ der Gumbel–Copula ist.
- Setze $u = \phi^{-1}(s\phi(t))$ und $v = \phi^{-1}((1 - s)\phi(t))$.

u, v sind dann erzeugt nach der Gumbel–Copula.

¹⁰⁰Entnommen aus Melchiori [63].

¹⁰¹Vgl. Abschnitt 2.4.14.3, S. 142



2.4.14 Auswahl einer Copula

In den letzten Abschnitten wurden Copulas, deren Eigenschaften und Schätzer vorgestellt. Wie schon erwähnt, kann die Schätzung der multivariaten Verteilung unter Anwendung von Copulas in zwei Schritte unterteilt werden:

1. Schätzung der Randverteilung
2. Schätzung der Copula

Im ersten Schritt werden die Randverteilungen geschätzt, d. h. man hat eine univariate Verteilung zu schätzen. Dies macht keine Probleme, da Schätzer für beliebige Verteilungen vorhanden sind. Zu berücksichtigen ist, dass auch überprüft werden muss, ob die Verteilungsannahme stimmt. Auch hierfür stehen viele Methoden zur Verfügung, wie zum Beispiel der Q-Q-Plot.

Der zweite Schritt gestaltet sich etwas komplexer. Die Schätzung der Parameter der Copula ist umso aufwendiger je höher die Dimension der Copula ist. Außerdem ist es auch hier wichtig, die richtige Copula auszuwählen¹⁰². Dieser Abschnitt setzt sich mit der Auswahl der richtigen Copula auseinander. Bei den Copulas gibt es zwei große Klassen, zum einem die elliptischen und zum anderen die archimedischen Copulas. Der erste Schritt ist es, eine dieser beiden Klassen auszuwählen. Dies wird in den Abschnitten 2.4.14.2 und 2.4.14.3 erörtert. Anschließend ist es wichtig, sich auf eine Familie von Copulas festzulegen. Dabei ist eine intuitive Vorauswahl¹⁰³ nötig, da sonst der Aufwand zur Auswahl zu groß wäre. Dies kann

¹⁰²Bei den Randverteilungen kann die empirische Verteilung zur Transformation benutzt werden.

¹⁰³Es gibt zu viele verschiedene Familien von Copulas.

zum Beispiel durch Visualisieren der Daten geschehen. Auswahlverfahren um die richtige Copula zu finden, die die zu Grunde liegende Abhängigkeit wieder gibt, werden in den nächsten Abschnitten vorgestellt.

2.4.14.1 Auswahl der elliptischen Copulas

Elliptische Copulas werden über elliptische Verteilungen hergeleitet¹⁰⁴. Sei X ein d -dimensionaler Zufallsvektor einer elliptischen Verteilung, dann kann X geschrieben werden als

$$X = \mu + RAU, \quad (2.171)$$

wobei $\mu \in \mathbb{R}^d$, A eine nicht singuläre $d \times d$ -Matrix, R ein nicht negativer Zufallsvektor und U gleichverteilt auf der Einheitssphäre $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$ ist. R und U sind unabhängig. Sei $\Sigma = AAT$ die Matrix der Gestaltsparameter und die Kovarianzmatrix Σ_0 von X sei proportional zu dieser. Seien X_1, X_2, \dots, X_n unabhängig identisch verteilte Stichproben der d -dimensionalen Verteilung. Die Null-Hypothese des Test ist, dass die Stichprobe von einer elliptischen Verteilung stammt. Sei \bar{X} der Stichprobenmittelwert und S die Stichprobenmatrix, dann seien die Residuen Y_k definiert als

$$Y_k = S^{-1/2}(X_k - \bar{X}), \quad k = 1, \dots, n. \quad (2.172)$$

Dann seien die W_k definiert durch

$$W_k = Y_k / \|Y_k\|, \quad k = 1, \dots, n, \quad (2.173)$$

welche Projektionen der Residuen Y_k auf die Einheitssphäre darstellen. Wenn X elliptisch verteilt ist, dann ist W annähernd gleichverteilt auf \mathcal{S}^{d-1} . Sei $\epsilon > 0$ fest und sei n_ϵ der ganzzahlige Teil von ϵn_ϵ , q_n ist das empirische ϵ -Quantil der Variablen $\|Y_1\|, \|Y_2\|, \dots, \|Y_n\|$. Das arithmetische Mittel ist

$$Q_n(h) = \frac{1}{n} \sum_{k=1}^n h(W_k) \mathbb{1}_{\{\|Y_k\| > q_n\}}, \quad (2.174)$$

wobei h eine Funktion, die definiert auf \mathcal{S}^{d-1} ist. Die Test-Statistik Z_n^2 ist dann gegeben durch

$$Z_n^2 = n \sum_{h \in \mathcal{H}} Q_n^2(h). \quad (2.175)$$

¹⁰⁴Die Ergebnisse aus diesem Abschnitt entstammen aus Breymann, Dias und Embrechts [10]. Der Test auf elliptische Symmetrie wurde von Manzotti et. al. [59] entwickelt.

Die Definition des Raumes \mathcal{H} kann Manzotti et. al. [59] entnommen werden.

Als nächstes wird ein spezieller Test für die Gauss-Copula vorgestellt.

2.4.14.2 Test auf Gauss-Copula

Die Gauss-Copula ist eine der meist verwendeten Copulas. Aber ist sie auch immer die Richtige? Ein Test¹⁰⁵ zu ihrer Überprüfung lautet wie folgt:

Die Nullhypothese lautet:

H_0 : Die Gauss-Copula beschreibt die Abhängigkeit. Die Test-Statistik wird beschrieben durch

$$z^2 = \sum_{i,j=1}^N \Phi^{-1}(F_i(x_i)) (\rho^{-1})_{ij} \Phi^{-1}(F_j(x_j)), \quad (2.176)$$

mit $\Sigma = Cov[\Phi^{-1}(F_i(x_i)), \Phi^{-1}(F_j(x_j))]$. z^2 ist χ^2 -verteilt mit N Freiheitsgraden. Sei X ein n -dimensionaler Zufallsvektor $X = (x_1, \dots, x_n)$. Dann sei F die gemeinsame Verteilungsfunktion und F_i die der Randverteilungen x_i . Es gilt für die gemeinsame Verteilungsfunktion

$$F(x_1, \dots, x_n) = \Phi_{\rho, N}(\Phi^{-1}(F_1(x_1)), \dots, \Phi^{-1}(F_n(x_n))), \quad (2.177)$$

mit $x_i \in [0, 1]$. Setze $y_i = \Phi^{-1}(F_i(x_i))$, wobei y_i normalverteilt ist. Dann hat z^2 die Form

$$z^2 = y^t \Sigma^{-1} y = \sum_{i,j=1}^N y_i \Sigma^{-1} y_j. \quad (2.178)$$

Wie schon erwähnt, ist z^2 annähernd χ^2 -verteilt.

Der Test hat folgenden Aufbau:

Es gelten folgende Bedingungen

- $N = 2$
- Die Stichprobendaten seien

$$\begin{aligned} &\{x_1(1), \dots, x_1(T)\}, \\ &\{x_2(1), \dots, x_2(T)\}, \end{aligned} \quad (2.179)$$

wobei T die Größe des Stichprobenumfangs beschreibt. Sie sollen unabhängig und identisch verteilt sein, mit Verteilungsfunktion F .

¹⁰⁵entnommen aus Malevergne und Sornette [57]

Die empirische Verteilungsfunktion \hat{F} ist somit ein Schätzer für F mit

$$\hat{F}_i(x) = \frac{1}{T} \sum_{k=1}^T 1_{\{x_i(k) \leq x\}}, \quad i = 1, 2. \quad (2.180)$$

Damit können die y_i geschätzt werden durch

$$\hat{y}_i = \Phi^{-1} \left(\hat{F}_i(x_i(k)) \right), \quad k \in \{1, \dots, T\}. \quad (2.181)$$

Die Stichprobenkovarianzmatrix $\hat{\rho}$ kann nun geschätzt werden mit

$$\hat{\rho} = \frac{1}{T} \sum_{i=1}^T \hat{y}(i) \hat{y}(i)^T, \quad (2.182)$$

und z^2 ist gegeben durch

$$\hat{z}^2(k) = \sum_{i,j=1}^2 \hat{y}(i) \Sigma^{-1} \hat{y}(i)^T. \quad (2.183)$$

Getestet wird nach folgendem Ablauf:

1. Gegeben seien die Stichprobendaten $x(t)$, $t \in \{1, \dots, T\}$, mit diesen Daten werden die normalverteilten $\hat{y}(t)$, $t \in \{1, \dots, T\}$ berechnet.
2. Damit wird dann die Kovarianzmatrix $\hat{\Sigma}$ geschätzt und \hat{z}^2 berechnet. Anschließend wird der Abstand zwischen der Verteilung von \hat{z}^2 und der χ^2 -Verteilung berechnet:

$$\begin{aligned} d_1 &= \max_z |F_{z^2}(z^2) - F_{\chi^2}(z^2)| \quad \text{Kolmogorovabstand,} \\ d_2 &= \int |F_{z^2}(z^2) - F_{\chi^2}(z^2)| dF_{\chi^2}(z^2) \quad \text{mittlerer Kolmogorovabstand,} \\ d_3 &= \max_z \frac{|F_{z^2}(z^2) - F_{\chi^2}(z^2)|}{\sqrt{F_{\chi^2}(z^2)[1 - F_{\chi^2}(z^2)]}} \quad \text{Anderson-Darling-Abstand,} \\ d_4 &= \int \frac{|F_{z^2}(z^2) - F_{\chi^2}(z^2)|}{\sqrt{F_{\chi^2}(z^2)[1 - F_{\chi^2}(z^2)]}} dF_{\chi^2}(z^2) \quad \text{mittlere Anderson-Darling-Abstand.} \end{aligned} \quad (2.184)$$

3. Mit der gegebenen Kovarianzmatrix $\hat{\Sigma}$ werden T normalverteilte Zufallsvektoren erzeugt.

4. Für diese Daten wird dann die Kovarianzmatrix $\tilde{\Sigma}$ berechnet.
5. Außerdem wird für jeden Vektor die Zufallsvariable z^2 berechnet. Sie wird mit $\tilde{z}^2(t)$ bezeichnet.
6. Danach wird die Verteilung von \tilde{z}^2 und der Abstand zur χ^2 -Verteilung berechnet.
7. Die Schritte (3)–(6) werden 10000 mal wiederholt, um die Genauigkeit der Schätzung des Abstandes zu verbessern. Dies liefert die Teststatistik, welche die Möglichkeit bietet die Nullhypothese H_0 zu einem bestimmten Signifikanzlevel anzunehmen oder zu verwerfen.

Malevergne und Sornette [57] wenden diesen Test auf verschiedene Finanzwerte an und kommen zu dem Schluss, dass die Gausscopula die Abhängigkeit in den meisten Fällen zutreffend beschreibt. Sie weisen aber darauf hin, dass bei der Auswahl der Copula immer ihr Anwendungsbereich berücksichtigt werden muss. Im Stress-Testing kann die Gausscopula an manchen Stellen nicht sinnvoll sein. Außerdem wird in ihrer Arbeit mit diesem Test die Gauss- und die Student-Copula verglichen. Für kleine Freiheitsgrade α der Studentschen Verteilung kann der Test eine klare Aussage treffen. Da die Gauss-Copula ein Spezialfall der Student-Copula für $\alpha \rightarrow \infty$ ist, kann bei großen α 's keine klare Entscheidung für die Gauss- oder Student-Copula getroffen werden.

2.4.14.3 Auswahl der Archimedischen Copulas

Die Archimedischen Copulas können nach Genest und Mackay [32], wie folgt definiert werden

$$C(u_1, \dots, u_N) = \begin{cases} \varphi^{-1}(\varphi(u_1) + \dots + \varphi(u_N)), & \text{für } \sum_{n=1}^N \varphi(u_n) \leq \varphi(0), \\ 0, & \text{sonst,} \end{cases} \quad (2.185)$$

wobei φ eine Funktion aus C^2 ist mit $\varphi(1) = 0$, $\varphi'(u) < 0$ und $\varphi''(u) > 0$ für alle $0 \leq u \leq 1$. $\varphi(u)$ wird als Erzeuger der Copula bezeichnet. Die Gumbel-, Joe- und Kimeldorf-Sampson-Copulas sind Beispiele für Archimedische Copulas. Über diesen Erzeuger kann festgestellt werden, ob den Daten eine Archimedische Copula zu Grunde¹⁰⁶ liegt.

¹⁰⁶Genest und Rivest [33] habe diese Methode entwickelt.

Sei X ein Vektor mit N Zufallsvariablen, C die Copula zu der der Erzeuger φ gehört und K eine Funktion, die definiert ist als

$$K(u) = P\{C(U_1, \dots, U_N) \leq u\}. \quad (2.186)$$

Diese Funktion kann geschrieben werden als

$$K(u) = u + \sum_{n=1}^N (-1)^n \frac{\varphi^n(u)}{n!} \chi_{n-1}(u), \quad (2.187)$$

mit $\chi_n(u) = \frac{\partial_u \chi_{n-1}(u)}{\partial_u \varphi(u)}$ und $\chi_0(u) = \frac{1}{\partial_u \varphi(u)}$ ¹⁰⁷. Für den bivariaten Fall vereinfacht sie sich zu

$$K(u) = u - \frac{\varphi(u)}{\varphi'(u)}. \quad (2.188)$$

Ein nicht parametrischer Schätzer für $K(u)$ ist gegeben durch

$$\hat{K}(u) = \frac{1}{T} \sum_{t=1}^T 1_{[\vartheta_t \leq u]}, \quad (2.189)$$

mit

$$\vartheta_t = \frac{1}{T-1} \sum_{i=1}^T 1_{[x_1^t < x_1^i, \dots, x_N^t < x_N^i]}. \quad (2.190)$$

Die Idee dabei ist, dass $\hat{K}(u)$ durch Wahl der Parameter einer Familie von Archimedischen Copulas angepasst wird.

2.4.14.4 Auswahl der Copula über die empirische Copula

Die empirische Copula ist unabhängig von einem Parameter und ist somit ein nichtparametrischer Schätzer für die zu Grunde liegende Copula. Steht eine Auswahl der möglichen Copulas fest, die in Frage kommen könnten, so wird im ersten Schritt der Parameter¹⁰⁸ $\hat{\theta}$ der jeweiligen Copula geschätzt. Dies kann mit den vorgestellten Schätzern aus Abschnitt 2.4.11 geschehen. Anschließend wird mit der

¹⁰⁷siehe Barbe et. al. [6]

¹⁰⁸Es können auch mehrere Parameter einer Copula geschätzt werden, auf Grund des besseren Verständnisses wird nur die Einzahl benutzt.

L^2 Norm der Abstand zwischen der empirischen Copula \hat{C} und der Copula aus der Auswahl mit den geschätzten Parametern bestimmt.

$$d_2(\hat{C}, C_{\hat{\theta}}) = \|\hat{C} - C_{\hat{\theta}}\|_{L^2} := \left(\int \dots \int_{[0,1]^N} |\hat{C}(u) - C_{\hat{\theta}}(u)|^2 du \right)^{\frac{1}{2}} \quad (2.191)$$

u ist ein Vektor da $N \geq 2$. Da die empirische Copula auf diskreten Punkten berechnet wird, kann das Integral auch als Summe geschrieben werden. Dies sieht für die diskrete Stützstellen wie folgt aus

$$\begin{aligned} d_2(\hat{C}_T, C_{\hat{\theta}}) &= \|\hat{C}_T - C_{\hat{\theta}}\|_{L^2} \\ &= \left(\sum_{t_1=1}^T \dots \sum_{t_N=1}^T \left[\hat{C}_T\left(\frac{t_1}{T}, \dots, \frac{t_N}{T}\right) - C_{\hat{\theta}}\left(\frac{t_1}{T}, \dots, \frac{t_N}{T}\right) \right]^2 \right)^{\frac{1}{2}}, \end{aligned} \quad (2.192)$$

dabei sei N die Dimension und $\left\{ \left(\frac{t_1}{T}, \dots, \frac{t_N}{T}\right) : 1 \leq n \leq N, t_n = 0, \dots, T \right\}$.

$d_2(\hat{C}_T, C_{\hat{\theta}})$ beschreibt somit den Unterschied zwischen der empirischen und der parametrischen Copula und ist damit ein Maß für die Güte der Übereinstimmung. Mit diesem Verfahren wird die Familie von Copulas ausgewählt, die den kleinsten Wert für $d_2(\hat{C}_T, C_{\hat{\theta}})$ besitzt. Es ist noch zu beachten, dass bei der Abstandsmessung über die Stützstellen deren Anzahl wichtig ist. Wird bei der Berechnung bei jeder Familie nicht die gleiche Anzahl an Stützstellen verwendet, so sollte der Wert von $d_2(\hat{C}_T, C_{\hat{\theta}})$ durch die Anzahl T geteilt werden. Im diskreten Fall ist die Güte dann gegeben durch

$$\frac{d_2(\hat{C}_T, C_{\hat{\theta}})}{T}. \quad (2.193)$$

2.4.14.5 Auswahl der Copula über die Tail-Dependence-Funktion

In diesem Abschnitt soll eine weitere Möglichkeit zur Auswahl einer Copula vorgestellt werden. Eine Copula beschreibt die Abhängigkeit zwischen univariaten Verteilungen. In der Risikoanalyse sind besonders die Flanken von Interesse. Die Abhängigkeit in den Flanken wird wie schon gesehen durch die Tail-Dependence-Funktionen beschrieben. Bei einer bivariaten Copula können vier verschiedene Flanken betrachtet werden. Sie werden durch die vier Quadranten ausgedrückt (siehe Abb. 2.36). Die Tail-Dependence-Funktion berechnet die be-

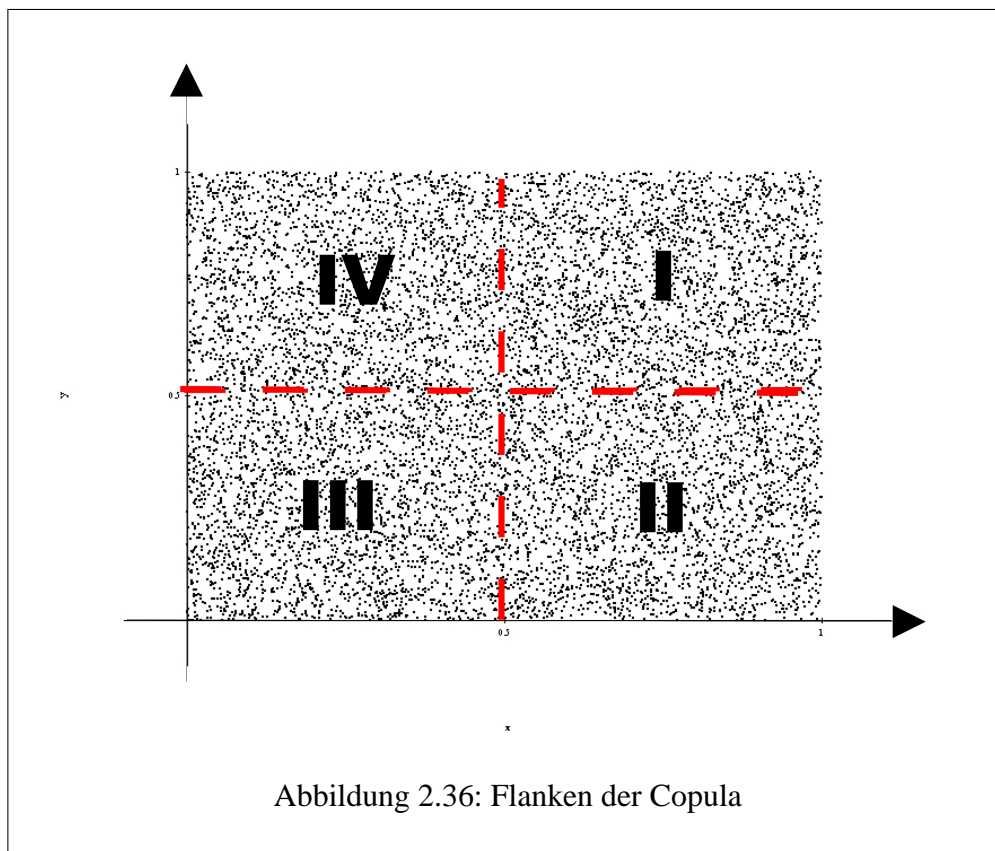


Abbildung 2.36: Flanken der Copula

dingte Wahrscheinlichkeit, dass die Randverteilungen über einer Schranke u liegen, gegeben dass schon eine Randverteilung über dieser Schranke liegt. Für die vier Quadranten lassen sich die Tail-Dependence-Funktionen wie folgt angeben:

1. $\chi_I(u) = \frac{P(U > u, V > u)}{P(U > u)} = \frac{1 - 2u - C(u, u)}{1 - u}, u \in (0, 1),$
2. $\chi_{II}(u) = \frac{P(U > u, V \leq 1 - u)}{P(U > u)} = \frac{u - C(1 - u, u)}{1 - u}, u \in (0, 1),$
3. $\chi_{III}(u) = \frac{P(U \leq u, V \leq u)}{P(U \leq u)} = \frac{C(u, u)}{u}, u \in (0, 1),$
4. $\chi_{IV}(u) = \frac{P(U \leq u, V > 1 - u)}{P(U \leq u)} = \frac{1 - u - C(u, 1 - u)}{u}, u \in (0, 1).$

Mit diesen vier Funktionen kann die Masse in den jeweiligen Flanken bzw. Quadranten, die über einer Schranke u liegen, bestimmt werden. Die Tail-Dependence-Funktionen beschreiben das Verhalten der Abhängigkeit in Richtung der jeweiligen Flanke¹⁰⁹. Diese Tail-Dependence-Funktion kann auch empirisch leicht be-

¹⁰⁹Es wird kein Grenzwert der Tail-Dependence-Funktion gebildet.

rechnet werden. Die Formeln ergeben sich direkt aus den oben gegebenen bedingten Wahrscheinlichkeiten. Sei (x_i, y_i) , $i = 1, \dots, n$ ein bivariater Datensatz auf $[0, 1] \times [0, 1]$, dann sehen die vier empirischen Tail-Dependence-Funktionen, wie folgt aus:

1. $\hat{\chi}_I(u) = \frac{\sum_{i=1}^n 1_{(x_i > u)} 1_{(y_i > u)}}{\sum_{i=1}^n 1_{(x_i > u)}}, u \in (0, 1).$
2. $\hat{\chi}_{II}(u) = \frac{\sum_{i=1}^n 1_{(x_i > u)} 1_{(y_i \leq 1-u)}}{\sum_{i=1}^n 1_{(x_i > u)}}, u \in (0, 1).$
3. $\hat{\chi}_{III}(u) = \frac{\sum_{i=1}^n 1_{(x_i \leq u)} 1_{(y_i \leq u)}}{\sum_{i=1}^n 1_{(x_i \leq u)}}, u \in (0, 1).$
4. $\hat{\chi}_{IV}(u) = \frac{\sum_{i=1}^n 1_{(x_i \leq u)} 1_{(y_i > 1-u)}}{\sum_{i=1}^n 1_{(x_i \leq u)}}, u \in (0, 1).$

Für einen großen Stichprobenumfang nähert sich die empirische Tail-Dependence-Funktion der zu Grunde liegenden immer mehr an. Es liegt deshalb nahe, den Abstand zwischen den beiden Funktionen als Güte der Übereinstimmung zwischen den empirischen Daten und der ausgewählten Copula zu nehmen. Dieser wird berechnet durch

$$\begin{aligned}
 d(\hat{\chi}_t, \chi_{t, C_{\hat{\theta}}}) &= \|\hat{\chi}_t - \chi_{t, C_{\hat{\theta}}}\|_{L^2} \\
 &:= \left(\int_{[0,1]} |\hat{\chi}_t(u) - \chi_{t, C_{\hat{\theta}}}(u)|^2 du \right)^{1/2}, t \in \{I, II, III, IV\}
 \end{aligned}
 \tag{2.194}$$

Damit diese Methode der Auswahl angewandt werden kann, muss eine Vorauswahl der zu verwendeten Copulas getroffen werden. Für diese Auswahl können mehrere Möglichkeiten hilfreich sein:

- Test auf elliptische Copula (siehe Abschnitt 2.4.14.2, S. 140),
- Test auf Archimedische Copula (siehe Abschnitt 2.4.14.3, S. 142),
- Visualisierung der Daten (siehe Kapitel 1, Abschnitt 1.2, S. 5).

Die Tests sind als Hilfsmittel zur Vorauswahl zu sehen. Die Festlegung auf eine endliche Menge an Familien von Copulas wird meist über Vorinformationen oder über Visualisierung der Daten getroffen. Ist die Menge an Familien von Copulas festgelegt worden, werden für jede Familie die Parameter geschätzt. Dies kann auf verschiedene Arten geschehen (siehe Abschnitt 2.4.11, 128). Aus jeder Familie ist

dann eine Copula bestimmt, die optimal im Sinne des Schätzverfahrens ist, für die der Abstand berechnet werden soll. Da der Abstand wegen der empirischen Daten an diskreten Stellen berechnet werden muss, sieht die Abstandsberechnung wie folgt aus:

$$d(\hat{\chi}_t, \chi_{t, C_{\hat{\delta}}}) = \left(\sum_{i=1}^n \left(\hat{\chi}_t \left(\frac{i}{n+1} \right) - \chi_{t, C_{\hat{\delta}}} \left(\frac{i}{n+1} \right) \right)^2 \right)^{1/2} \quad (2.195)$$

Der Abstand kann durch die Anzahl der Berechnungsstellen n geteilt werden, damit dieser mit anderen Abständen, die mit unterschiedlichen vielen Stützstellen berechnet wurden, vergleichbar wird. Dies ist jedoch nicht notwendig für die Auswahl der Copula, da immer der selbe empirische Datensatz zur Berechnung verwendet wird.

Als nächstes wird das Auswahlverfahren über die Tail-Dependence-Funktion mit dem Auswahlverfahren über die empirische Copula verglichen. Ausgangspunkt bilden Simulation¹¹⁰ der Gauss-, Clayton- und Farlie-Gumbel-Morgenstern Copulas. In Tabelle 2.6 wurden die Ergebnisse aufgelistet. Die erste Spalte gibt an unter welcher Copula und mit welchem Parameter die Stichprobe erzeugt wurde. Die zweite Spalte enthält den Parameter der Copula, der zur Erzeugung der Copula eingegangen ist. Der Parameter wird mit Kendall's τ geschätzt. Es gilt

- Gauss: $\rho = \sin((\pi/2)\tau)$
- Clayton: $\delta = \frac{2\tau}{1-\tau}$
- Farlie-Gumbel-Morgenstern: $\delta = 4.5\tau$

In der dritte Spalte wird in Prozenten die Wahrscheinlichkeit angegeben, dass das Auswahlverfahren über die empirische Copula eine richtige Auswahl trifft. In der letzten Spalte werden die Wahrscheinlichkeit in Prozenz für die richtige Auswahl über die Tail-Dependence-Funktion aufgelistet. Beide Verfahren benutzen zur Berechnung 100 Stützstellen, es sei angemerkt, dass dann die empirische Copula an 10000 Stützstellen¹¹¹ berechnet wird. Es wurden für jeweils jede Zeile mindestens 500 Simulationen gemacht. Aus Tabelle 2.6 kann man entnehmen,

¹¹⁰Stichprobenumfang $N = 10000$

¹¹¹Die Stützstellen sind auf einem Gitter (100×100) angeordnet.

empirische Daten	Parameter	emp. Copula	Tail-Dependence
		Trefferwahrscheinlichkeit	Trefferwahrscheinlichkeit
Clayton	$\delta = 1.0$	0	0
Clayton	$\delta = 100.0$	>99.9	>99.9
FGM	$\delta = -0.9$	75%	75%
FGM	$\delta = 0$	>99%	>99%
FGM	$\delta = 0.9$	85%	85%
Gauss	$\rho = 0.9$	>99%	>99%
Gauss	$\rho = 0.8$	>99%	>99%
Gauss	$\rho = 0.7$	>99%	>99%
Gauss	$\rho = 0.6$	>99%	>99%
Gauss	$\rho = 0.5$	>99%	>99%
Gauss	$\rho = 0.4$	>99%	>99%
Gauss	$\rho = 0.3$	11.4%	10%
Gauss	$\rho = 0.2$	0%	0%
Gauss	$\rho = 0.1$	0%	0%
Gauss	$\rho = 0.0$	0%	0%
Gauss	$\rho = -0.1$	0%	0%
Gauss	$\rho = -0.2$	0%	0%
Gauss	$\rho = -0.3$	25 %	20%
Gauss	$\rho = -0.4$	>99%	>99%
Gauss	$\rho = -0.5$	>99%	>99%
Gauss	$\rho = -0.6$	>99%	>99%
Gauss	$\rho = -0.7$	>99%	>99%
Gauss	$\rho = -0.8$	>99%	>99%
Gauss	$\rho = -0.9$	>99%	>99%

Tabelle 2.6: Ergebnisse der Simulation

dass beide Auswahlmethoden¹¹² sehr ähnliche Ergebnisse liefern. Die erste Vorauswahl findet schon beim Schätzen der Parameter statt. Im Falle von nahezu unabhängigen Daten haben die beiden Auswahlverfahren Probleme eine richtige Aussage zu treffen. Anzumerken ist:

- Keines der beiden Verfahren wählt immer die richtige Copula aus, die auch der Simulation zu Grunde lag, deshalb sollte die Auswahl durch eine Visualisierung überprüft werden. Liegen die Abstände zweier Copulas nahe bei einander, so sollten beide in den weiteren Berechnungen benutzt werden.
- Das Verfahren der Auswahl über die Tail-Dependence-Funktion ist wesentlich schneller, wie das der Auswahl über die empirische Copula. Beim Auswahlverfahren basierend auf der empirischen Copula werden $T \times T$ Stützstellen betrachtet. Dies kann zu einer höheren Genauigkeit führen, da bei dem Auswahlverfahren mit der Tail-Dependence-Funktion nur $4T$ Stützstellen¹¹³ betrachtet werden.
- Die Auswahl über die Tail-Dependence-Funktion lässt sich dort gut anwenden, wo besonders die Abhängigkeiten von einer Flanke von Interesse ist. Dieses Verfahren kann aber nur auf zweidimensionale Copulas angewandt werden. Erweiterungsmöglichkeiten auf den mehrdimensionalen Fall $n \geq 3$ werden in dieser Arbeit nicht betrachtet.

Das Verfahren der Auswahl über die Tail-Dependence-Funktion ist in dieser Arbeit nur an Hand von drei Familien von Copulas überprüft worden. Es müsste auf weitere Copulas angewandt werden, wobei die gelieferten Ergebnisse noch überprüft werden müssten. Dieses Verfahren steht noch in der Entwicklungsphase und ist noch erweiterungsbedürftig. Es bietet sich in seinem momentanen Entwicklungsstand als Entscheidungshilfe an. Eine Entscheidung sollte nicht nur basierend auf dieses eine Auswahlverfahren stattfinden.

Um dieses Verfahren noch verbessern zu können, müssen folgende Fragestellungen weiter verfolgt werden:

- Berücksichtigung der Anzahl der Stützstellen und die Größe des Stichprobendatensatzes.

¹¹²Stehen + oder - in Klammern so ist das Ergebnis gerade so angenommen oder verworfen worden.

¹¹³Dabei ist noch zu beachten, dass immer zwei der Tail-Dependence-Funktion mit den gleichen Stützstellen berechnet werden, z. B. Quadrant I und III.

- Soll der Abstand der Tail–Dependence–Funktion über ganz $[0, 1]$ oder nur über ein Teilintervall berechnet werden?
- Hilft eine Gewichtsfunktion bei der Verbesserung der Auswahlergebnisse?

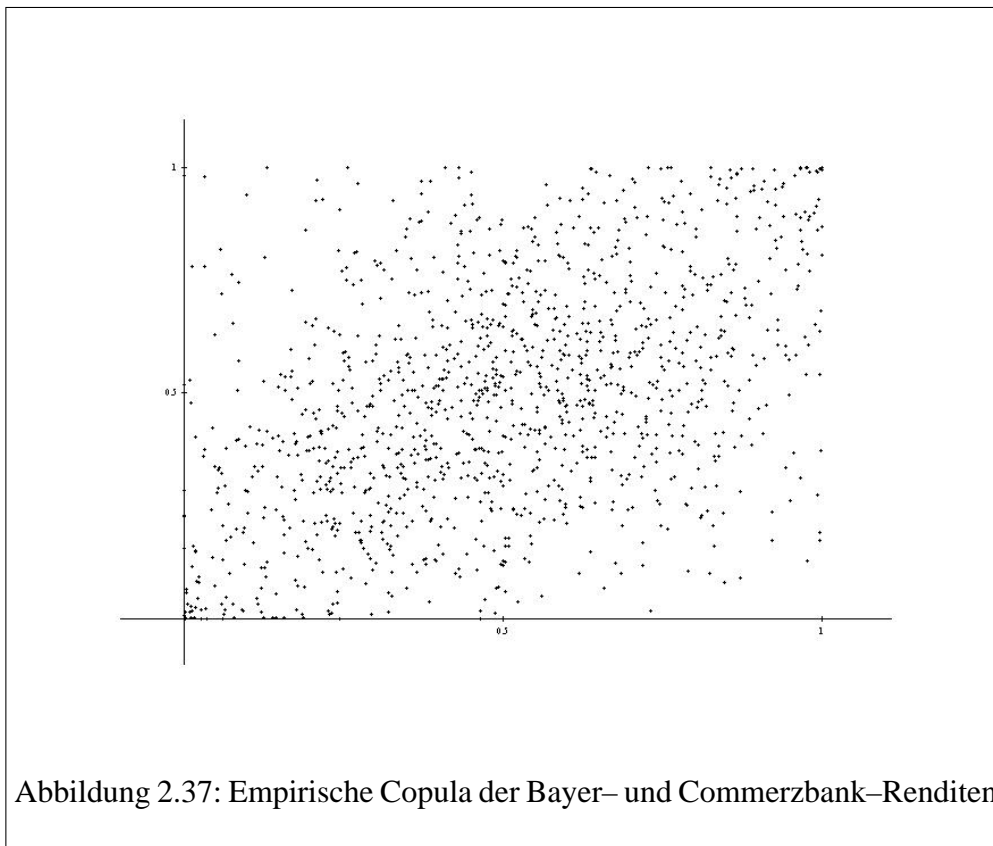
Als nächstes werden die beiden Auswahlverfahren auf Finanzdaten angewandt.

2.4.15 Datenanalyse mit Finanzdaten

Die Schwächen der Modellierung von Finanzdaten mit multivariaten Verteilungen liegt in der fehlenden Flexibilität. Es konnte bei der multivariaten Studentischen Verteilung festgestellt werden, dass bei der Schätzung mit dem EM–Algorithmus eine Randverteilung überschätzt und die andere unterschätzt wurde. Durch ein flexibleres Modell wäre es möglich, die Verteilungen für die univariaten Randverteilungen besser anzupassen. Copulas bieten sich in diesem Fall an. Bei Copulas kann, wie schon erwähnt, die Abhängigkeit unabhängig von den Randverteilungen betrachtet werden. Bei der Transformation der Randverteilung auf $[0, 1]$ wird die Verteilungsfunktion der ausgewählten Verteilung für diese benutzt. Die Verteilungsfunktion ist eine monotone Funktion, es ändert sich bei dieser Transformation somit nichts an der Abhängigkeitsstruktur, die die Copula widerspiegelt, nichts. Die Copula gibt die Abhängigkeit der Randverteilungen wieder.

Aus diesem Grund sollen in diesem Abschnitt verschiedene Finanzdaten auf ihre Abhängigkeit untersucht werden. Es wurde die Abhängigkeit der Bayer–Renditen mit den Commerzbank–Renditen untersucht. Die Abhängigkeit kann nach Transformation der Finanzdaten über eine Verteilung in einem Scatterplot (siehe Abb. 2.37) ausgegeben werden. In Abbildung 2.38 wird anschaulich der Vergleich der empirischen Copulas mit parameterischen Copulas dargestellt. Der Contourplot gibt die empirische Copula (schwarz) im Vergleich zur Gauss–Copula (rot), Clayton–Copula (grün) und Farlie–Gumbel–Morgenstern–Copula (blau) wieder. Für die parameterischen Copulas wurden folgende Parameter geschätzt (siehe Tabelle 2.7). Es kann festgestellt werden, dass die Gauss–Copula die Abhängigkeit der Finanzdaten¹¹⁴ vernünftig wiedergibt. Dies kann auch durch den Plot der Tail–Dependence–Funktion nicht verworfen werden. Auf Grund des geringen Stichprobenumfangs können die Plots, die die Abhängigkeitsstruktur wiedergeben, zur Hilfe genommen werden, um eine Auswahl für eine Copula treffen

¹¹⁴Bivariate Daten vom Umfang $T = 1300$.



zu können. Die Tail-Dependence-Funktionen der Gauss-Copula zur empirischen Tail-Dependence-Funktionen werden in Abbildung 2.39 ausgegeben. Stimmen die beiden Tail-Dependence-Funktionen überein, so wird im Plot eine Gerade ausgegeben. Zur besseren Übersicht wurde eine Regressionsgerade durch die Daten der Tail-Dependence-Funktionen gelegt. Der Plot kann wie folgt definiert werden:

Definition 35 (T-T-Plot). Sei $\hat{\chi}$ die empirische Tail-Dependence-Funktion und $\chi_{C_{\hat{\theta}}}$ die Tail-Dependence-Funktion einer parameterischen Copula mit geschätztem Parameter¹¹⁵ $\hat{\theta}$, dann ist der T-T-Plot definiert als Plot der Punkte

$$(\hat{\chi}(u), \chi_{C_{\hat{\theta}}}(u)), u \in (0, 1). \quad (2.196)$$

In der nächsten Abbildung werden sowohl die Tail-Dependence-Funktionen der Clayton¹¹⁶ – als auch die der Farlie¹¹⁷–Copula wiedergegeben. Diese Ergebnisse

¹¹⁵bzw. Parametervektor

¹¹⁶siehe Abbildung 2.40

¹¹⁷siehe Abbildung 2.41

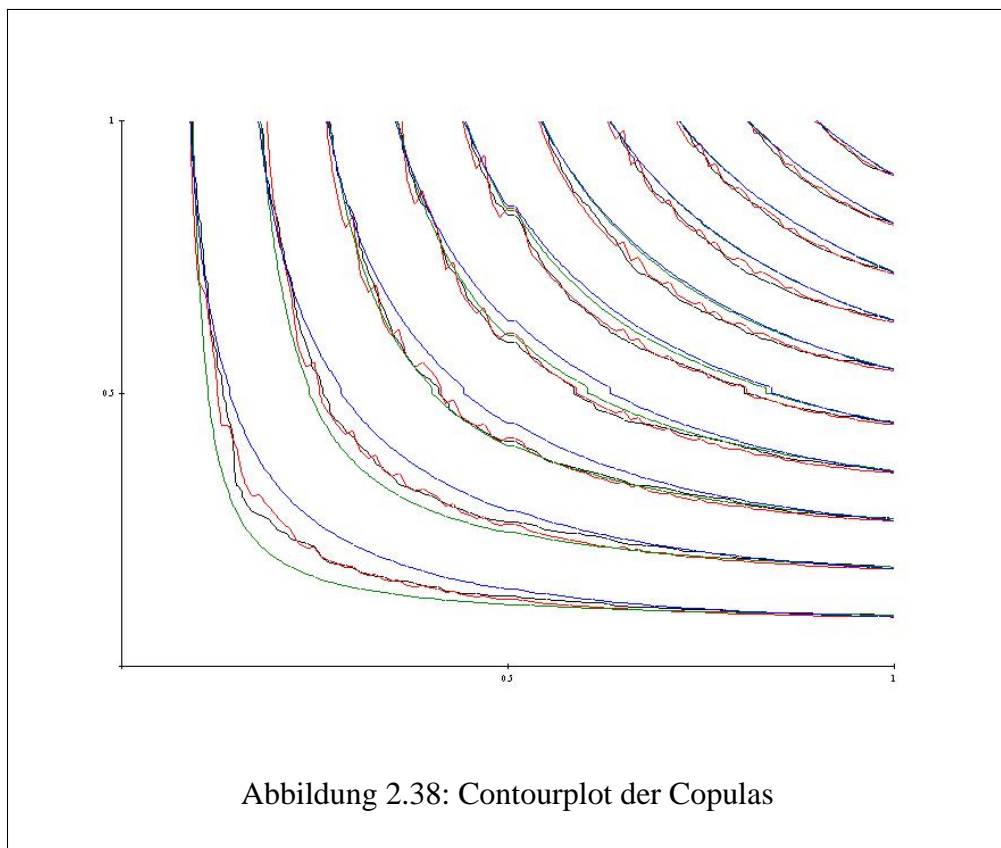


Abbildung 2.38: Contourplot der Copulas

bekommt man auch bei der Verwendung von zwei andere Aktien¹¹⁸. In Abbildung 2.42 wurde die Commerzbank-Aktie mit der Deutschen Bank-Aktie verglichen. Es werden drei Scatterplots ausgegeben, der erste Plot ist der der empirischen Daten, der zweite ist der der Gauss-Copula, der dritte gibt die Clayton-Copula wieder. Bei der Gauss- und der Clayton-Copula wurden zur Simulation die geschätzten Parameter $\rho = 0.596118$ und $\delta_C = 1.3703$ verwendet. Auch aus dem Vergleich der Simulationsdaten mit den empirischen Daten kann man erkennen, dass die Gauss-Copula die Abhängigkeit der Finanzdaten besser wiedergibt, wie die Clayton-Copula. Dies bestätigt auch die Aussage von Malevergne und Sornette [57].

Auch bei der Analyse von empirischen Daten mit Copulas gibt es weitere Forschungsmöglichkeiten. In dieser Analyse wurden Daten über einen Zeitraum von ungefähr sechs Jahren betrachtet. Es bleibt die Frage offen, wie sich die Analyse

¹¹⁸Für die Analyse werden deren Renditen betrachtet.

Copula	Parameter
Gauss	$\rho = 0.469318$
Clayton	$\delta = 0.902798$
Farlie–Gumbel–Morgenstern	$\delta = 1.3395$

Tabelle 2.7: Parameter der Copulas für die Bayer/Commerzbank–Renditen

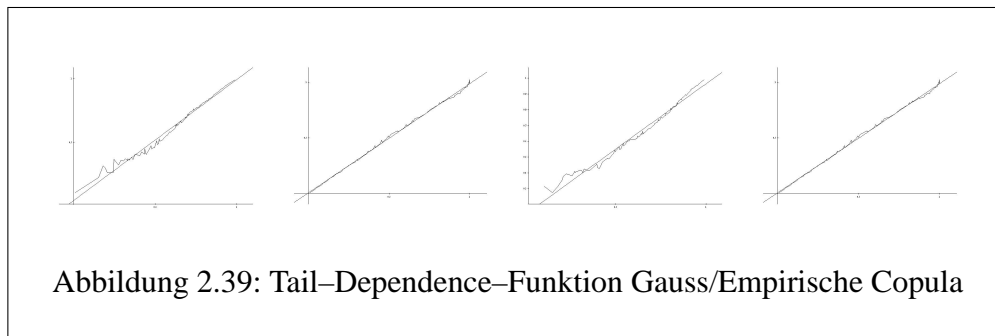
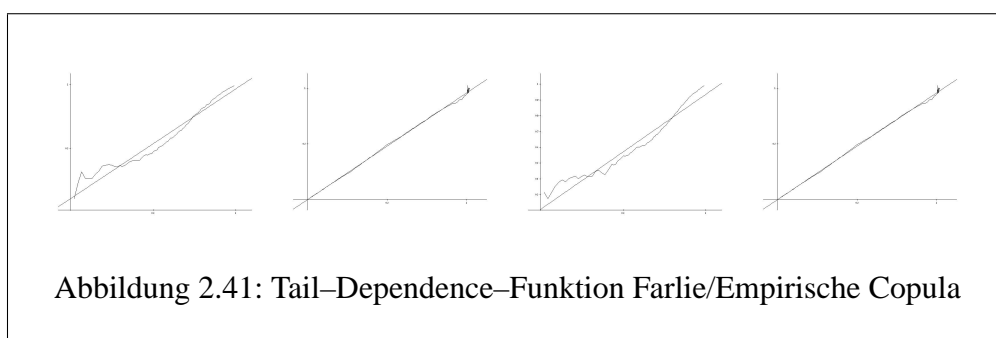
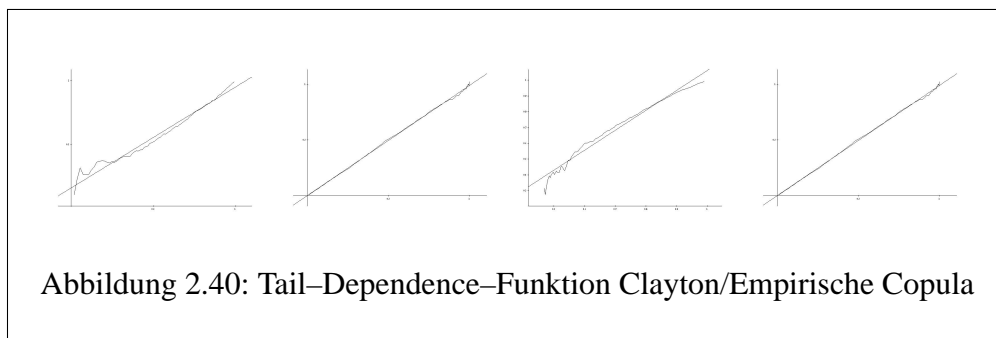


Abbildung 2.39: Tail–Dependence–Funktion Gauss/Empirische Copula

von Copulas in einem kürzeren Zeitraum verhält. Dabei sollte auch berücksichtigt werden, ob auf dem Markt starke oder schwache Volatilität herrscht.



2.4.16 Abschlussbemerkung zum finanzmathematischen Teil

In den vorangehenden Abschnitten wurden verschiedene Alternativen zum Normalverteilungsmodell bei Finanzdaten vorgestellt. Das vorgestellte Zeitreihenmodell gibt die zuvor erwähnten Stylized Facts entsprechend wieder. Dieses Modell ist jedoch schwierig zu handhaben. Die Eigenschaften des Modells konnten in der vorliegenden Arbeit berechnet und mit Finanzdaten verglichen werden. Weiterführend müssten Schätzer für das Modell entwickelt werden. Allerdings wird sich dies auf Grund der Komplexität des Modells schwierig gestalten.

Bei der Entwicklung von multivariaten Modellen wurde anstatt der Verwendung von multivariaten Verteilungen der Ansatz mit Copulas betrachtet. Mit Copulas können flexiblere multivariate Modelle unter der Verwendung von den univariaten Randverteilungen (univariate Modelle) konstruiert werden. Es hat sich gezeigt, dass mittels Copulas flexiblere multivariate Modelle mit univariaten Randverteilungen konstruiert werden können. Den univariaten Randverteilungen können die schon vorgestellten univariaten Modelle zu Grunde liegen. Hierzu wurden für die Copulas verschiedene Schätzer eingeführt. Außerdem wurden Tests und Aus-

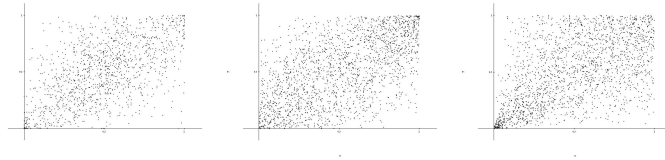


Abbildung 2.42: Banken (links), Gauss-Copula (Mitte), Clayton-Copula (rechts)

wahlverfahren zur Wahl der richtigen Copula vorgestellt. In diesem Bereich gibt es noch weitere Forschungsmöglichkeiten, die den Rahmen dieser Arbeit sprengen würden. Besonders das zweite Auswahlverfahren über die Tail-Dependence-Funktionen bietet noch interessante Ansatzpunkte und Weiterentwicklungsmöglichkeiten. Außerdem wurden Simulationsverfahren für die verschiedenen Copulas vorgestellt.

Weiterer Forschungsbedarf besteht bei der Kombination von Copulas und Zeitreihen. Dabei sind noch viele empirische Analysen notwendig, damit die Eigenschaften der Finanzdaten entsprechend wiedergegeben werden. Bei der Finanzdatenanalyse ist es wichtig, die Modelle immer an der Realität zu überprüfen.

Anhang A

Ergänzungen zur Client/Server–Architektur

A.1 Mathematische Grundlagen

A.1.1 Window–Viewport–Transformation

Die Window–Viewport–Transformation wandelt die 2D–Koordinaten des Weltraumes oder des 2D–Übergangsraumes¹ in 2D–Koordinaten des Bildraumes um. Dabei soll das Verhältnis der Größen und Abstände nicht verändert werden, damit die Objekte aus Weltraum realistisch im Bildraum abgebildet werden können. Es dürfen bei der Window–Viewport–Transformation keine Transformationen durchgeführt werden, die die Verhältnisse der Objekte² zueinander verändern (siehe Freisleben und Baumgart [29]). Ebenso dürfen auch keine Seitenverhältnisse der Objekte geändert werden³. Transformationen sind Verschiebungen, Skalierungen, Scherungen oder Rotationen eines Punktes. Die Window–Viewport–Transformation setzt sich aus vier Schritten zusammen.

1. Clipping
2. Translation⁴ des Windows⁵ in den Koordinatenursprung

¹siehe Kapitel 1.2.4.5, S. 13 und AnhangA.1.3, S. 160

²Man stelle sich ein grafisches Objekt, wie z. B. ein Haus vor.

³D. h. auch die Winkel in den Objekten sollen erhalten bleiben.

⁴Auch Verschiebung genannt.

⁵Window ist der Ausschnitt aus dem Weltraum, der ausgegeben werden soll.

3. Skalierung auf die Größe des Viewports

4. Translation in den Viewport

Der erste Schritt, das Clipping, wird meist vom verwendeten Fenstersystem übernommen.

Mathematisch werden jetzt die drei Schritte ohne das Clipping beschrieben. Seien (x_{min}, y_{min}) bzw. (x_{max}, y_{max}) der linke untere bzw. rechte obere Eckpunkt des Windows. Für den Viewport, den Ausschnitt aus dem Bildraum, seien (u_{min}, v_{min}) bzw. (u_{max}, v_{max}) der linke untere bzw. rechte obere Eckpunkt. Ein Punkt (x, y) aus dem Window⁶ kann folgendermaßen in den Bildraum abgebildet werden. Bei der Darstellung werden homogene Koordinaten verwendet. T bezeichnet die Translation in Matrixdarstellung und S die Skalierung in Matrixdarstellung.

$$\begin{aligned}
 M_{WV} &= T(u_{min}, v_{min})S\left(\frac{u_{max} - u_{min}}{x_{max} - x_{min}}, \frac{v_{max} - v_{min}}{y_{max} - y_{min}}\right)T(-x_{min}, -y_{min}) \\
 &= \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} & 0 & 0 \\ 0 & \frac{v_{max} - v_{min}}{y_{max} - y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} & 0 & -x_{min} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} + u_{min} \\ 0 & \frac{v_{max} - v_{min}}{y_{max} - y_{min}} & -y_{min} \frac{v_{max} - v_{min}}{y_{max} - y_{min}} + v_{min} \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Die Transformation des Punktes (x, y) kann dann geschrieben werden als

$$\begin{aligned}
 \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= M_{WV} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} & 0 & -x_{min} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} + u_{min} \\ 0 & \frac{v_{max} - v_{min}}{y_{max} - y_{min}} & -y_{min} \frac{v_{max} - v_{min}}{y_{max} - y_{min}} + v_{min} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
 \end{aligned} \tag{A.1}$$

Bei der internen Darstellung im Prototypen wird auf homogene Koordinaten verzichtet. Diese Darstellung ist nur von Vorteil, wenn mehrere Transformationen

⁶Für (x, y) gilt $x_{min} < x < x_{max}$, $y_{min} < y < y_{max}$.

hintereinander ausgeführt werden sollen. Die Window–Viewport–Transformation kann in diesem Fall noch weiter vereinfacht werden. Weitere Informationen zur Window–Viewport–Transformationen sind in Freisleben und Baumgart[29] zu finden.

A.1.2 Rotation

Rotationen werden in der Client/Server–Architektur benötigt, um dreidimensionalen Plots von allen Seiten betrachten zu können. Nur so ist gewährleistet, dass alle Eigenschaften eines statistischen Plots erkennbar sind. Die Ausführung der Rotation erfolgt im Objekt–Server vor der Projektion. Im dreidimensionalen Raum gibt es drei Rotationen um die jeweilige Koordinatenachse. Sie werden mathematisch so beschrieben:

- *Rotation um die z–Achse in homogenen Koordinaten*

$$\begin{aligned} \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} & 1 \end{bmatrix} &= \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \\ &= \begin{bmatrix} x \cos \gamma - y \sin \gamma & x \sin \gamma + y \cos \gamma & z & 1 \end{bmatrix}^T \end{aligned}$$

- *Rotation um die x–Achse in homogenen Koordinaten*

$$\begin{aligned} \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} & 1 \end{bmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \\ &= \begin{bmatrix} x & y \cos \alpha - z \sin \alpha & y \sin \alpha + z \cos \alpha & 1 \end{bmatrix}^T \end{aligned}$$

- *Rotation um die y–Achse in homogenen Koordinaten*

$$\begin{aligned} \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} & 1 \end{bmatrix} &= \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \\ &= \begin{bmatrix} x \cos \beta + z \sin \beta & y & -x \sin \beta + z \cos \beta & 1 \end{bmatrix}^T \end{aligned}$$

Die Rotationen wurden hier in homogenen Koordinaten dargestellt. Im Prototypen wird darauf verzichtet, da immer nur eine Rotation ausgeführt werden muss. Die Berechnung der Rotation kann schneller in normaler Darstellung durchgeführt werden.

A.1.3 Projektion

Die Projektion ist neben der Window–Viewport–Transformation⁷ die wichtigste Berechnung, die der Objekt–Server bei der Ausgabe von 3D–Grafikobjekten durchzuführen hat. Ohne die Projektion kann die Ausgabe nicht erfolgen. Im Prototypen wird eine Zentralprojektion verwendet, da sie eine realistische Objektdarstellung gewährleistet.

Im Folgenden werden die mathematischen Grundlagen der verwendeten Zentralprojektion erläutert. Den Ausgangspunkt bildet ein rechtshändiges Koordinatensystem. Um eine Zentralprojektion definieren zu können, benötigt man den Standpunkt des Betrachters und die Projektionsebene. Bei der in dem Prototypen verwendeten Projektion werden, bevor die eigentliche Berechnung der Projektionskoordinaten durchgeführt wird, alle Objekte des Weltraumes⁸ in einen Quader der Größe $(-1, 1)^3$ transformiert. Diese Transformation hat den Vorteil, dass der Mittelpunkt des statistischen Plots im Ursprung des Koordinatensystem liegt. Dadurch wird bei der Rotation der Plot um die Koordinatenachsen gedreht, ohne das er sich aus dem Ursprung entfernt (siehe Freisleben und Baumgart [29]). Die Transformation auf die Größe des Quaders sieht folgendermaßen aus:

$$\begin{aligned}x' &= \frac{x + x_1 - x_0}{x_1 - x_0}, \\y' &= \frac{y + y_1 - y_0}{y_1 - y_0}, \\z' &= \frac{z + z_1 - z_0}{z_1 - z_0}.\end{aligned}$$

In Tabelle (A.1) werden die Schritte für die Transformation nochmals kurz erläutert. Es wird nur die x-Achse betrachtet, d. h. das Window⁹ hat die Koordinaten (x_{min}, x_{max}) . Die Projektion wird durch folgende Formel A.2 berechnet, wobei $p = (1, -5, 1)^T$ der Standpunkt des Betrachters ist, d. h. man schaut von links auf

⁷siehe A.1.1, S. 157

⁸Bemerkung: Der Weltraum entspricht dem Window. Er ist dreidimensional.

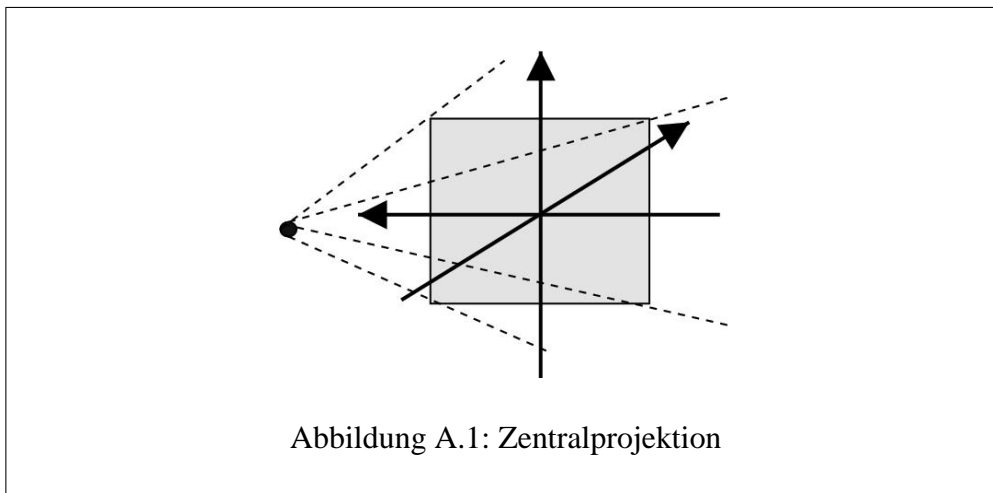
⁹Window: Ausschnitt des Weltraumes, der ausgegeben werden soll.

1. Schritt Verschiebung des linken Eckpunktes der Koordinatenachse in den Ursprung mit $-x_{min}$. Ebenso wird jeder andere Punkt im Window, auch x_{max} , um $-x_{min}$ verschoben.
2. Schritt Skalierung des Windows auf die Länge zwei mit $\frac{2}{x_{max}-x_{min}}$.
3. Schritt Verschiebung aller Punkte um minus eins. Alle Punkte liegen jetzt im Intervall $[-1, 1]$
4. Schritt Somit folgt für die Verschiebung des Punktes x :

$$x' = 2 \frac{x-x_{min}}{x_{max}-x_{min}} - 1 = \frac{x+x-x_1-x_0}{x_1-x_0}$$

Tabelle A.1: Schritte der Transformation

den Plot (siehe Abb. A.1). Die Projektionsebene der Zentralprojektion befindet



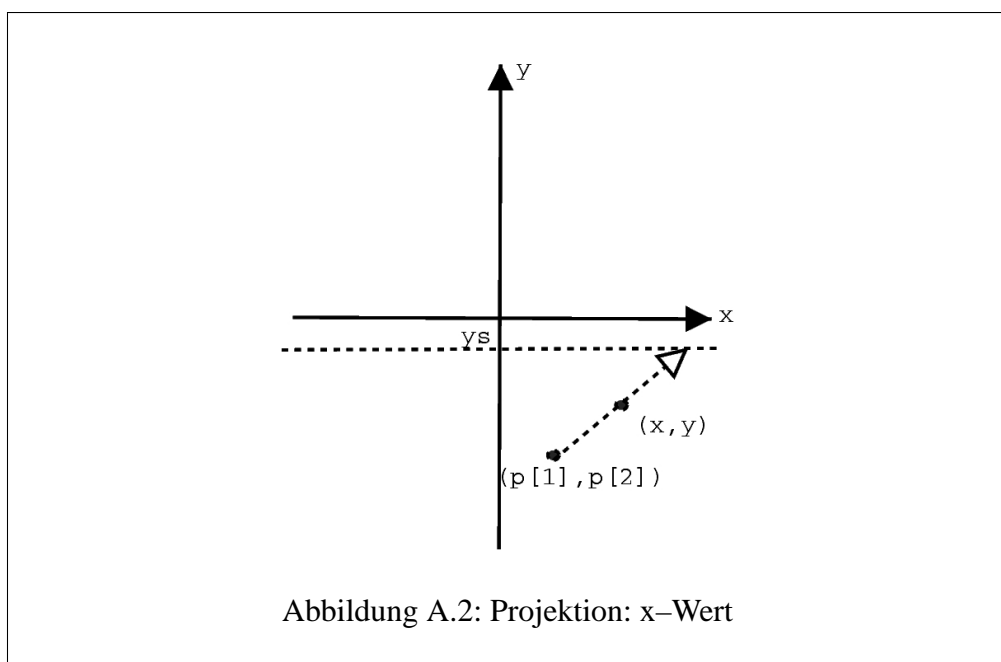
sich im Koordinatenursprung parallel zur x - und z -Achse. Die Berechnung der Projektion sieht dann folgendermaßen aus:

$$\begin{aligned}
 xs &= x - y \frac{p[1] - x}{p[2] - y}, \\
 ys &= z - y \frac{p[3] - z}{p[2] - y}.
 \end{aligned}
 \tag{A.2}$$

Dabei ist $p[1]$ die erste, $p[2]$ die zweite und $p[3]$ die dritte Koordinate des Punktes p . Den Schnittpunkt mit der Projektionsebene¹⁰ wird dann mit der Formel

$$x_s = \frac{p[1] - x}{p[2] - y}(y_s - p[2]) + x$$

berechnet, wobei für y_s der y -Wert der Projektionsebene genommen wird. In diesem Fall ist der Wert y_s gleich null. Der y -Wert wird mit der gleichen Formel und anderen Koordinatenpunkten berechnet. Anzumerken ist noch, dass die Zen-



tralprojektion nicht das reale Verhältnisse von Linien oder Punkten zueinander wiedergibt. Möchte man reale Verhältnisse haben, so muss eine Parallelprojektion verwendet werden.

¹⁰Für den x -Wert ist die Projektionsebene eine Gerade.

A.2 Entwurfsmuster oder Design Patterns

In diesem Abschnitt werden dem Prototyp der Client/Server-Architektur zu Grunde liegende Programmierstrukturen anhand von Entwurfsmustern¹¹ (siehe Gamma et. al.[31]) erklärt. Ein Entwurfsmuster beschreibt, motiviert und erklärt systematisch das Design, welches ein wiederkehrendes Design-Problem in einem objektorientierten System darstellt. Es beschreibt das Problem und dessen Lösung dar, wobei die Lösung durch eine Anordnung von Objekten und Klassen aufgezeigt wird. Die Entwurfsmuster können anschaulich durch die Unified Modelling Language (UML) beschrieben werden. Im Folgenden werden nur Design-Pattern dargestellt, die auch im Prototypen der Architektur verwendet wurden.

A.2.1 Factory oder Manager

Die Factory- bzw. Manager-Struktur wird in der Client/Server-Architektur sehr

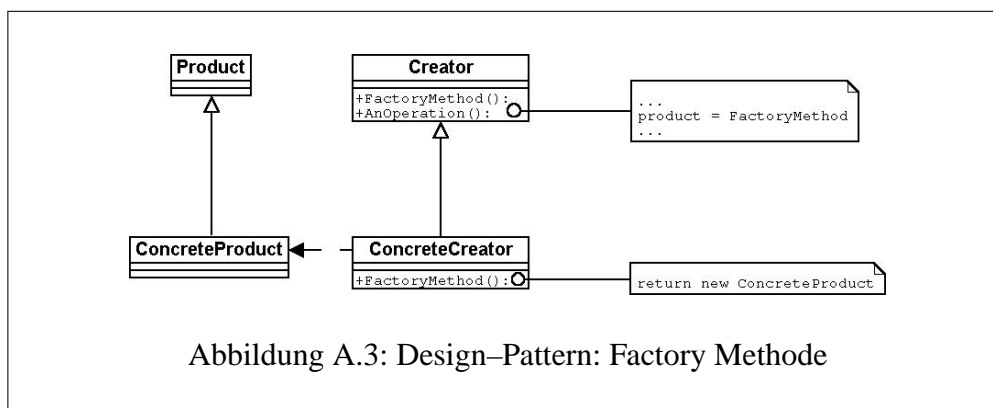


Abbildung A.3: Design-Pattern: Factory Methode

häufig verwendet. Sie wird dort eingesetzt, wo eine bestimmte Anzahl von Objekten verwaltet werden soll (Factory + Manager). Dabei soll ein Zugriffsmöglichkeit auf die Objekte nur über die Factory bzw. den Manager bestehen.

Die Factory¹² unterstützt im Unterschied zum Manager zusätzlich noch die Erzeugung der Objekte. Mittels einer Factory können zur Laufzeit beliebig viele Objekte erzeugt und wieder zerstört werden. Bei den Grafikobjekten im Objekt-Server wird eine Factory benutzt, da vor Laufzeitbeginn nicht feststeht, wieviele

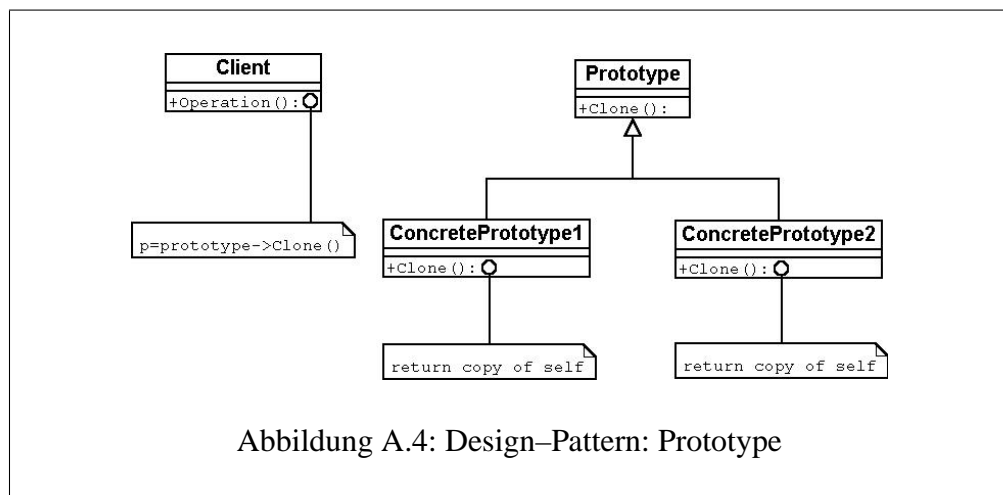
¹¹engl. Design Patterns

¹²Wird nur von der Verwaltung von Objekten gesprochen, so kann eine Factory in diesem Zusammenhang auch als Manager bezeichnet werden.

Grafikobjekte der Client benötigt. Der Client kann zur Laufzeit über die Factory die von ihm benötigten Grafikobjekte erzeugen. Diese werden nach der Erzeugung über die Factory angesprochen¹³, d. h. die Ausgabe der Grafikobjekte erfolgt nur über die Factory.

Ein Manager wird z. B. bei der Verwaltung der Display-Server benutzt (siehe 1.5.4, S. 38).

A.2.2 Prototyp

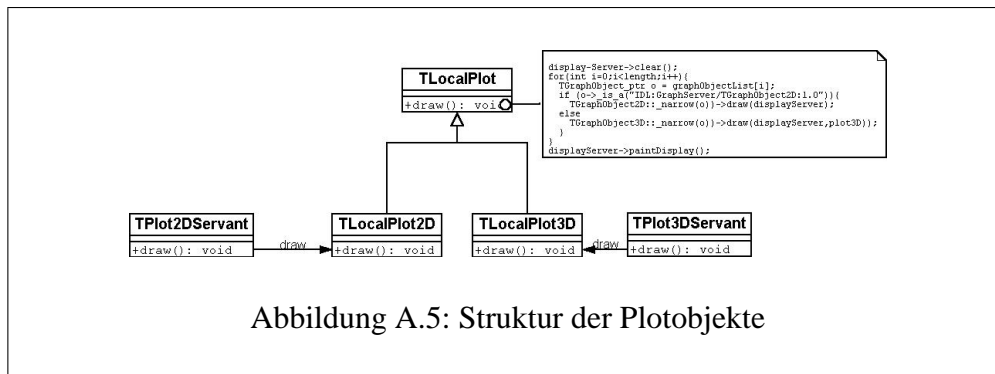


Die Struktur des Design Pattern Prototyp wird bei der Erzeugung von Grafikobjekten eingesetzt. Die Prototypen werden beim Starten des Objekt-Servers geladen. Mittels der vorhandenen Prototypen können zur Laufzeit reale Grafikobjekte erzeugt werden. Zur Erzeugung wird die **clone**-Methode benutzt, anschließend werden die Daten mit der **setData**-Methode an das Grafikobjekt übergeben. Mit der Erzeugungsart können beliebig viele Objekte zur Laufzeit erzeugt und wieder entfernt werden. Der Objekt-Server benötigt nicht die Anzahl der Grafikobjekte, die ein Client erzeugen will. Bei der Implementierung wird die Klasse Grafikfactory als Prototyp-Factory benutzt. Somit ist sichergestellt, dass alle erzeugten Grafikobjekte verwaltet und auch wieder zerstört werden.

¹³siehe Abschnitt A.2.2

A.2.3 Delegation

Das Entwurfsmuster Delegation wird dort eingesetzt, wo eine Mehrfachvererbung umgangen werden soll. Da nicht jede Programmiersprache die Mehrfachvererbung unterstützt (wie z. B. JAVA), wurde in der Implementierung auf diese verzichtet. Bei der Delegation wird der Methodenaufruf nicht vom Objekt ausgeführt welches die Anfrage bekommen hat, sondern dieses Objekt leitet den Aufruf zu einem anderen Objekt weiter. Dieses führt dann den Methodenaufruf aus und gibt das Ergebnis an das Ursprungsobjekt zurück. Diese Struktur wird im Objekt-Server bei den Plot-Objekten verwendet (siehe Abb. A.5).



A.3 CORBA–Schnittstelle des Prototypen

Listing A.1: Schnittstelle der Client/Server–Architektur

```
module GraphicServer {
  typedef sequence<double> doubleseq;
  typedef sequence<doubleseq> doublefield;
  typedef sequence<string> stringseq;
  typedef sequence<float> floatseq;

  interface TDisplayServer;
  interface TPlot;
  interface TDisplayManager;
  interface TPolyLine3D;
  interface TGraphObject;
  interface TColor;
  interface TClientCallback;

  typedef sequence<TDisplayManager> TDMseq;
  typedef sequence<TPolyLine3D> TPLseq;
  typedef sequence<TGraphObject> TGOseq;

  // Events sent by TDisplayServer

  interface TDisplayServerEvent {
  };

  interface TDisplayServerShutdownEvent : TDisplayServerEvent {
  };

  interface TDisplayServerDragEvent : TDisplayServerEvent {
    readonly attribute double x0;          // not sorted!
    readonly attribute double x1;
    readonly attribute double y0;
    readonly attribute double y1;
  };

  interface TDisplayServerSliderChangedEvent : TDisplayServerEvent {
    readonly attribute long sliderNr;      // starting with 0
    readonly attribute double sliderValue;
  };

  // Interfaces of Display Server
```



```

interface TDisplayManager {
    TDisplayServer createDisplayServer ();
    string getDisplayName ();
    void notifyDisplayServerShutdown (in TDisplayServer ds);
    void closeAll ();
};

interface TDisplayServer {
    void setDisplayManager (in TDisplayManager dm);
    void setPlot (in TPlot plot);
    void close ();

    // event handling
    void notifyClosed ();
    void notifyMouseDragged (in double x0, in double y0,
                             in double x1, in double y1);
    void performZoomOnDragging (in boolean f);
    void displayRectOnDragging (in boolean f);
    void performRepaintOnZoom (in boolean f);
    void openSlider (in string titel, in stringseq name,
                    in doubleseq minval, in doubleseq maxval,
                    in doubleseq startval,
                    in doubleseq mineditval,
                    in doubleseq maxeditval);
    void setCoordinates (in double x0, in double y0,
                        in double x1, in double y1);
    void setAxesDisplay (in boolean f);
    void clear ();

    void drawArc (in double x0, in double y0, in double x1,
                 in double y1, in long startAngle,
                 in long arcAngle);
    void drawArcC (in double x0, in double y0, in double x1,
                  in double y1, in long startAngle,
                  in long arcAngle, in TColor pColor);
    void drawArcL (in double x0, in double y0, in double x1,
                  in double y1, in long startAngle,
                  in long arcAngle, in octet pLineStyleID);
    void drawArcCL(in double x0, in double y0, in double x1,
                  in double y1, in long startAngle,
                  in long arcAngle, in TColor pColor,
                  in octet pLineStyleID);
};

```

```

void fillArc (in double x0, in double y0, in double x1,
             in double y1, in long startAngle,
             in long arcAngle);
void fillArcC (in double x0, in double y0, in double x1,
              in double y1, in long startAngle,
              in long arcAngle, in TColor pColor);
void drawCircle (in double x0, in double y0, in double r);
void drawCircleC (in double x0, in double y0, in double r,
                 in TColor pColor);
void drawCircleL (in double x0, in double y0, in double r,
                 in octet pLineStyleID);
void drawCircleCL(in double x0, in double y0, in double r,
                 in TColor pColor, in octet pLineStyleID);
void fillCircle (in double x0, in double y0, in double r);
void fillCircleC (in double x0, in double y0, in double r,
                 in TColor pColor );
void drawEllipse (in double x0, in double y0, in double x1,
                 in double y1);
void drawEllipseC (in double x0, in double y0, in double x1,
                  in double y1, in TColor pColor);
void drawEllipseL (in double x0, in double y0, in double x1,
                  in double y1, in octet pLineStyleID);
void drawEllipseCL(in double x0, in double y0, in double x1,
                  in double y1, in TColor pColor,
                  in octet pLineStyleID);
void fillEllipse (in double x0, in double y0, in double x1,
                 in double y1);
void fillEllipseC (in double x0, in double y0, in double x1,
                  in double y1,
                  in TColor pColor);
void drawLine (in double x0, in double y0, in double x1,
              in double y1);
void drawLineC (in double x0, in double y0, in double x1,
               in double y1, in TColor pColor);
void drawLineL (in double x0, in double y0, in double x1,
               in double y1, in octet pLineStyleID);
void drawLineCL(in double x0, in double y0, in double x1,
               in double y1, in TColor pColor,
               in octet pLineStyleID);
void drawPolygon (in doubleseq x, in doubleseq y);
void drawPolygonC (in doubleseq x, in doubleseq y,
                  in TColor pColor);
void drawPolygonL (in doubleseq x, in doubleseq y,

```

```

        in octet pLineStyleID);
void drawPolygonCL(in doubleseq x, in doubleseq y,
                  in TColor pColor, in octet pLineStyleID);
void fillPolygon (in doubleseq x, in doubleseq y);
void fillPolygonC (in doubleseq x, in doubleseq y,
                  in TColor pColor);
void drawPolyLine (in doubleseq x, in doubleseq y);
void drawPolyLineC (in doubleseq x, in doubleseq y,
                   in TColor pColor);
void drawPolyLineL (in doubleseq x, in doubleseq y,
                   in octet pLineStyleID);
void drawPolyLineCL(in doubleseq x, in doubleseq y,
                   in TColor pColor, in octet pLineStyleID);
void drawRect (in double x0, in double y0,
              in double x1, in double y1);
void drawRectC (in double x0, in double y0,
               in double x1, in double y1,
               in TColor pColor);
void drawRectL (in double x0, in double y0,
               in double x1, in double y1,
               in octet pLineStyleID);
void drawRectCL(in double x0, in double y0,
               in double x1, in double y1,
               in TColor pColor, in octet pLineStyleID);
void fillRect (in double x0, in double y0,
              in double x1, in double y1);
void fillRectC (in double x0, in double y0,
               in double x1, in double y1,
               in TColor pColor);
void drawRoundRect (in double x0, in double y0,
                   in double x1, in double y1,
                   in double arcWidth,
                   in double arcHeight);
void drawRoundRectC (in double x0, in double y0,
                    in double x1, in double y1,
                    in double arcWidth, in double arcHeight,
                    in TColor pColor);
void drawRoundRectL (in double x0, in double y0,
                    in double x1, in double y1,
                    in double arcWidth, in double arcHeight,
                    in octet pLineStyleID);
void drawRoundRectCL(in double x0, in double y0,
                    in double x1, in double y1,

```

```

        in double arcWidth , in double arcHeight ,
        in TColor pColor , in octet pLineStyleID);
void fillRoundRect (in double x0 , in double y0 , in double x1 ,
        in double y1 , in double arcWidth ,
        in double arcHeight);
void fillRoundRectC (in double x0 , in double y0 , in double x1 ,
        in double y1 , in double arcWidth ,
        in double arcHeight ,
        in TColor pColor);
void drawSquare (in double x0 , in double y0 , in double a);
void drawSquareC (in double x0 , in double y0 , in double a ,
        in TColor pColor);
void drawSquareL (in double x0 , in double y0 , in double a ,
        in octet pLineStyleID);
void drawSquareCL(in double x0 , in double y0 , in double a ,
        in TColor pColor , in octet pLineStyleID);
void fillSquare (in double x0 , in double y0 , in double a);
void fillSquareC (in double x0 , in double y0 , in double a ,
        in TColor pColor);
void drawString (in double x , in double y , in string s);
void drawStringC (in double x , in double y , in string s ,
        in TColor pColor);
void drawPoints (in doubleseq x , in doubleseq y ,
        in long thickness);
void drawPointsC (in doubleseq x , in doubleseq y ,
        in long thickness , in TColor pColor );
void drawEmptyPoints (in doubleseq x , in doubleseq y ,
        in long thickness);
void drawEmptyPointsC(in doubleseq x , in doubleseq y ,
        in long thickness , in TColor pColor);
void drawESQPoints (in doubleseq x , in doubleseq y ,
        in long thickness);
void drawESQPointsC (in doubleseq x , in doubleseq y ,
        in long thickness , in TColor pColor);

void paintDisplay ();
};

```

```
// Interfaces of Object Server
```

```
interface TPlot {
    void setDisplayServer (in TDisplayServer ds);
    TDisplayServer getDisplayServer ();
};

```

```

void close ();
void notifyDisplayServerEvent (in TDisplayServerEvent ev);
void registerClientCallback (in TClientCallback cb);
void removeClientCallback ();
void openSlider (in string titel , in stringseq name,
                in doubleseq minval , in doubleseq maxval,
                in doubleseq startval , in doubleseq mineditval ,
                in doubleseq maxeditval);

void draw ();
void clear ();
void appendObject (in TGraphObject go);
void appendColor (in TColor col);
void drawCoordinate(in boolean f);
};

interface TPlot2D : TPlot {
    void setCoordinates (in double x0, in double y0,
                        in double x1, in double y1);
    void performZoomOnDragging (in boolean f);
    void performRepaintOnZoom (in boolean f);
};

interface TPlot3D : TPlot {
    void setCoordinates (in double x0, in double y0, in double z0,
                        in double x1, in double y1, in double z1);
    void project (in double x, in double y, in double z,
                 out double xs, out double ys);
};

interface TGraphObject {
    TGraphObject cloneObject ();
    void setColor (in TColor pcol);
    void destroy ();
    string name ();
};

interface TGraphObject2D : TGraphObject {
    void draw (in TDisplayServer ds);
};

interface TGraphObject3D : TGraphObject {
    void draw (in TDisplayServer ds, in TPlot3D plot);
};

```

```

// subinterfaces of TGraphObject2D
interface TCircle : TGraphObject2D {
    void setLineStyle (in octet pLineStyle);
    void setData (in double x, in double y, in double r);
};

interface TFullCircle : TGraphObject2D {
    void setData (in double x, in double y, in double r);
};

interface TText : TGraphObject2D {
    void setData (in double x, in double y, in string s);
};

interface TPoint : TGraphObject2D {
    void setData (in double x, in double y, in long thickness0);
};

interface TLine : TGraphObject2D {
    void setLineStyle (in octet pLineStyle);
    void setData (in double x0, in double y0,
                 in double x1, in double y1);
};

interface TPolyLine : TGraphObject2D {
    void setLineStyle (in octet pLineStyle);
    void setData (in doubleseq x, in doubleseq y);
};

interface TScatterplot : TGraphObject2D {
    void setData (in doubleseq x, in doubleseq y,
                 in long thickness0);
};

interface TBoxPlot2D : TGraphObject2D {
    void setData(in doubleseq x);
};

// subinterfaces of TGraphObject3D
interface TPoint3D : TGraphObject3D {
    void setData (in double x, in double y, in double z,
                 in long thickness0);
};

```

```

};

interface TPolyLine3D : TGraphObject3D {
    void setLineStyle (in octet pLineStyle);
    void setData (in doubleseq x, in doubleseq y,
                 in doubleseq z);
};

interface TWireframe3D : TGraphObject3D {
    void setLineStyle (in octet pLineStyle);
    void setData (in doubleseq x, in doubleseq y,
                 in doublefield z);
};

// represents a Color in the RGBa - format
// all values in range [0..255]
// alpha: value 0 = completely transparent ,
// value 255 = opaque
interface TColor {
    void setAlpha (in short alpha);
    void setR(in short red);
    void setG(in short green);
    void setB(in short blue);
    void setRGB(in short red , in short green ,
               in short blue);
    readonly attribute short red;
    readonly attribute short green;
    readonly attribute short blue;
    readonly attribute short alpha;
    long getRGB(); // returns rgb-Value with alpha=255(opaque)
    long getARGB(); // returns argb-Value
    void destroy ();
};

interface TGraphFactory {
    TPlot2D createPlot2D ();
    TPlot3D createPlot3D ();
    TGraphObject createObject (in string name);
    void registerPrototype (in TGraphObject prototype);
    TColor createColor (in TPlot plot , in long r , in long g ,
                      in long b , in long alpha);
    TColor createColorByName (in TPlot plot , in string name);
    TColor createColorByNr (in TPlot plot , in long nr);
};

```

```

};

interface TDisplayManagerPool {
    void registerDisplayManager (in TDisplayManager dm);
    void unregisterDisplayManager (in TDisplayManager dm);
    TGraphFactory getGraphFactory ();
    boolean connect (in TPlot plot , in string displayname);
};

// Interfaces of Client Program

interface TClientCallback {
    void notifyDisplayServerEvent (in TPlot plot ,
                                   in TDisplayServerEvent ev);
};

};

```


A.4 Technische Details zum Prototypen

In diesem Abschnitt wird ein Prototyp der Client/Server-Architektur vorgestellt, der seine Anwendung in der Finanzmathematik findet. Der Prototyp lädt z. B. den Dax als Datensatz, gibt die Kursentwicklung, die Renditen und die Dichten für verschiedene Schätzer in jeweils einem Applet aus. In dem letzten Applet wird die Kerndichte der Dax-Renditen mit der Normalverteilungsdichte und Studentschen Dichte geplottet. Die Parameter wurden mit dem Maximum-Likelihood-Schätzer geschätzt. Über einen Slider kann der Betrachter die Bandbreite der Kerndichte ändern und diese somit an die anderen beiden Dichten anpassen. Das Applet visualisiert den Stylized Fact 4.

Wie schon erwähnt wird zur Kommunikation die Middleware CORBA verwendet. Der Objekt-Server wurde in Java und C++ implementiert. Es wurden die Middleware JacORB für Java und Mico für C++ verwendet. Der Display-Server wurde in ein Applet eingebunden, welches über ein PHP-Skript auf einem Web-Server aufgerufen werden kann. Ebenso wurde der Prototyp mit der in dem JDK1.5.0 integrierten Middleware umgesetzt. Der Prototyp ist durch die verwendete Middleware sehr stark an die vorhandenen Browser gebunden.

Der Prototype basierend auf dem JacORB funktioniert nur mit einem JDK1.3 Plugin im Browser Mozilla. Dabei müssen auch die Sicherheitseinstellung des Java-Plugins richtig gesetzt werden. Wird das JDK1.5.0 verwendet, so müssen im Quelltext des Prototypen einigen Änderungen gemacht werden. Außerdem müssen wieder spezielle Sicherheitsstellungen¹⁴ bzgl. Java gemacht werden. Bei der Verwendung des JDK1.5.0 wird die URL im Browser nicht in einer IP aufgelöst. Daher kann es leicht zu Kommunikationsproblemen kommen und die Architektur stürzt ab.

Es wird jetzt kurz vorgestellt, wie die einzelnen Komponenten der Client/Server-Architektur gestartet werden. Der Objekt-Server wird als erstes gestartet.

Befehl zum Starten des Objekt-Servers

- **JacORB:** ./myjaco ObjectServer
- **JDK1.5.0:** ./java ObjectServer

¹⁴Java-Policy

Dieser wartet auf die Anfragen des Display-Servers. Der Display-Server wird mit dem Aufruf des PHP-Skriptes gestartet. Das Applet wird bei Verwendung des JacORB mit dem Plugin der Java-Version 1.3 übersetzt. Bei dem JDK1.5.0 ist es das Plugin dieser Version, welches Verwendung findet. Nach dem Starten des Applets wird durch dieses der Client gestartet. Er kann ebenso von Hand gestartet werden, in diesem Fall bekommt er eindeutige Referenznummer übergeben, mit der er den Display-Server eindeutig identifizieren kann.

Befehl zum Starten eine Clients

- **JacORB:** `./myjaco ClientUnivariate2 555`
- **JDK1.5.0:** `./java ClientUnivariate2 555`

Sobald der Client gestartet wurde und die Berechnungen erfolgt sind, wird die erste Ausgabe sichtbar. Dies kann zum Beispiel wie folgt aussehen:

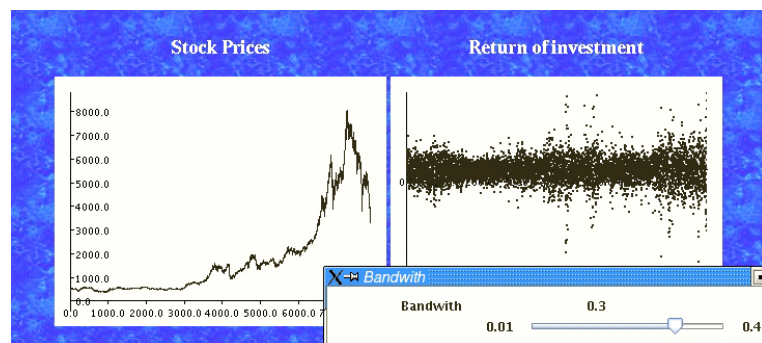


Abbildung A.6: Screenshot des Prototypen

Anhang B

Finanzmathematischer Teil

B.1 Mathematische Grundlagen für die Finanzdatenanalyse

B.1.1 Allgemeine mathematische Grundlagen

Definition 36 (Quantilfunktion, Reiss [71]). Sei X eine Zufallsvariable mit Verteilungsfunktion F . Dann ist die Quantilfunktion F^{-1} definiert als

$$F^{-1}(\alpha) = \inf\{x | F(x) \geq \alpha\}, \quad (\text{B.1})$$

mit $\alpha \in (0, 1)$. Es gilt

- Für jede Gleichverteilung $U \sim U(0, 1)$ gilt dass $F^{-1}(U) \sim F$. Durch diese Eigenschaft kann leicht eine Stichprobe zu einer Zufallsvariable mit Verteilungsfunktion F erzeugt werden.
- Wenn F stetig ist, dann ist die Zufallsvariable $F(U)$ gleichverteilt auf $[0, 1]$, es gilt $F(X) \sim U(0, 1)$.

Definition 37 (Korrelationskoeffizient, Fahrmeir et. al. [26]). Der Korrelationskoeffizient ist bestimmt durch

$$\rho = \rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}}.$$

X und Y seien quadrat-integrierbar.

Definition 38 (Stichprobenmittelwert). Für eine Stichprobe $\{x_i\}$, $i = 1, \dots, n$, ist der Stichprobenmittelwert (empirische Mittelwert) definiert als

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (\text{B.2})$$

Definition 39 (Stichprobenvarianz). Für eine Stichprobe $\{x_i\}$, $i = 1, \dots, n$ ist die Stichprobenvarianz (empirische Varianz) definiert als

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2. \quad (\text{B.3})$$

Einen erwartungstreuen Schätzer für die Varianz erhält man mit der Formel

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{\mu})^2. \quad (\text{B.4})$$

Definition 40 (Dichte der Normalverteilung). Die Dichte der Normalverteilung mit Lokationsparameter μ und Skalenparameter $\sigma > 0$ ist gegeben durch

$$\phi_{\mu, \sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (\text{B.5})$$

Die Dichte der Standard–Normalverteilung ist also $\phi(x)$ bzw. $\phi_{0,1}(x)$.

Bezeichnung: Wir schreiben für eine Zufallsvariable, die normal–verteilt mit Lokationsparameter μ und Skalenparameter σ^2 ist

$$\mathcal{L}(X) = N_{\mu, \sigma^2}.$$

Für die Standard–Normalverteilung wird geschrieben

$$\mathcal{L}(X) = N \text{ oder } \mathcal{L}(X) = N_{0,1}.$$

Eigenschaften der Normalverteilung: Sei $\mathcal{L}(X) = N_{\mu, \sigma^2}$, dann gilt

- $E(X) = \mu$,
- $Var(X) = \sigma^2$, $E(X^2) = \mu^2 + \sigma^2$,
- Schiefe = 0, Steilheit = 0.

- Für $r > 1$ gilt

$$E[(X - \mu)^r] = \begin{cases} 0 & \text{für } r \text{ ungerade} \\ 1 \cdot 3 \cdot \dots \cdot (r - 1)\sigma^r = \frac{\sigma^r r!}{2^r} \left(\frac{r}{2}\right)! & \text{für } r \text{ gerade} \end{cases}$$

Definition 41 (Gamma-Funktion). Die Gamma-Funktion mit Parameter λ ist definiert als

$$\Gamma(\lambda) = \int_0^\infty \exp(-x)x^{\lambda-1}dx, \text{ mit } \lambda > 0. \quad (\text{B.6})$$

Die Gamma-Funktion hat folgende Eigenschaften:

- $\Gamma(x + 1) = x\Gamma(x)$
- $\Gamma(1) = 1$

B.1.1.1 Sätze und Beweise

Definition 42 (Mischung). Sei $\{F_\vartheta\}$ eine Familie von Verteilungsfunktionen mit $\vartheta \in \Theta$ und h eine Dichte. Θ bezeichnet den Parameterraum. Dann ist

$$F_h(x) = \int F_\vartheta(x)h(\vartheta)d\vartheta \quad (\text{B.7})$$

eine neue Verteilungsfunktion, die als Mischung der Verteilungsfunktion F_ϑ bezüglich ϑ bezeichnet wird. Für die Dichte f_ϑ von F_ϑ gilt dem entsprechend

$$f_h(x) = \int f_\vartheta(x)h(\vartheta)d\vartheta \quad (\text{B.8})$$

Satz 10. Transformationssatz für Dichten ¹

Es gebe eine offene zusammenhängende Menge $M \subset \mathbb{R}^n$ so, dass für die Dichte $f(x)$ der Verteilung \mathcal{P}^X und den Zufallsvektor $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ die nachfolgenden Bedingungen erfüllt sind²:

1. Für $x \notin M$ ist $f(x) = 0$

¹entnommen aus Graf [37].

²Wir befinden uns in folgender Situation: $(\mathbb{R}^n, \mathcal{B}_n, \mathcal{P}^X) \xrightarrow{G} (\mathbb{R}^n, \mathcal{B}_n, \mathcal{P}^G)$.

2. Die Komponenten $G_j(x)$ von G sind auf M stetig partiell differenzierbar und es ist $J_G(x) \neq 0$ für alle $x \in M$, mit

$$J_G(x) = \det \begin{pmatrix} \frac{\partial G_1}{\partial x_1}(x) & \frac{\partial G_1}{\partial x_2}(x) & \cdots & \frac{\partial G_1}{\partial x_n}(x) \\ \frac{\partial G_2}{\partial x_1}(x) & \frac{\partial G_2}{\partial x_2}(x) & \cdots & \frac{\partial G_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G_n}{\partial x_1}(x) & \frac{\partial G_n}{\partial x_2}(x) & \cdots & \frac{\partial G_n}{\partial x_n}(x) \end{pmatrix}$$

als **Funktionaldeterminante**.

3. Ist $M^* = G(M) = \{y \in \mathbb{R}^n : y = G(x) \text{ mit } x \in M\}$ das Bild der Menge M unter G , so ist die Abbildung $G : M \rightarrow M^*$ bijektiv mit der Umkehrabbildung $G^{-1} : M^* \rightarrow M$.

Dann besitzt die Verteilung P^G des Zufallsvektors G die Dichte

$$g(y) = \begin{cases} f(G^{-1}(y)) \frac{1}{|J_G(G^{-1}(y))|} & \text{falls } y \in M^* \\ 0 & \text{sonst} \end{cases} \quad (\text{B.9})$$

Beweis. Siehe Graf [37] Kapitel 13. □

Definition 43 (Langsame Variation). Eine positive, Lebesgue-messbare Funktion L auf $(0, \infty)$ variiert langsam auf ∞ , wenn

$$\lim_{x \rightarrow \infty} \frac{L(tx)}{x} = 1, \quad t > 0. \quad (\text{B.10})$$

Satz 11 (Maximum-Anziehungsbereich (MDA) der Fréchet-Verteilung). Die Verteilungsfunktion F befindet sich genau dann im Maximum-Anziehungsbereich der Fréchet-Verteilung Φ_α ($\alpha > 0$), wenn

$$\bar{F}(x) = x^{-\alpha} L(x) \quad (\text{B.11})$$

für eine langsam variiierende Funktion L gilt, wobei $\bar{F}(x) = 1 - F(x)$. Ist $F \in MDA(\Phi_\alpha)$, dann

$$\frac{M_n}{F^{-1}(1 - n^{-1})} \rightarrow \Phi_\alpha, \quad (\text{B.12})$$

wobei $M_n = \max(X_1, \dots, X_n)$ ist.

Beweis. Siehe Embrechts et. al. [23], Seite 131. □

Es folgt der Beweis des Satzes, dass die Studentsche Verteilung im Anziehungsbereich der Fréchet–Verteilung liegt.

Beweis: MDA der Studentschen Verteilung. Sei $f_{\alpha,\mu,\sigma}$ die Studentsche Dichte der Form

$$f_{\alpha,\mu,\sigma}(x) = \frac{\Gamma((\alpha+1)/2)}{\Gamma(\alpha/2)\Gamma(1/2)\sigma} \left(1 + \frac{(x-\mu)^2}{\alpha\sigma^2}\right)^{-(\alpha+1)/2}$$

Ohne Beschränkung der Allgemeinheit sei $\mu = 0$ und $\sigma = 1$. Dann hat die Dichte folgende Form

$$f_{\alpha}(x) = c(\alpha) \left(1 + \frac{x^2}{\alpha}\right)^{-(\alpha+1)/2}$$

mit

$$c(\alpha) = \frac{\Gamma((\alpha+1)/2)}{\Gamma(\alpha/2)\Gamma(1/2)}$$

Mit dem Satz von de l'Hospital folgt

$$\begin{aligned} \lim_{x \rightarrow \infty} \frac{1 - F(x)}{c(\alpha)\alpha^{(\alpha-1)/2}x^{-\alpha}} &= \lim_{x \rightarrow \infty} \frac{f(x)}{c(\alpha)\alpha^{(\alpha+1)/2}x^{-(\alpha+1)}} \\ &= \lim_{x \rightarrow \infty} \frac{\left(1 + \frac{x^2}{\alpha}\right)^{-(\alpha+1)/2}}{\alpha^{(\alpha+1)/2}x^{-(\alpha+1)}} \\ &= \lim_{x \rightarrow \infty} \frac{\alpha^{-(\alpha+1)/2}x^{\alpha+1}}{\left(1 + \frac{x^2}{\alpha}\right)^{(\alpha+1)/2}} \\ &= \lim_{x \rightarrow \infty} \frac{\alpha^{-(\alpha+1)/2}x^{\alpha+1}}{\left(\frac{x^2}{\alpha} \left(\frac{n}{x^2} + 1\right)\right)^{(\alpha+1)/2}} \\ &= \lim_{x \rightarrow \infty} \frac{\alpha^{-(\alpha+1)/2}x^{\alpha+1}}{\alpha^{-(\alpha+1)/2}x^{\alpha+1} \left(\frac{n}{x^2} + 1\right)^{(\alpha+1)/2}} \\ &= \lim_{x \rightarrow \infty} \frac{1}{\left(\frac{n}{x^2} + 1\right)^{(\alpha+1)/2}} = 1. \end{aligned}$$

Mit dem obigen Satz und den Bedingungen $\alpha > 0$ und $c(\alpha)\alpha^{-(\alpha-1)/2} > 0$ gilt, dass $F \in MDA(\Phi_{\alpha})$, d. h. die Studentsche Verteilung besitzt schwere Flanken.

□

Satz 12 (Ungleichung 1). *Es gilt folgende Ungleichung mit*

$$1 - \frac{x+y}{2} \leq \frac{1}{((1+x)(1+y))^{1/2}},$$

für $x, y \in [0, \infty)$.

Beweis. Der Beweis der Ungleichung kann [72] entnommen werden. □

Satz 13 (Ungleichung 2). *Es gilt für die Ungleichung $x, y \in [0, \infty)$*

$$1 - (x + y + xy) \leq \frac{1}{(1+x)(1+y)}.$$

Beweis.

Es gelte immer $x, y \in [0, \infty)$. Für den Logarithmus folgt

$$\begin{aligned} \log(1+z) &= \frac{z}{1+\vartheta z}, \vartheta \in \mathbb{R}^+ \\ &\leq z \end{aligned} \tag{B.13}$$

Damit folgt

$$\begin{aligned} \log(1+x+y+xy) &\leq x+y+xy \\ \Leftrightarrow -\log(1+x+y+xy) &\geq -(x+y+xy). \end{aligned} \tag{B.14}$$

Damit kann die Ungleichung bewiesen werden, es gilt

$$\begin{aligned} \exp(-\log((1+x)(1+y))) &\geq \exp(-(x+y+xy)) \\ &\geq 1 - (x+y+xy). \end{aligned} \tag{B.15}$$

□

B.1.1.2 Grundlagen für Zeitreihen

Im folgenden Abschnitt werden die Definition einer Zeitreihe sowie wichtige Definitionen und Resultate angegeben, die für den finanzmathematischen Teil benötigt werden³.

Definition 44 (Stochastischer Prozess). *Ein Stochastischer Prozess ist eine Familie von Zufallsvariablen $\{X_t, t \in T\}$ definiert auf dem Wahrscheinlichkeitsraum (Ω, \mathcal{F}, P) .*

³Als Grundlage diente das Buch von Brockwell und Davis [14].

Definition 45 (Zeitreihe). Eine Zeitreihe ist ein Stochastischer Prozess $\{X_t, t \in T\}$, wobei T eine Menge von Zeitpunkten ist, zum Beispiel $\{0, \pm 1, \pm 2, \pm 3, \dots\}$, $\{1, 2, 3, \dots\}$, $[0, \infty)$ oder $(-\infty, \infty)$.

Definition 46 (Autokovarianzfunktion). Wenn $\{X_t, t \in T\}$ ein stochastischer Prozess ist mit $\text{Var}(X_t) < \infty$ für jedes $t \in T$, dann ist die Autokorrelationsfunktion $\gamma_X(\cdot, \cdot)$ von $\{X_t\}$ definiert durch

$$\gamma_X(r, s) = \text{Cov}(X_r, X_s) = E[(X_r - EX_r)(X_s - EX_s)], \quad r, s \in T.$$

Definition 47 (Stationarität). Die Zeitreihe $\{X_t, t \in \mathbb{Z}\}$, mit der Indexmenge $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$ ist stationär, wenn

1. $E|X_t|^2 < \infty$ für alle $t \in \mathbb{Z}$,
2. $EX_t = m$ für alle $t \in \mathbb{Z}$ und
3. $\gamma_X(r, s) = \gamma_X(r + t, s + t)$ für alle $r, s, t \in \mathbb{Z}$

gilt.

Definition 48 (Autokorrelationsfunktion). Die Autokorrelationsfunktion (ACF) des stochastischen Prozesses $\{X_t, t \in T\}$ mit $\text{Var}(X_t) < \infty$ für jeder $t \in T$ ist gegeben durch

$$\rho_X(h) := \gamma_X(h) / \gamma_X(0), \quad h \in T.$$

Dies gilt nur, wenn der stochastische Prozess $\{X_t, t \in T\}$ stationär ist, denn dann kann $\gamma_X(h, 0)$ geschrieben werden als $\gamma_X(h) = \text{Cov}(X_{t+h}, X_t)$, $t, h \in \mathbb{Z}$.

Definition 49 (Strenge Stationarität). Eine Zeitreihe $\{X_t, t \in \mathbb{Z}\}$ heißt streng stationär, wenn für jede Verteilung der $(X_{t_1}, \dots, X_{t_k})$ und $(X_{t_1+h}, \dots, X_{t_k+h})$ gilt, dass sie die gleichen Verteilungen haben für alle $k \in \mathbb{N}$ und für alle $t_1, \dots, t_k, h \in \mathbb{Z}$. Diese Definition ist äquivalent dazu, dass (X_1, \dots, X_k) und $(X_{1+h}, \dots, X_{k+h})$ die gleichen Verteilung haben für alle $k \in \mathbb{N}$ und $h \in \mathbb{Z}$.

B.1.2 Schätzer und Test's

Definition 50 (Schätzfunktion, Schätzstatistik , Fahmeir et. al. [26]).

Eine Schätzfunktion oder Schätzstatistik für den Grundgesamtheitsparameter ψ ist eine Funktion

$$T = g(X_1, \dots, X_n)$$

der Stichprobenvariablen X_1, \dots, X_n . Der aus den Realisationen x_1, \dots, x_n resultierende numerische Wert

$$g(x_1, \dots, x_n)$$

ist der zugehörige Schätzwert.

B.1.2.1 EM–Schätzer (EM–Algorithmus)

Der EM–Algorithmus⁴ findet seine Anwendungen bei Schätzungen von Parametern mit nicht vollständigen Daten. Er wird außerdem auch da angewendet, wo der Maximum–Likelihood–Schätzer keine explizite Lösung hat, dabei müssen meistens die Lösungen komplexer Funktionsgleichungen mit der Newton–Raphson Methode bestimmt werden. In diesem Fall wird aus den vollständigen Daten eine Problemstellung mit nicht vollständigen Daten definiert. Diese Vorgehensweise wird auch bei der Schätzung der Parameter der Studentschen Verteilung angewandt.

Den Ausgangspunkt bilden, wie bei jeder Schätzung, die beobachteten Realisationen $\mathbf{y} = (y_1, \dots, y_n)^T$ des Zufallsvektors Y . Y soll die Dichte $g(y, \psi)$ besitzen, wobei $\psi = (\psi_1, \dots, \psi_d)^T$ der Parametervektor mit Parameterraum Ω ist. Es wird nun angenommen, dass die Realisationen \mathbf{y} nicht vollständig sind, d. h. es fehlen nicht beobachtbare Daten, die durch den Vektor \mathbf{z} ausgedrückt werden. Die kompletten, vollständigen Daten werden durch den Vektor \mathbf{x} ($\mathbf{x} = (\mathbf{y}^T, \mathbf{z}^T)^T$) repräsentiert. Z und X bezeichnen dabei die zu den Daten zugehörigen Zufallsvektoren. $g_c(x, \psi)$ sei die Dichte, die dem Zufallsvektor X zu Grunde liegt. Somit ergibt sich für die vollständigen Daten die log–Likelihood–Funktion

$$\log L_c(\psi) = \log g_c(x, \psi). \tag{B.16}$$

Der EM–Algorithmus löst nicht diese Gleichung, sondern die log–Likelihood Gleichung der nicht vollständigen Daten, indem er die log–Likelihood–Funktion ($\log L_c(x, \psi)$) der kompletten Daten benutzt. Weil sie aber nicht beobachtbare Daten enthält, werden diese fehlenden Daten durch den bedingten Erwartungswert

⁴McLachlan und Kristnan [62], S.21 ff. .

gegeben y ersetzt. Bei dieser Berechnung werden die momentanen Parameter von ψ benutzt, die iterativ bestimmt werden.

Der EM-Algorithmus setzt sich aus zwei Schritten zusammen, dem Erwartungsschritt (E-Schritt)⁵ und dem Maximierungsschritt (M-Schritt)⁶. Im E-Schritt werden die fehlenden Daten z über die bedingte Erwartung gegeben der beobachtbaren Daten y bestimmt. Für die $(k+1)$ -te Iteration gilt dann

$$Q(\psi; \psi^{(k)}) = E_{\psi^{(k)}} \{ \log L_c(\psi) | y \} \quad (\text{B.17})$$

Damit der EM-Algorithmus konvergiert, muss ein passender Startschätzer $\psi^{(0)}$ für ψ gefunden werden. Nach dem E-Schritt wird dann der M-Schritt ausgeführt. Im M-Schritt wird $Q(\psi; \psi^{(k)})$ bezüglich ψ maximiert. Der neue Parameter $\psi^{(k+1)}$ wird so bestimmt, dass für alle $\psi \in \Omega$ gilt

$$Q(\psi^{(k+1)}; \psi^{(k)}) \geq Q(\psi; \psi^{(k)}). \quad (\text{B.18})$$

Die Maximierung wird bezüglich der vollständigen Daten durchgeführt. Diese beiden Schritte werden so oft hintereinander wiederholt, bis eine Abbruchbedingung erreicht worden ist. Diese kann durch die Differenz der Likelihood-Funktion ausgedrückt werden

$$L(\psi^{(k+1)}) - L(\psi^{(k)}) \leq \epsilon, \quad \epsilon > 0. \quad (\text{B.19})$$

Der EM-Algorithmus findet in dieser Arbeit seine Anwendung bei der Schätzung der Parameter der Studentschen Verteilung mit einem α .

B.1.3 Spezielle bivariate Copulas und deren Dichten

In diesem Abschnitt werden einige Copulas vorgestellt. Außerdem wird jeweils ein Contourplot und ein 3D-Plot der Copula angegeben. Anhand des Contourplots kann man die Abhängigkeitsstruktur der Copula erkennen.

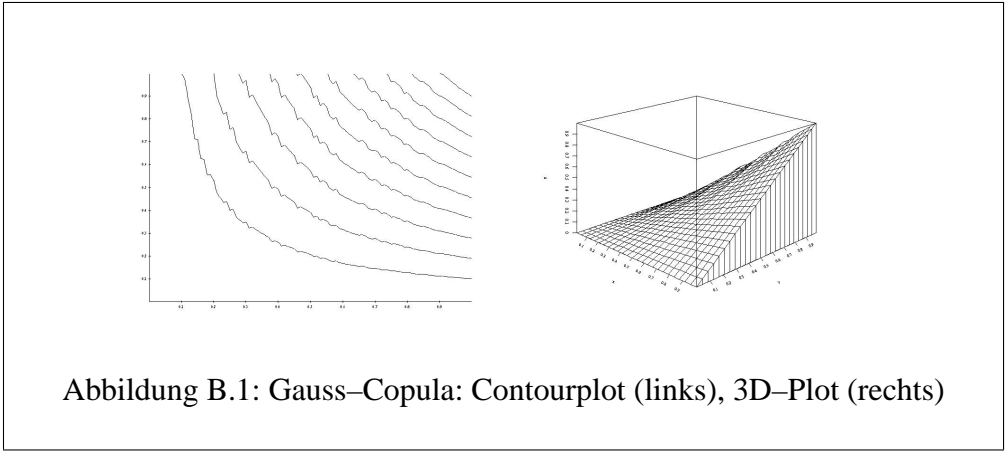
Definition 51 (Gauss-Copula). Für $u, v \in [0, 1]$ und $\rho \in [-1, 1]$ gilt

$$C^G(u, v) = \int_{-\infty}^{\phi^{-1}(u)} \int_{-\infty}^{\phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \mathbf{a}\right) dsdt, \quad (\text{B.20})$$

mit $\mathbf{a} = s^2 - 2\rho st + t^2$.

⁵engl. E-Step

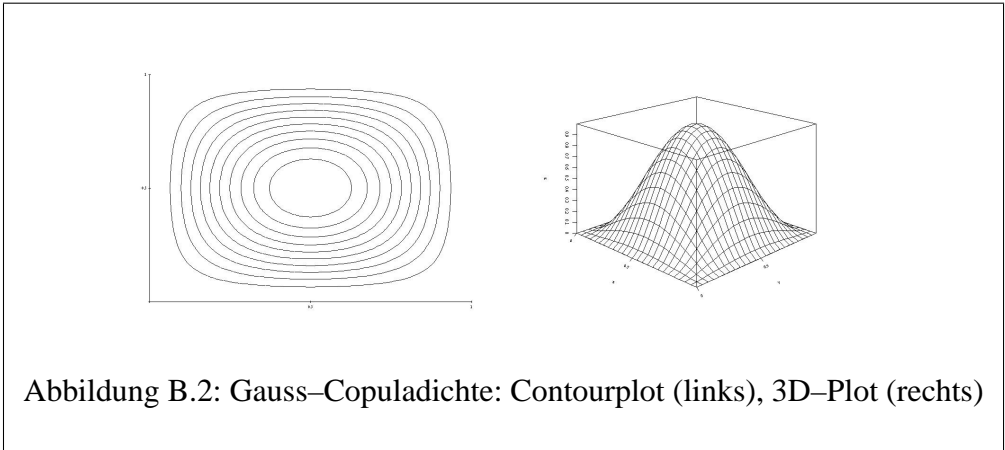
⁶engl. M-Step



Lemma 18 (Eigenschaften).

$$\rho = 2 \sin \left(\frac{\pi}{6} \rho_s \right), \tag{B.21}$$

$$\rho = \sin \left(\frac{\pi}{2} \tau \right), \tag{B.22}$$



Lemma 19 (Copuladichte der Gauss–Copula, Patton [67]). Für $u, v \in \mathbf{I} = [0, 1]$ und $\rho \in [-1, 1]$ gilt

$$c^G(u, v; \rho) = \frac{1}{\sqrt{1 - \rho^2}} \exp \left\{ -\frac{\Phi^{-1}(u)^2 + \Phi^{-1}(v)^2 - 2\rho\Phi^{-1}(u)\Phi^{-1}(v)}{2(1 - \rho^2)} \right\} \exp \left\{ -\frac{\Phi^{-1}(u)^2 + \Phi^{-1}(v)^2}{2} \right\}, \quad (\text{B.23})$$

wobei $\Phi^{-1}(u)$ die Quantilfunktion der Standard–Normalverteilung ist. Für $(u, v) \notin [0, 1]^2$ ist $c^G(u, v; \rho)$ gleich null.

Beweis. Mit Satz 9 folgt:

$$\begin{aligned} \frac{\partial C(u, v; \rho)}{\partial u \partial v} &= \frac{\partial}{\partial u \partial v} \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1 - \rho^2}} \\ &\quad \cdot \exp \left(-\frac{1}{2(1 - \rho^2)} (s^2 - 2\rho st + t^2) \right) ds dt \\ &= \frac{1}{2\pi\sqrt{1 - \rho^2}} \\ &\quad \exp \left(-\frac{1}{2(1 - \rho^2)} (\Phi^{-1}(u)^2 - 2\rho\Phi^{-1}(u)\Phi^{-1}(v) + \Phi^{-1}(v)^2) \right) \\ &\quad \cdot \frac{\partial \Phi^{-1}(u)}{\partial u} \frac{\partial \Phi^{-1}(v)}{\partial v} \\ &= \frac{1}{2\pi\sqrt{1 - \rho^2}} \\ &\quad \exp \left(-\frac{1}{2(1 - \rho^2)} (\Phi^{-1}(u)^2 - 2\rho\Phi^{-1}(u)\Phi^{-1}(v) + \Phi^{-1}(v)^2) \right) \\ &\quad \cdot \frac{1}{\varphi(\Phi^{-1}(u) + \Phi^{-1}(v))} \end{aligned}$$

$$\begin{aligned}
\cdots &= \frac{1}{2\pi\sqrt{1-\rho^2}} \\
&\exp\left(-\frac{1}{2(1-\rho^2)}(\Phi^{-1}(u)^2 - 2\rho\Phi^{-1}(u)\Phi^{-1}(v) + \Phi^{-1}(v)^2)\right) \\
&\cdot \frac{1}{\frac{1}{\sqrt{2\pi}} \exp\{(\Phi^{-1}(u) + \Phi^{-1}(v))/2\}} \\
&= \frac{1}{\sqrt{1-\rho^2}} \\
&\exp\left(-\frac{1}{2(1-\rho^2)}(\Phi^{-1}(u)^2 - 2\rho\Phi^{-1}(u)\Phi^{-1}(v) + \Phi^{-1}(v)^2)\right) \\
&\cdot \exp\{-(\Phi^{-1}(u) + \Phi^{-1}(v))/2\}.
\end{aligned}$$

□

Definition 52 (Student–Copula). Für $u, v \in [0, 1]$ und $\alpha > 0, \rho \in [-1, 1]$ gilt

$$C^S(u, v; \alpha) = \int_{-\infty}^{t_\alpha^{-1}(u)} \int_{-\infty}^{t_\alpha^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} \left(1 + \frac{s^2 - 2\rho st + t^2}{\alpha(1-\rho^2)}\right)^{-(\alpha+2)/2} ds dt. \tag{B.24}$$

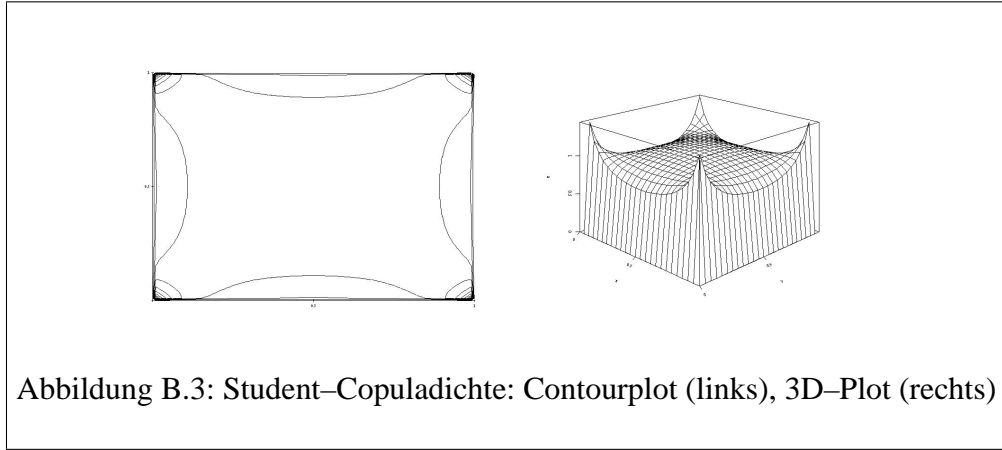


Abbildung B.3: Student–Copuladichte: Contourplot (links), 3D–Plot (rechts)

Lemma 20 (Copuladichte der Student–Copula).

$$c^S(u, v; \alpha, \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \left(1 + \frac{t_\alpha^{-1}(u)^2 - 2\rho t_\alpha^{-1}(u)t_\alpha^{-1}(v) + t_\alpha^{-1}(v)^2}{\alpha(1-\rho^2)}\right)^{-(\alpha+2)/2} \cdot \frac{1}{f_\alpha^S(t_\alpha^{-1}(u))f_\alpha^S(t_\alpha^{-1}(v))},$$

(B.25)

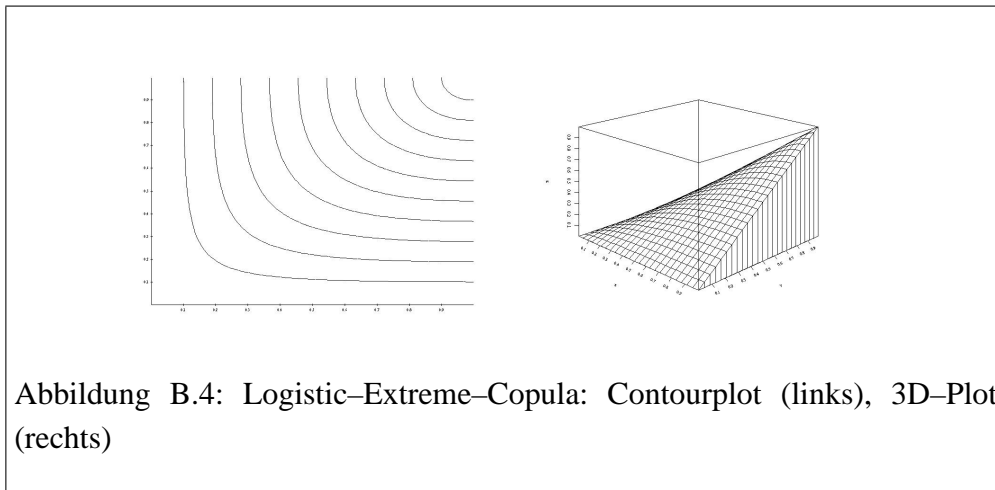
wobei t_α^{-1} die Quantilfunktion der Studentischen Verteilungen ist.

Beweis.

$$\begin{aligned} \frac{\partial C^S(u, v; \alpha)}{\partial u \partial v} &= \frac{\partial}{\partial u \partial v} \int_{-\infty}^{t_\alpha^{-1}(u)} \int_{-\infty}^{t_\alpha^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} \\ &\quad \left(1 + \frac{s^2 - 2\rho st + t^2}{\alpha(1-\rho^2)}\right)^{-(\alpha+2)/2} ds dt \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \\ &\quad \left(1 + \frac{t_\alpha^{-1}(u)^2 - 2\rho t_\alpha^{-1}(u)t_\alpha^{-1}(v) + t_\alpha^{-1}(v)^2}{\alpha(1-\rho^2)}\right)^{-(\alpha+2)/2} \\ &\quad \frac{1}{f_\alpha^S(t_\alpha^{-1}(u))f_\alpha^S(t_\alpha^{-1}(v))}. \end{aligned}$$

□

Bemerkung 9. Es gibt für die Student–Copula keine analytischen Formeln, die den Zusammenhang zwischen den Parametern der Copula und Kendall's τ und Spearman's ρ wiedergeben.



Definition 53 (Logistic–Extreme–Copula). Für $u, v \in [0, 1]$ und $\alpha \in [0, 1]$ ⁷ gilt

$$C^{LE}(u, v; \alpha) = \exp \left[- \left\{ (-\log(u))^{1/\alpha} + (-\log(v))^{1/\alpha} \right\}^\alpha \right]. \quad (\text{B.26})$$

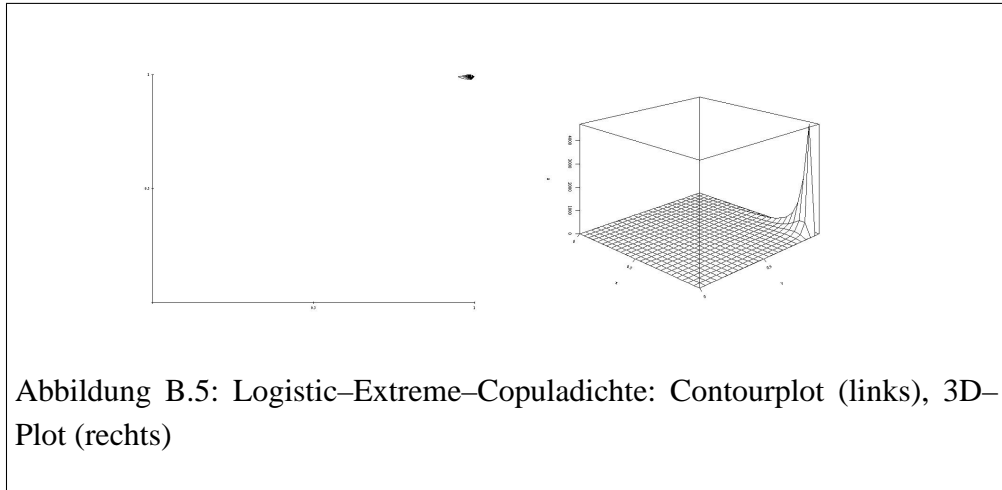


Abbildung B.5: Logistic–Extreme–Copuladichte: Contourplot (links), 3D–Plot (rechts)

⁷Für $\alpha = 1$ gilt Unabhängigkeit, für $\alpha \rightarrow 0$ folgt perfekte Abhängigkeit.

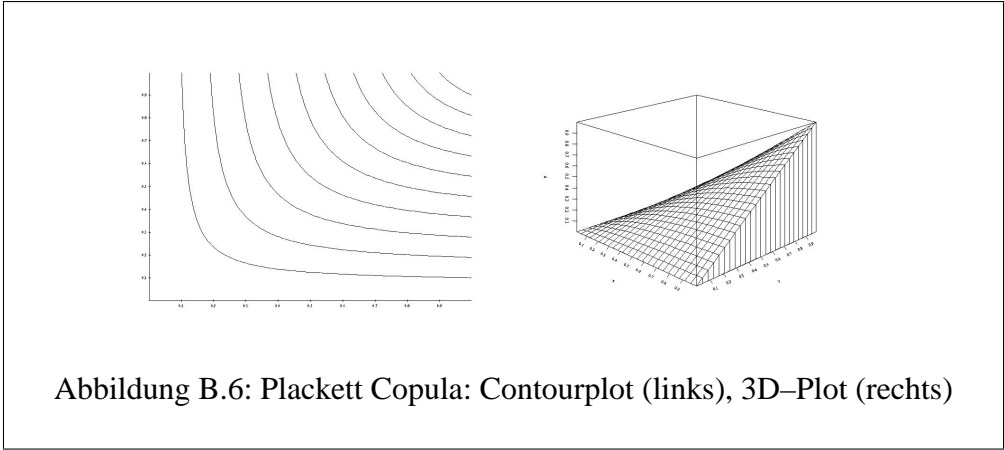
Lemma 21 (Copuladichte der Logistic–Extreme–Copula).

$$\begin{aligned}
c^{LE}(u, v; \alpha) &= \frac{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha (-\ln(u))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\}}{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u)} \\
&+ \frac{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha (-\ln(u))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\} (-\ln(v))^{1/\alpha}}{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u) \alpha} \\
&+ \frac{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^{2\alpha} (-\ln(u))^{1/\alpha} (-\ln(v))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\}}{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u)}. \tag{B.27}
\end{aligned}$$

Beweis.

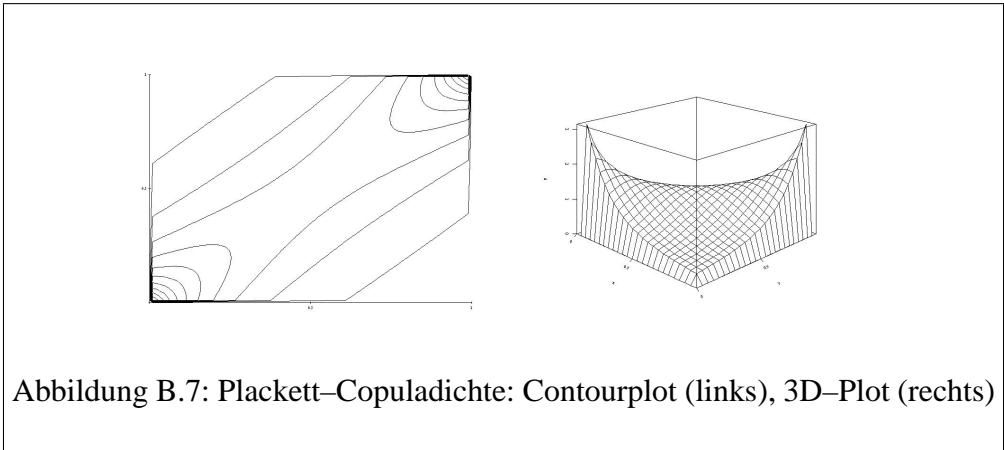
$$\begin{aligned}
\frac{\partial C^{LE}(u, v; \alpha)}{\partial u \partial v} &= \frac{\partial}{\partial u \partial v} \exp\left[-\{(-\log(u))^{1/\alpha} + (-\log(v))^{1/\alpha}\}^\alpha\right] \\
&= \frac{\partial}{\partial v} \frac{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha (-\ln(u))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\}}{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha (-\ln(u))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\}} \\
&= \frac{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u)}{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha (-\ln(u))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\}} \\
&+ \frac{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u) \alpha}{((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^{2\alpha} (-\ln(u))^{1/\alpha} (-\ln(v))^{1/\alpha} \exp\{-((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^\alpha\}} \\
&+ \frac{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u)}{v \ln(v) ((-\ln(u))^{1/\alpha} + (-\ln(v))^{1/\alpha})^2 u \ln(u)}.
\end{aligned}$$

□



Definition 54 (Plackett-Copula).

$$C^P(u, v; \psi) = \begin{cases} \frac{1+(\psi-1)(u+v) - \sqrt{(1+(\psi-1)(u+v))^2 - 4\psi(\psi-1)uv}}{2(\psi-1)} & \text{für } 0 \leq \psi \leq \infty, \psi \neq 1 \\ uv & \text{für } \psi = 1 \end{cases} \tag{B.28}$$



Lemma 22 (Plackett–Copuladichte).

$$c^P(u, v; \psi) = \begin{cases} \frac{\psi(1+(\psi-1)(u+v-2uv))}{\sqrt{([1+(\psi-1)(u+v)]^2 - 4\psi(\psi-1)uv)^3}} & \text{für } \psi \geq 0, \psi \neq 1 \\ 1 & \text{für } \psi = 1 \end{cases} \quad (\text{B.29})$$

Beweis. Mit Ableiten und Fallunterscheidung folgt:

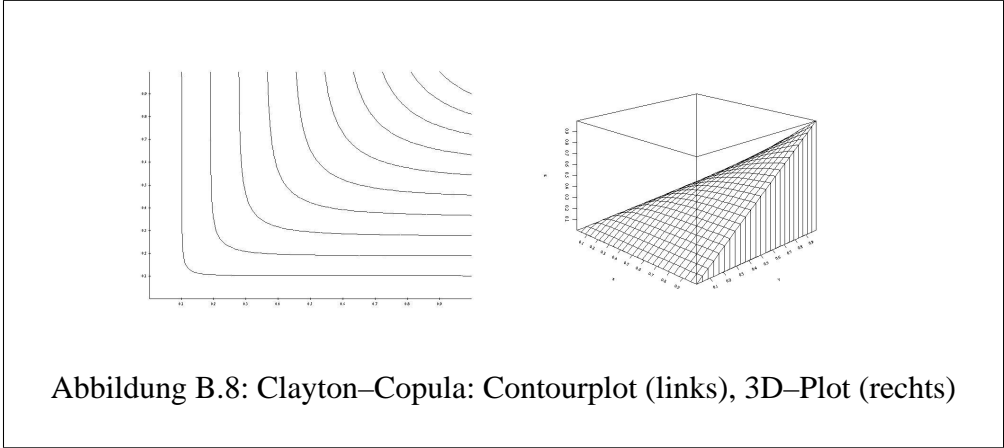
1. Fall: $\psi \geq 0, \psi \neq 1$

$$\begin{aligned} \frac{\partial C^P(u, v; \psi)}{\partial u \partial v} &= \frac{\partial}{\partial u \partial v} \frac{1 + (\psi - 1)(u + v) - \sqrt{(1 + (\psi - 1)(u + v))^2 - 4\psi(\psi - 1)uv}}{2(\psi - 1)} \\ &= \frac{\partial}{\partial u \partial v} \frac{1 + \psi u + \psi v - u - v}{2(\psi - 1)} \\ &= \frac{\sqrt{1 - 2u - 2v + 2\psi u + 2\psi v + \psi^2 u^2 - 2\psi u^2 + \psi^2 v^2 - 2\psi v^2 + u^2 + 2uv + v^2 - 2\psi^2 uv}}{2(\psi - 1)} \\ &= \frac{1}{2(\psi - 1)} \frac{\partial}{\partial v} \frac{-(\psi - 1) \sqrt{1 - 2u - 2v + 2\psi u + 2\psi v + \psi^2 u^2 - 2\psi u^2 + \psi^2 v^2 - 2\psi v^2 + u^2 + 2uv + v^2 - 2\psi^2 uv}}{-2 + 2\psi + 2\psi^2 u - 4\psi u + 2u + 2v - 2\psi^2 v} \\ &= \frac{(-2 + 2\psi + 2\psi^2 u - 4\psi u + 2u + 2v - 2\psi^2 v)(-2 + 2\psi + 2\psi u + 2\psi v + \psi^2 u^2 - 2\psi v^2 + u^2 + 2uv + v^2 - 2\psi^2 uv)}{4(1 - 2u - 2v + 2\psi u + 2\psi v + \psi^2 u^2 - 2\psi u^2 + \psi^2 v^2 - 2\psi v^2 + u^2 + 2uv + v^2 - 2\psi^2 uv)^{3/2}} \frac{2 - 2\psi^2}{2\sqrt{1 - 2u - 2v + 2\psi u + 2\psi v + \psi^2 u^2 - 2\psi u^2 + \psi^2 v^2 - 2\psi v^2 + u^2 + 2uv + v^2 - 2\psi^2 uv}} \\ &= \frac{\psi(1 + (\psi - 1)(u + v - 2uv))}{\sqrt{([1 + (\psi - 1)(u + v)]^2 - 4\psi(\psi - 1)uv)^3}} \cdot 2(\psi - 2) \end{aligned}$$

2. Fall: $\psi = 1$

$$\frac{\partial C(u, v; \psi)}{\partial u \partial v} = \frac{\partial}{\partial u \partial v} uv = 1$$

□

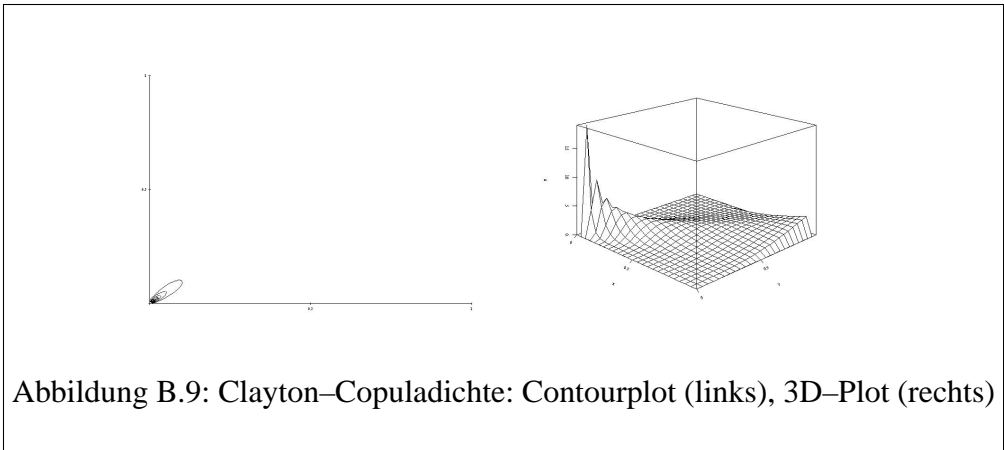


Definition 55 (Clayton-Copula).

$$C^{Cl}(u, v; \delta) = \begin{cases} (u^{-\delta} + v^{-\delta} - 1)^{-1/\delta} & , \delta > -1 \\ uv & , \delta = 1 \end{cases} \tag{B.30}$$

Lemma 23 (Eigenschaften).

$$\delta = \frac{2\tau}{1 - \tau}. \tag{B.31}$$



Lemma 24 (Clayton–Copuladichte).

$$c^{Cl}(u, v; \delta) = \begin{cases} (1 + \delta)(uv)^{-1-\delta}(u^{-\delta} + v^{-\delta} - 1)^{-2-1/\delta} & \text{für } \delta > -1 \\ 1 & \text{für } \delta = 0 \end{cases} . \quad (\text{B.32})$$

Beweis. Mit Ableiten und Fallunterscheidung folgt:

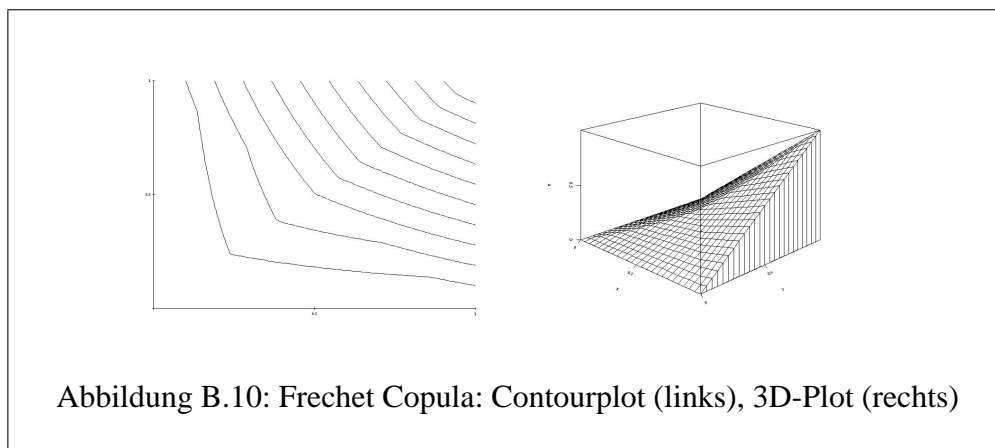
- 1. Fall: $\delta > 0$

$$\begin{aligned} \frac{\partial C^{Cl}(u, v; \delta)}{\partial u \partial v} &= \frac{\partial}{\partial u \partial v} (u^{-\delta} + v^{-\delta} - 1)^{-1/\delta} \\ &= \frac{\partial}{\partial v} \left([(u^{-\delta} + v^{-\delta} - 1)^{-1/\delta}]'_u \right) \\ &= \frac{\partial}{\partial v} (u^{-1-\delta} (u^{-\delta} + v^{-\delta} - 1)^{-1/\delta - 1}) \\ &= (\delta + 1)(uv)^{-1-\delta} (u^{-\delta} + v^{-\delta} - 1)^{-2-(1/\delta)}. \end{aligned}$$

- 2. Fall: $\delta = 0$

$$\frac{\partial C(u, v; \delta)}{\partial u \partial v} = \frac{\partial}{\partial u \partial v} uv = 1.$$

□



Definition 56 (Fréchet–Copula, Nelsen [65]). Für $\alpha, \beta \in [0, 1]$ mit $\alpha + \beta \leq 1$ ist die Fréchet–Copula definiert

$$C_{\alpha, \beta}^F(u, v) = \alpha M(u, v) + (1 - \alpha - \beta) \prod(u, v) + \beta W(u, v). \quad (\text{B.33})$$

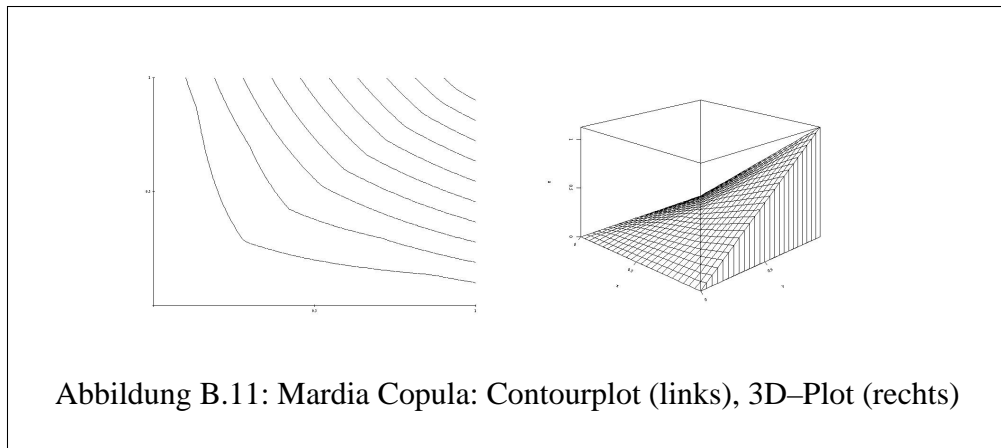
M, \prod, W sind definiert als

$$M(u, v) = \min(u, v), \quad (\text{B.34})$$

$$\prod(u, v) = uv, \quad (\text{B.35})$$

und

$$W(u, v) = \max(u + v - 1, 0). \quad (\text{B.36})$$



Definition 57 (Mardia–Copula, Nelsen [65]). Sei $\delta \in [-1, 1]$, dann ist

$$C_{\delta}^M = \frac{\delta^2(1 + \delta)}{2} M(u, v) + (1 - \delta^2) \prod(u, v) + \frac{\delta^2(1 - \delta)}{2} W(u, v). \quad (\text{B.37})$$

die Mardia–Copula. M, \prod, W wie oben.

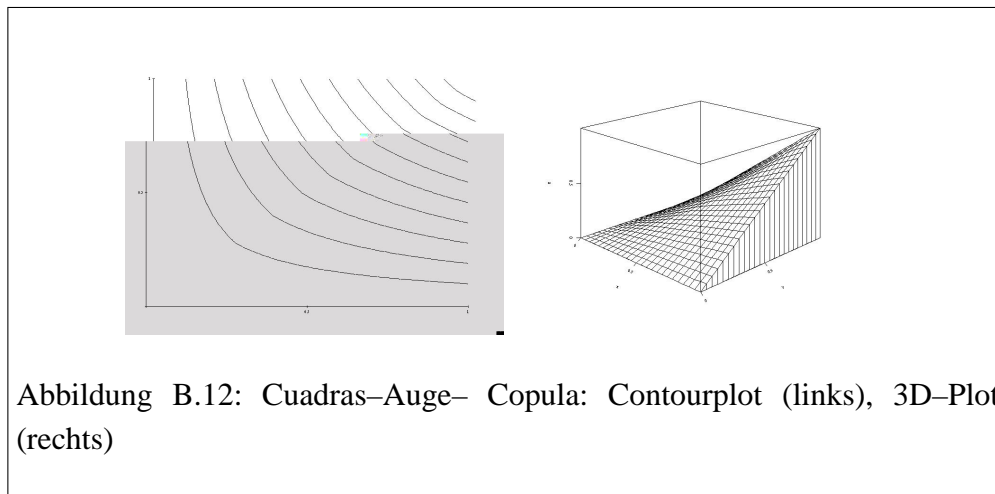


Abbildung B.12: Cuadras–Auge– Copula: Contourplot (links), 3D–Plot (rechts)

Definition 58 (Cuadras–Auge–Copula, Nelsen [65]). Für $\delta \in [0, 1]$ ist die Cuadras–Auge–Copula definiert durch

$$C^{Cu}(u, v) = (\min(u, v))^\delta (uv)^{1-\delta} = \begin{cases} uv^{1-\delta} & , u \leq v \\ vu^{1-\delta} & , v \leq u. \end{cases} \quad (\text{B.38})$$

Definition 59 (Bivariate Gumbel–Logistic–Copula , Nelsen[65]).

$$C^{GL}(u, v) = \frac{uv}{u + v - uv}. \quad (\text{B.39})$$

Lemma 25 (Copuladichte der bivariaten Gumbel–Logistic–Copula). Die Dichte der bivariaten Gumbel–Logistic–Copula ist gegeben durch

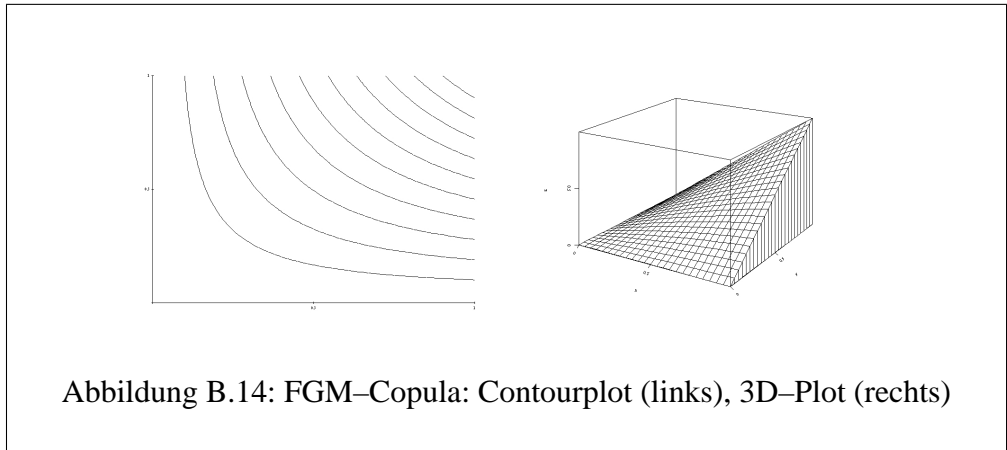
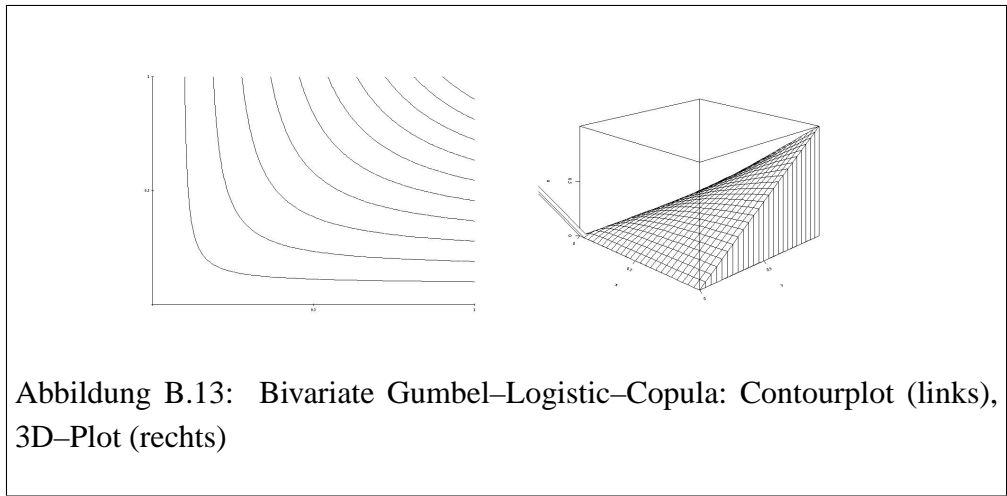
$$c^{GL}(u, v) = -2 \frac{uv}{(-u - v + uv)^3}. \quad (\text{B.40})$$

für $u, v \in [0, 1]$.

Beweis. Mit Satz 9 folgt

$$\begin{aligned} c^{GL}(u, v) &= \frac{\partial C(u, v)}{\partial u \partial v} = \frac{\partial}{\partial u \partial v} \frac{uv}{u + v - uv} \\ &= \frac{\partial}{\partial v} \frac{v^2}{(-u - v + uv)^2} = -2 \frac{uv}{(-u - v + uv)^3} \end{aligned}$$

□



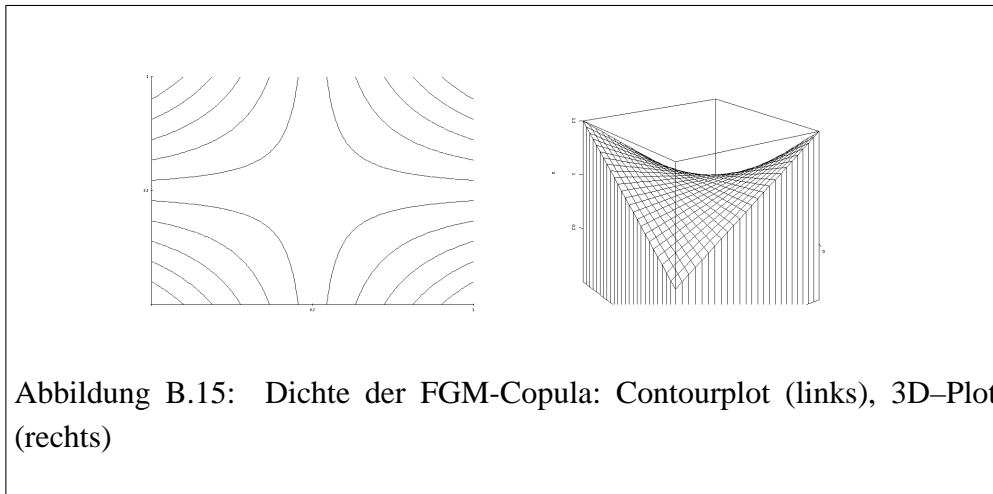
Definition 60 (Farlie-Gumbel-Morgenstern-Copula (FMG), Nelsen [65]).

$$C_{\delta}^{FGM}(u, v) = uv [1 + \alpha(1 - u)(1 - v)], -1 \leq \alpha \leq 1. \quad (\text{B.41})$$

Lemma 26 (Eigenschaften).

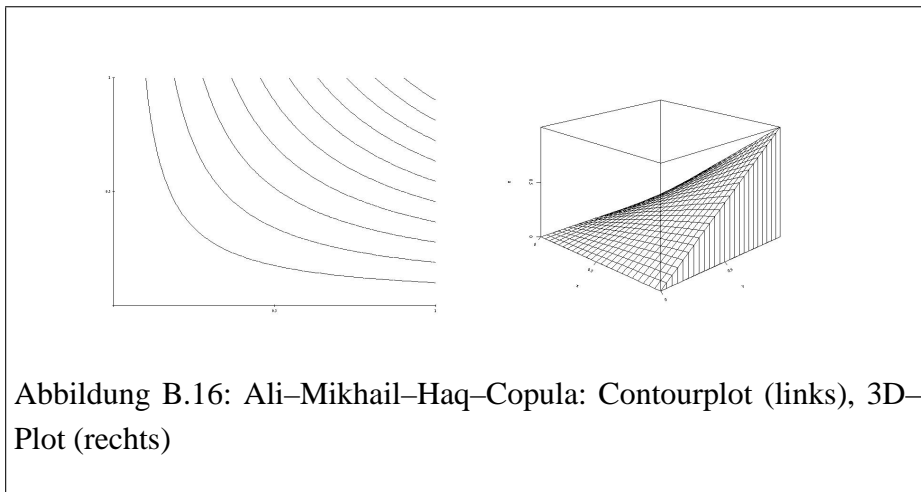
$$\delta = 3\rho_S, \quad (\text{B.42})$$

$$\delta = \frac{9}{2}\tau, \quad (\text{B.43})$$



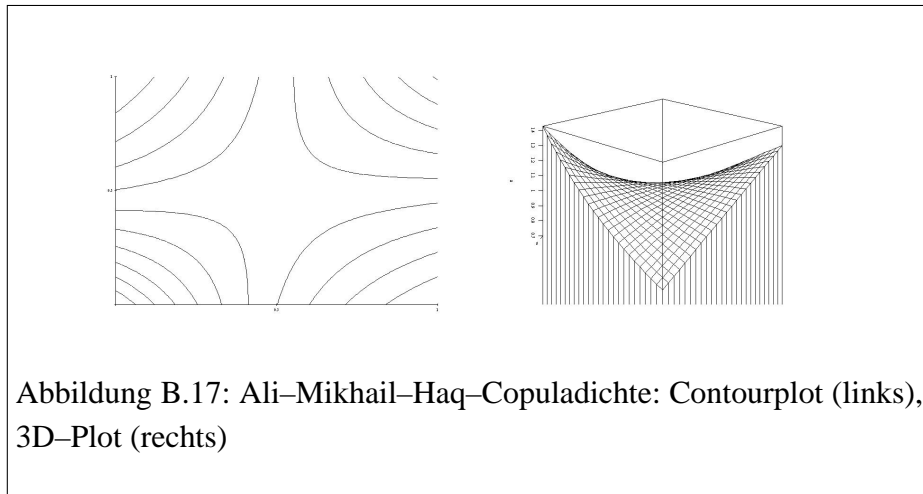
Lemma 27 (Dichte der Farlie–Gumbel–Morgenstern–Copula).

$$c_{\delta}^{FGM}(u, v) = 1 + \alpha(1 - u)(1 - v) - \alpha v(1 - u) - \alpha u(1 - v) + \alpha uv \quad (\text{B.44})$$



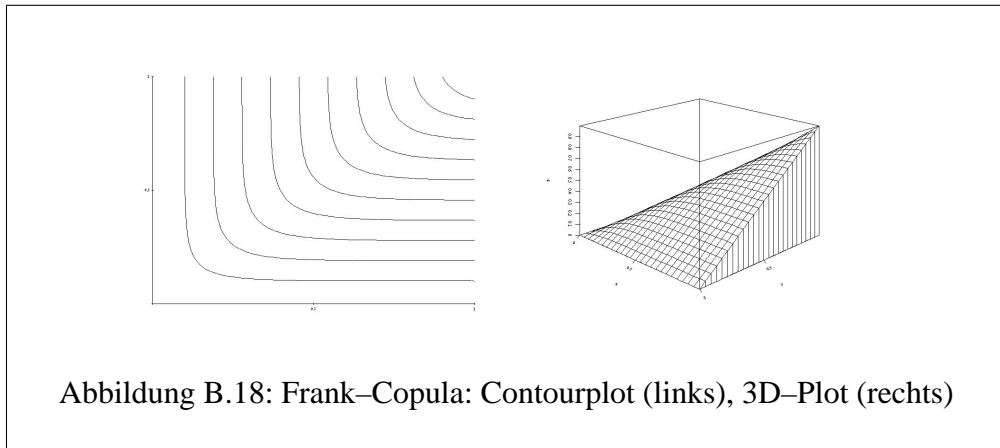
Definition 61 (Ali–Mikhail–Haq–Copula, Nelsen [65]). Für $\delta \in [-1, 1]$ ist die Ali–Mikhail–Haq–Copula definiert als

$$C_{\delta}^A(u, v) = \frac{uv}{1 - \delta(1 - u)(1 - v)}. \quad (\text{B.45})$$



Lemma 28 (Dichte der Ali-Mikhail-Haq-Copula).

$$c_{\delta}^A = -\frac{1 - 2\delta + \delta uv + \delta^2 uv + \delta v + \delta u + \delta^2 - \delta^2 v - \delta^2 u}{(-1 + \delta - \delta v - \delta u + \delta uv)^3}, \delta \in [-1, 1]. \quad (\text{B.46})$$



Definition 62 (Frank-Copula, Joe [48]). Für $0 \leq \vartheta \leq \infty$ ist

$$C^F(u, v; \vartheta) = -\vartheta^{-1} \log([\eta - (1 - \exp(-\vartheta u))(1 - \exp(-\vartheta v))]/\eta), \quad (\text{B.47})$$

mit

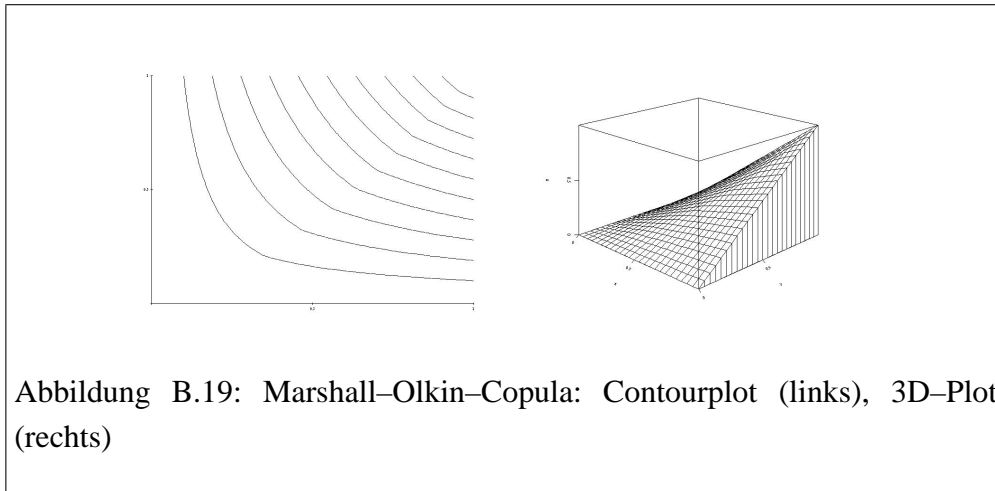
$$\eta = 1 - \exp(-\vartheta), \quad (\text{B.48})$$

die Frank Copula.

Lemma 29 (Dichte der Frank–Copula). Für $0 \leq \vartheta \leq \infty$ ist

$$c^F(u, v; \vartheta) = \vartheta \eta \exp(-\vartheta(u + v)) / [\eta - (1 - \exp(-\vartheta u))(1 - \exp(-\vartheta v))]^2, \quad (\text{B.49})$$

und η wie oben, die Dichte der Frank–Copula.

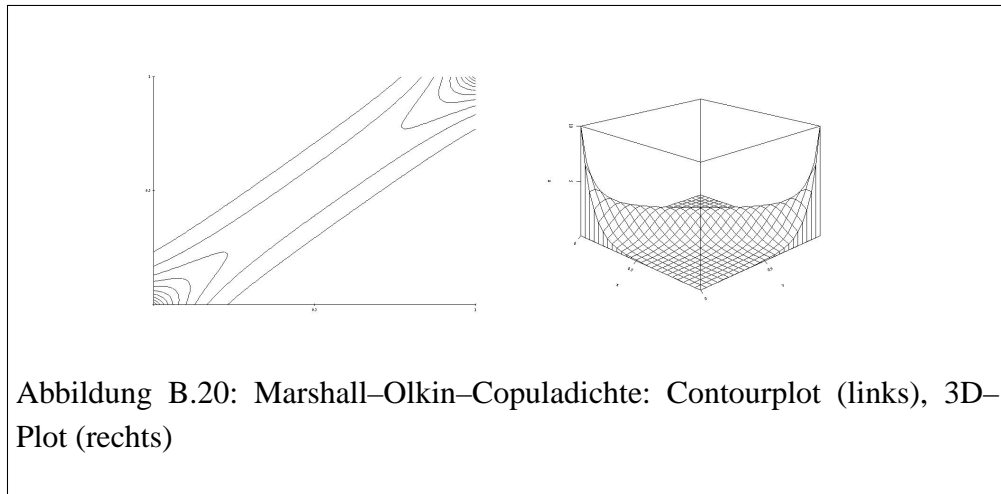


Definition 63 (Marshall–Olkin–Copula[65]). Für $0 \leq \alpha, \beta \leq 1$ ist die Marshall–Olkin–Copula definiert als

$$C_{\alpha, \beta}^{MO}(u, v) = \min(u^{1-\alpha}v, uv^{1-\beta}) = \begin{cases} u^{1-\alpha}v & u^\alpha \geq v^\beta \\ v^{1-\beta}u & u^\alpha \leq v^\beta. \end{cases} \quad (\text{B.50})$$

Lemma 30 (Dichte Marshall–Olkin–Copula, Nelsen [65]). Für $0 \leq \alpha, \beta \leq 1$ ist die Dichte der Marshall–Olkin–Copula definiert als

$$c_{\alpha, \beta}^{MO}(u, v) = \begin{cases} u^{-\alpha} & u^\alpha > v^\beta \\ v^{-\beta} & u^\alpha < v^\beta. \end{cases} \quad (\text{B.51})$$



Definition 64 (Gumbel–Exponential–Copula). Für $\vartheta \in [0, 1]$ ist die Gumbel–Exponential–Copula⁹ folgendermaßen definiert:

$$C_{\vartheta}^{GE}(u, v) = u + v - 1 + (1 - u)(1 - v) \exp(-\vartheta(\ln(1 - u))(\ln(1 - v))) \quad (\text{B.52})$$

Lemma 31 (Dichte der Gumbel–Exponential–Copula).

$$\begin{aligned} c_{\vartheta}^{GE}(u, v) &= (1 - u)^{-\vartheta \ln(1 - v)} - \vartheta \ln(1 - u)(1 - u)^{-\vartheta \ln(1 - v)} \\ &\quad - \vartheta \ln(1 - v)(1 - u)^{-\vartheta \ln(1 - v)} - \vartheta(1 - u)^{-\vartheta \ln(1 - v)} \\ &\quad + \vartheta^2 \ln(1 - v) \ln(1 - u)(1 - u)^{-\vartheta \ln(1 - v)}. \end{aligned} \quad (\text{B.53})$$

Definition 65 (Gumbel–Copula). Für $\delta \geq 1$ ist

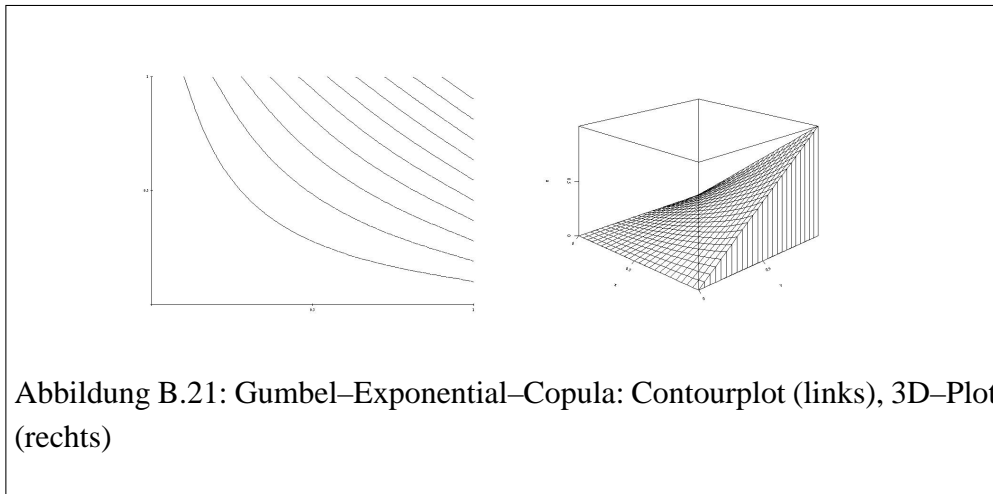
$$C_{\delta}^{Gu}(u, v) = \exp \left\{ -\left[(-\ln(u))^{\delta} + (-\ln(v))^{\delta} \right]^{1/\delta} \right\} \quad (\text{B.54})$$

die Gumbel Copula.

Lemma 32 (Dichte der Gumbel–Copula). Mit $a := (-\ln(u))^{\delta} + (-\ln(v))^{\delta}$ folgt

$$\begin{aligned} c_{\delta}^{Gu}(u, v) &= C_{\delta}^{Gu}(u, v; \delta)(uv)^{-1} \frac{(\ln(u) \ln(v))^{\delta-1}}{\left((-\ln(u))^{\delta} + (-\ln(v))^{\delta} \right)^{2-1/\delta}} \\ &\quad \left[\left((-\ln(u))^{-\delta} + (-\ln(v))^{-\delta} \right)^{1/\delta} + \delta - 1 \right] \end{aligned} \quad (\text{B.55})$$

⁹kurz: GE-Copula



Bemerkung 10 (Eigenschaften der Gumbel-Copula). Für die Gumbel-Copula und deren Parameter δ gilt:

- $\tau = 1 - \frac{1}{\delta}$;
- $\lambda_U = 2 - 2^{1/\delta}$, mit λ_U wird die obere Tail-Dependence bezeichnet;
- für die untere Tail-Dependence gilt $\lambda_L = 0$.

Definition 66 (Galambos-Copula). Für $0 \leq \delta < \infty$ ist

$$C_\delta^{Ga}(u, v) = uv \exp \{ [(-\log(u))^{-\delta} + (-\log(v))^{-\delta}]^{-1/\delta} \} \quad (\text{B.56})$$

die Galambos-Copula.

Lemma 33 (Dichte der Galambos-Copula). Mit $a := \frac{1}{(-1)^\delta \ln(u)^\delta} + \frac{1}{(-1)^\delta \ln(v)^\delta}$ folgt

$$\begin{aligned} c_\delta^{Ga}(u, v) = & [C_\delta^{Ga}(u, v; \delta)/(uv)] \\ & \cdot \left[1 - ((-\log(u))^{-\delta} + (-\log(v))^{-\delta})^{-1-1/\delta} \right. \\ & \cdot ((-\log(u))^{-\delta-1} + (-\log(v))^{-\delta-1}) \\ & + ((-\log(u))^{-\delta} + (-\log(v))^{-\delta})^{-2-1/\delta} \\ & \cdot ((-\log(u))(-\log(v)))^{-\delta-1} \\ & \left. \cdot \left\{ 1 + \delta + ((-\log(u))^{-\delta} + (-\log(v))^{-\delta})^{-1/\delta} \right\} \right] \end{aligned} \quad (\text{B.57})$$

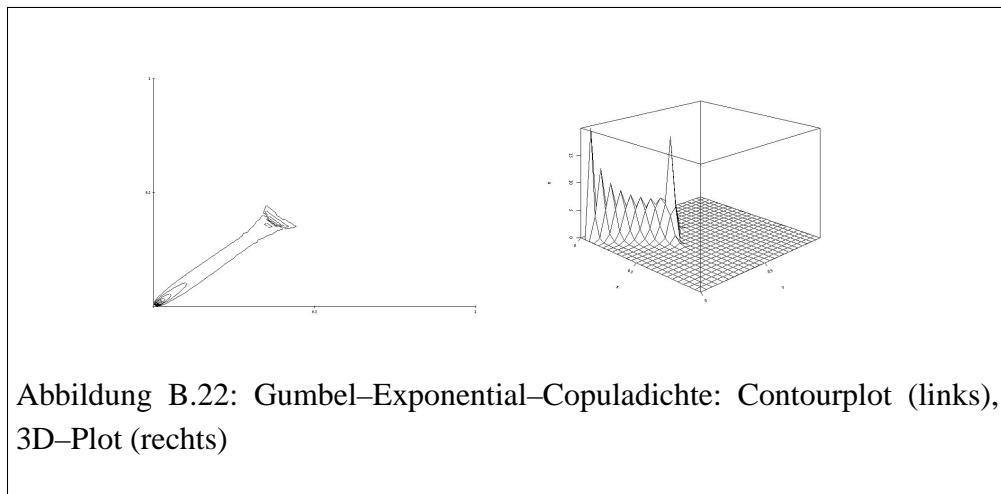


Abbildung B.22: Gumbel-Exponential-Copuladichte: Contourplot (links), 3D-Plot (rechts)

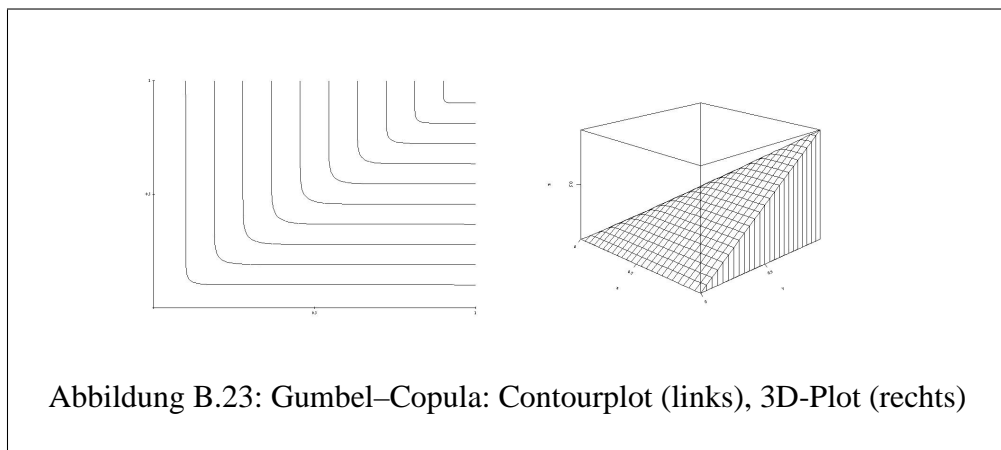
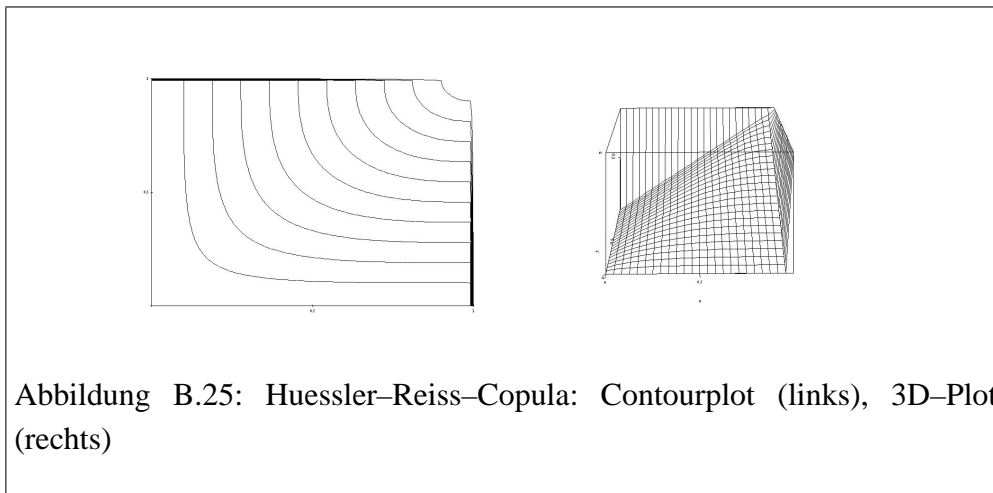
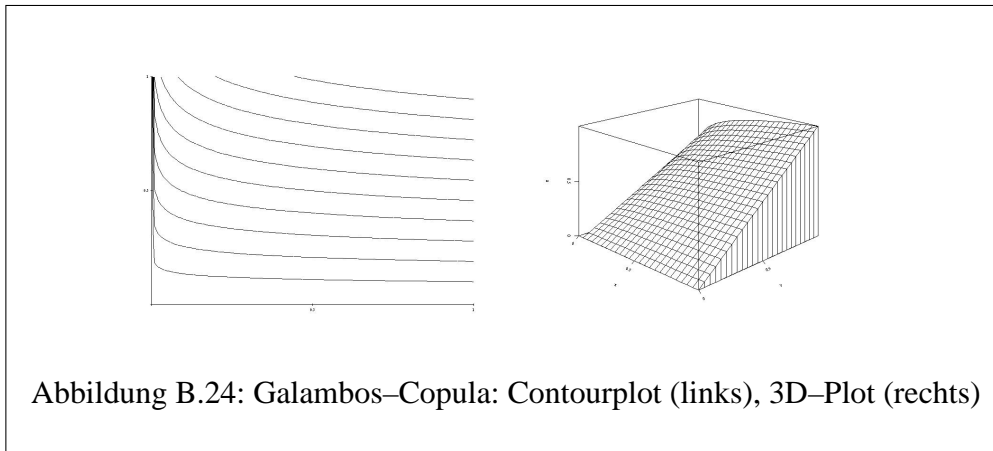


Abbildung B.23: Gumbel-Copula: Contourplot (links), 3D-Plot (rechts)

Definition 67 (Huessler-Reiss-Copula). Für $\delta \geq 0$ ist

$$\begin{aligned}
 C_{\delta}^{HR} = \exp \left\{ \log(u) \Phi \left[\frac{1}{\delta} + \frac{1}{2} \delta \log \left(\frac{-\log(u)}{-\log(v)} \right) \right] \right. \\
 \left. + \log(v) \Phi \left[\frac{1}{\delta} + \frac{1}{2} \delta \log \left(\frac{-\log(v)}{-\log(u)} \right) \right] \right\}
 \end{aligned}
 \tag{B.58}$$

die *Huessler-Reiss Copula*.

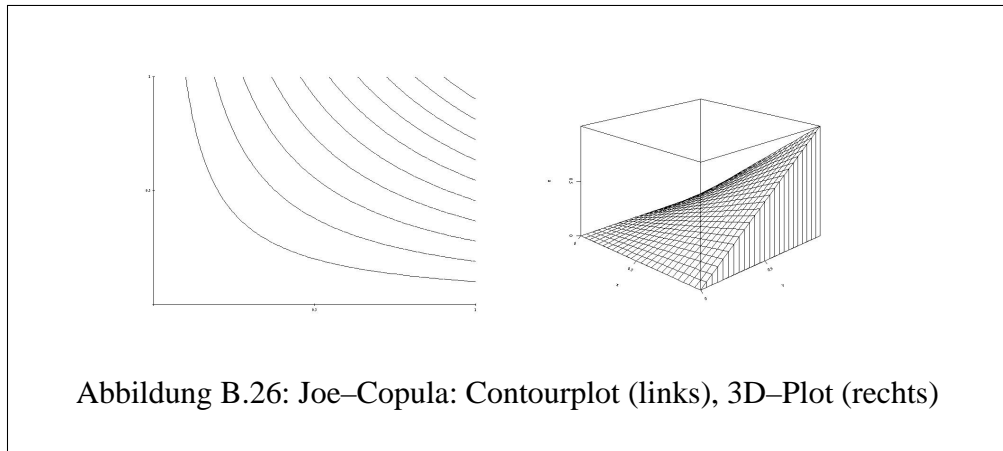


Definition 68 (Dichte der Huessler–Reiss–Copula). Für $\delta \geq 0$ und $z = \frac{-\log(u)}{-\log(v)}$, ist die Dichte gegeben durch

$$\begin{aligned}
 c^{HR}(u, v; \delta) &= (uv)^{-1} C^{HR}(u, v; z\delta) \left[\Phi \left(\delta^{-1} + \frac{1}{2} \delta \log(z^{-1}) \right) \Phi \left(\delta^{-1} + \frac{1}{2} \delta \log(z) \right) \right. \\
 &\quad \left. + \frac{1}{2} \delta (-\log(v))^{-1} \phi \left(\delta^{-1} + \frac{1}{2} \delta \log(z) \right) \right].
 \end{aligned}
 \tag{B.59}$$

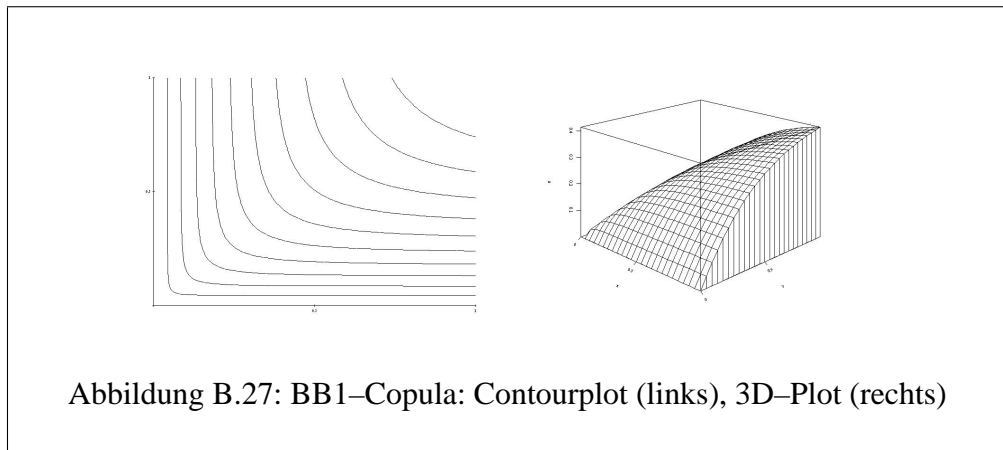
Definition 69 (Joe–Copula). Für $\delta \geq 0$ ist die Joe–Copula definiert durch

$$C_{\delta}^J(u, v) = 1 - \left((1-u)^{\delta} + (1-v)^{\delta} - (1-u)^{\delta}(1-v)^{\delta} \right)^{1/\delta}.
 \tag{B.60}$$



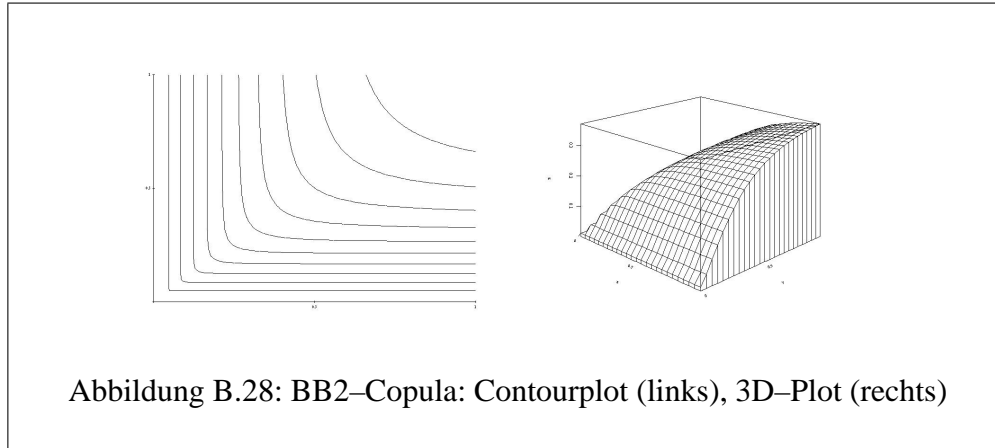
Lemma 34 (Dichte der Joe-Clayton-Copula). Mit $a = (1 - u)^\delta + (1 - v)^\delta - (1 - u)^\delta(1 - v)^\delta$ folgt

$$c_\delta^J(u, v) = \frac{((1 - u)^\delta + (1 - v)^\delta - (1 - u)^\delta(1 - v)^\delta)^{-2+1/\delta}}{(1 - u)^{\delta-1}(1 - v)^{\delta-1} [\delta - 1 + (1 - u)^\delta + (1 - v)^\delta - (1 - u)^\delta(1 - v)^\delta]} \quad (\text{B.61})$$



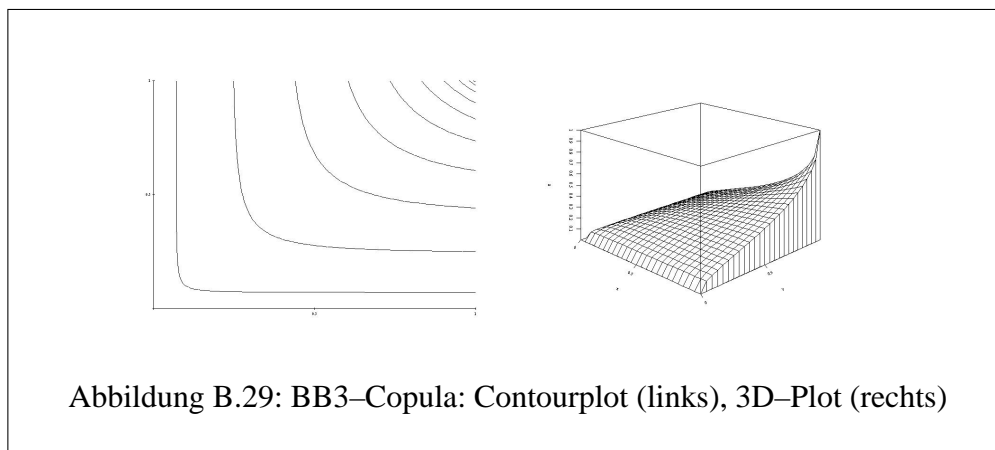
Definition 70 (BB1-Copula). Für $\vartheta > 0, \delta \geq 1$ ist die BB1-Copula definiert durch

$$C_{\vartheta, \delta}^{BB1}(u, v) = \left\{ 1 + [(u^{-\vartheta} - 1)^\delta + (v^{-\vartheta} - 1)^\delta]^{1/\delta} \right\}^{-1/\vartheta}. \quad (\text{B.62})$$



Definition 71 (BB2-Copula). Für $\vartheta > 0$, $\delta > 0$ ist die BB2-Copula definiert durch

$$C_{\vartheta, \delta}^{BB2}(u, v) = \left[1 + \frac{1}{\delta} \log (\exp(\delta(u^{-\vartheta}) + \exp(\delta(v^{-\vartheta}))) \right]^{1/\vartheta}. \quad (\text{B.63})$$



Definition 72 (BB3-Copula). Für $\vartheta \geq 1$, $\delta > 0$ ist die BB3-Copula definiert durch

$$C_{\vartheta, \delta}^{BB3}(u, v) = \exp \left\{ - \left[\frac{1}{\delta} \log (\exp(\delta(-\log(u))^{\vartheta} + \exp(\delta(-\log(v))^{\vartheta} - 1)) \right]^{1/\vartheta} \right\}.$$

(B.64)

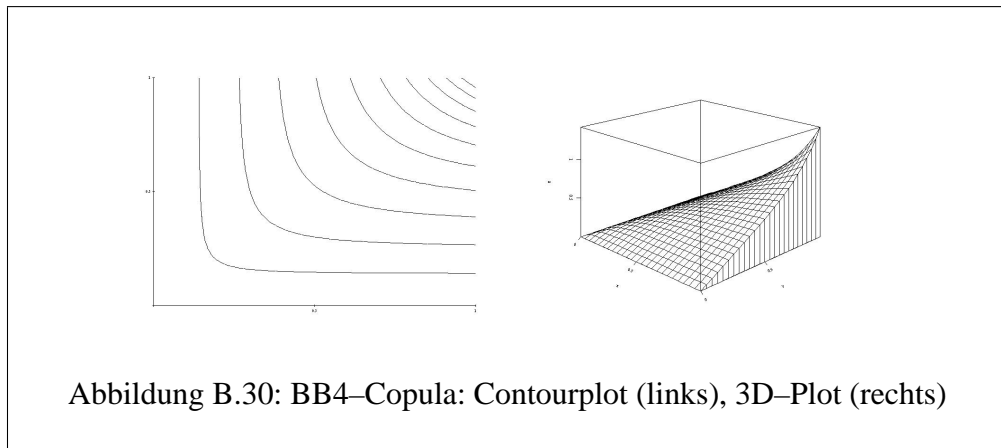


Abbildung B.30: BB4-Copula: Contourplot (links), 3D-Plot (rechts)

Definition 73 (BB4-Copula). Für $\vartheta \geq 1$, $\delta > 0$ ist die BB4-Copula definiert durch

$$C_{\vartheta, \delta}^{BB4}(u, v) = \left(u^{-\vartheta} + v^{-\vartheta} - 1 - \left[(u^{-\vartheta} - 1)^{-\delta} + (v^{-\vartheta} - 1)^{-\delta} \right]^{-1/\delta} \right)^{-1/\vartheta} \quad (\text{B.65})$$

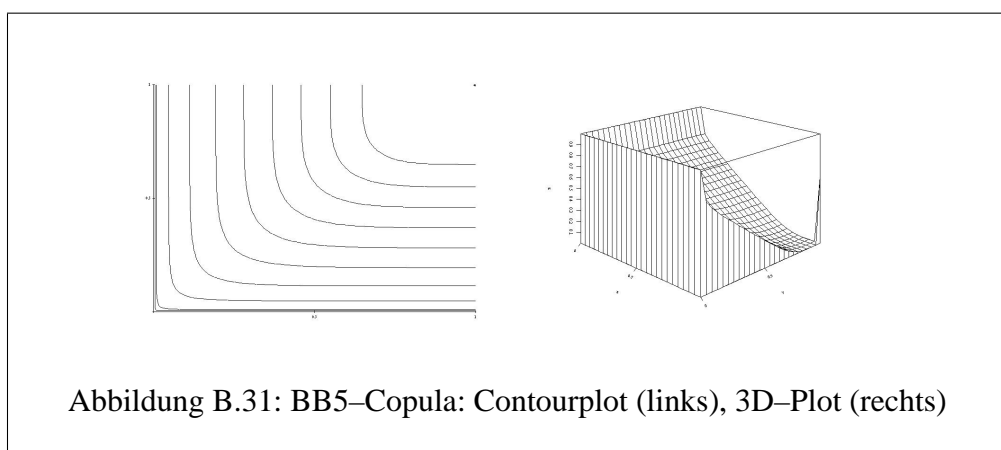
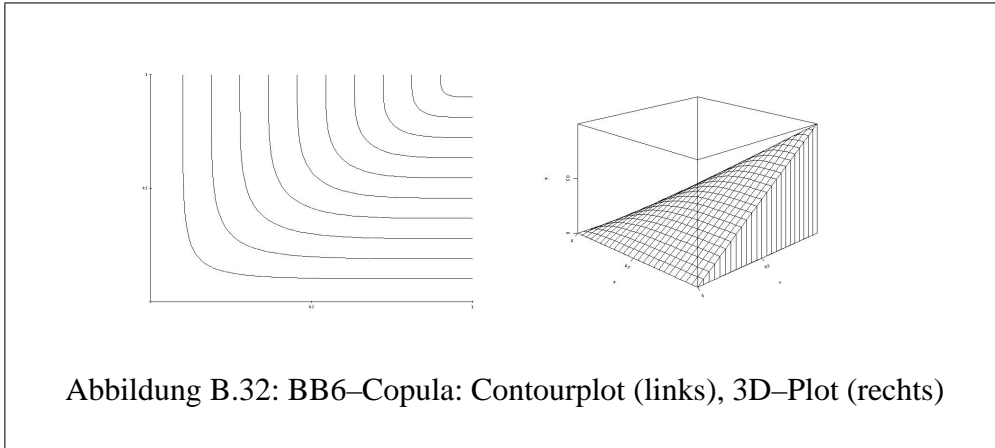


Abbildung B.31: BB5-Copula: Contourplot (links), 3D-Plot (rechts)

Definition 74 (BB5–Copula). Für $\delta > 0$, $\vartheta \geq 1$ ist die BB5–Copula definiert durch

$$C_{\delta, \vartheta}^{BB5}(u, v) = \exp \left\{ - \left[(-\log(u))^\theta + (-\log(v))^\theta - \left((-\log(u))^{-\vartheta\delta} + (-\log(v))^{-\vartheta\delta} \right)^{-1/\delta} \right]^{1/\vartheta} \right\} \quad (\text{B.66})$$

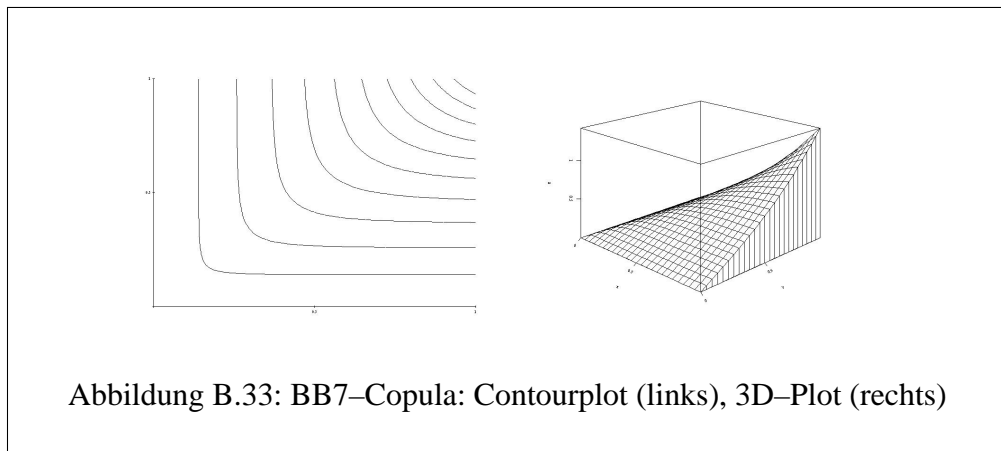


Definition 75 (BB6–Copula). Für $\vartheta \geq 1$, $\delta \geq 1$ ist die BB6–Copula definiert durch

$$C_{\vartheta, \delta}^{BB6}(u, v) = 1 - \left(1 - \exp \left\{ - \left[(-\log(1 - (1 - u)^\delta))^\delta + (-\log(1 - (1 - v)^\delta))^\delta \right]^{1/\delta} \right\} \right)^{1/\vartheta}. \quad (\text{B.67})$$

Definition 76 (BB7–Copula). Für $\vartheta \geq 1$, $\delta > 0$ ist die BB7–Copula definiert durch

$$C_{\vartheta, \delta}^{BB7}(u, v) = 1 - \left(1 - \left[(1 - (1 - u)^\vartheta)^{-\delta} + (1 - (1 - v)^\vartheta)^{-\delta} - 1 \right]^{-1/\delta} \right)^{1/\vartheta}. \quad (\text{B.68})$$



B.1.4 Simulationsalgorithmus der Studentschen Zeitreihe

Listing B.1: Algorithmus der Studentschen Zeitreihe

```

/* *****
Student timeseries
***** */

TTimeStudentTSample :: TTimeStudentTSample ()
{
    sigma=GlobalData . TSSigma ;
    alpha=GlobalData . TSAAlpha ;
    beta=GlobalData . TSBeta ;
    q=GlobalData . TSQ ;
    T=GlobalData . TSSize ;
    kern=0 ;
    filename="studtime . dat" ;
    gammadata=0 ;
    rmdata=0 ;
    smsum=0 ;
    gmu=GlobalData . TSMu ;
    gsigma=GlobalData . TSSigma2 ;
    count=2 ;
}
unsigned TTimeStudentTSample :: EditParameter (TWindow * parent)
{

```

```

TCalcDoubleField falpha (IDC_TimeStalpha , alpha ,0 , MYMAXDOUBLE) ,
    fsigma (IDC_TimeStsigma , sigma ,0 ,MYMAXDOUBLE) ,
    fbeta (IDC_TimeStbeta , beta , -MYMAXDOUBLE,MYMAXDOUBLE) ,
    fgmu (IDC_STTMU , gmu ,0 , MYMAXDOUBLE) ,
    fgsigma (IDC_STTSIGMA , gsigma ,0 ,MYMAXDOUBLE);

TIntField fsize (IDC_TimeStsize , T, 1 , MAXSAMPLESIZE ) ,
    fq ( IDC_TimeStq ,q,1 ,MAXSAMPLESIZE/2);

TAlterField fkern (kern ,4 ,IDC_RBETA ,IDC_RGAUSS ,IDC_RTEST ,IDC_IID );

TWriteFileField fname (IDC_NAME, filename );

TDialog d (IDD_TimeStudent);

d << falpha << fsize << fsigma << fbeta <<fq << fgmu << fgsigma
    << fkern << fname;

unsigned ch = d.Handle ( parent );
if (ch != IDCANCEL) {
    GlobalData.TSAlpha = alpha = falpha.GetValue ();
    GlobalData.TSSize = T = fsize.GetValue ();
    GlobalData.TSQ = q = fq.GetValue ();
    GlobalData.TSBeta = beta = fbeta.GetValue ();
    GlobalData.TSSigma = sigma = fsigma.GetValue ();
    GlobalData.TSMu = gmu = fgmu.GetValue ();
    GlobalData.TSSigma2 = gsigma = fgsigma.GetValue ();
    kern = (int)fkern.Selector ();
    DataSize = T;
}
calcm ();
return ch;
}

double TTimeStudentTSample::kdf(double x , int k)
{
    switch(k)
    {
        case 0: if ((x>-1) && (x<0))
            return exp(log(1+x)*(1+beta));
        else
            if (x>=0)
                return 1.0;
            else
                return 0.0;
    }
}

```

```

    case 1:
        if (x<=0)
            return XGaussianDF ((x-gmu)/ gsigma)
                /(XGaussianDF(-gmu/ gsigma));
        else
            return 1.0;
    case 2:
        if (x<=0){
            if (x>(-1)/q){
                return 1.0;
            }
            else {
                if(x>-1){
                    if ((x+1)*0.9*q/(-1+q)>0)
                        return (x+1)*0.9*q/(-1+q);
                    else
                        return 0.0;
                }
                else
                    return 0.0;
            }
        }
    }

    default: return 0.0;
}

}

double TTimeStudentTSample::mymin(double * data ,int s){
    double min=data [0];
    for (int i=1;i<s;i++){
        if ((data [ i]<min)){
            min=data [ i];
        }
    }
    return min;
}

int TTimeStudentTSample::mymax(int * data , int s){
    int max=data [0];
    for(int i=1;i<q;i++)
        if (data [i]>max)

```

```

        max=data [ i ];
    return max;
}

void TTimeStudentTSample :: calcsm () {
    double r=alpha / 2;
    double * sm = new double [ q ];
    for ( int i=1; i<q+1; i++) {
        double a1 = kdf ( -(i - 1) / q, kern ),
               a2 = 2 * kdf ( -i / q, kern ),
               a3 = kdf ( -(i - 1) / q, kern );
        sm [ i - 1 ] = a1 - a2 + a3;
    }
    rmdata = new double [ q ];
    for ( i=0; i<q; i++)
    {
        if ( sm [ i ] >= 0.0 )
            rmdata [ i ] = sm [ i ] * r;
        else
            rmdata [ i ] = 0.0;
    }
    double test=0.0;
    for ( i=0; i<q; i++)
        test=test+rmdata [ i ];
    int de, si;
    String a = fcvt ( test, 6, &de, &si );
    int laenge=(int)(q*q);
    gammadata = new double [ laenge ];
    int k=0;
    for ( i=0; i<q; i++)
        for ( int j=0; j<q; j++) {
            gammadata [ k ] = XGammaData ( rmdata [ j ] );
            k++;
        }
    delete [] sm;
    count=0;
}

double TTimeStudentTSample :: calcYt ()
{
    double sum=0.0;
    for ( int i=0; i<q; i++)
        for ( int j=i; j<q; j++)

```

```

        sum=sum+gamdata[i*((int)q)+j];
return sum;
}

void TTimeStudentTSample::shift(){
    for (int i=q-1;i>0;i--)
        for (int j=0;j<q;j++)
            gamdata[i*((int)q)+j]=gamdata[i*((int)q-1)+j];

    for (i=0;i<q;i++)
        gamdata[i]=XGammaData(rmdata[i]);
}

BOOL TTimeStudentTSample::CreateData (TWindow * parent)
{
    for (int i=0;i<T;i++)
    {
        DataPtr[i].x=i+1;
        DataPtr[i].y=XGaussianData()*sigma/sqrt(calcYt()/alpha/2);
        shift();
    }
    return TRUE;
}

double TTimeStudentTSample::acf1(){
    double r=11.268/2;
    double acf=0.0;
    for (int i=1;i<q;i++)
        acf=acf+i*sqr(kdf(-i/q,kern));
    double h1=xgamma(r-0.5),
        h2=xgamma(r);
    return 1/(sqrt(sqr(r)+sqr(r)*acf)); // -sqr(h1/h2);
}

TTimeStudentTSample::~TTimeStudentTSample(){
    if (gamdata!=0)
        delete [] gamdata;
    if (rmdata!=0)
        delete [] rmdata;
}

```


B.1.5 Statistische Plots: Parameter der Copulas

Copula	Parameter
Gauss	$\rho = 0.0$
Student	$\alpha = 5.0, \rho = 0.0$
Logistic	$\alpha = 0.5$
Plackett	$\psi = 5.0$
Clayton	$\delta = 3.0$
Marshall–Olkin	$\alpha = 0.3, \beta = 0.2$
Frank	$\delta = 10.0$
Frechet	$\alpha = 0.3, \beta = 0.2$
Mardia	$\delta = 0.5$
Cuadras–Auge	$\delta = 0.2$
Ali–Mikhail–Haq	$\delta = 0.3$
Farlie–Gumbel–Morgenstern	$\delta = 0.5$
Gumbel–Exponential	$\theta = 0.5$
Gumbel	$\delta = 10.0$
Galambos	$\delta = 5.0$
Huessler–Reiss	$\delta = 1.0$
Joe	$\delta = 1.0$
BB1	$\theta = 1.0, \delta = 2.0$
BB2	$\theta = 1.0, \delta = 1.0$
BB3	$\theta = 2.0, \delta = 1.0$
BB4	$\theta = 2.0, \delta = 1.0$
BB5	$\theta = 2.0, \delta = 1.0$
BB6	$\theta = 2.0, \delta = 2.0$
BB7	$\theta = 2.0, \delta = 1.0$

Tabelle B.1: Parameter der Copulas für die statistischen Plots

B.1.6 Autokorrelationsfunktion der absoluten Renditen



Tabelle B.2: Autokorrelationsfunktionen der absoluten Renditen I

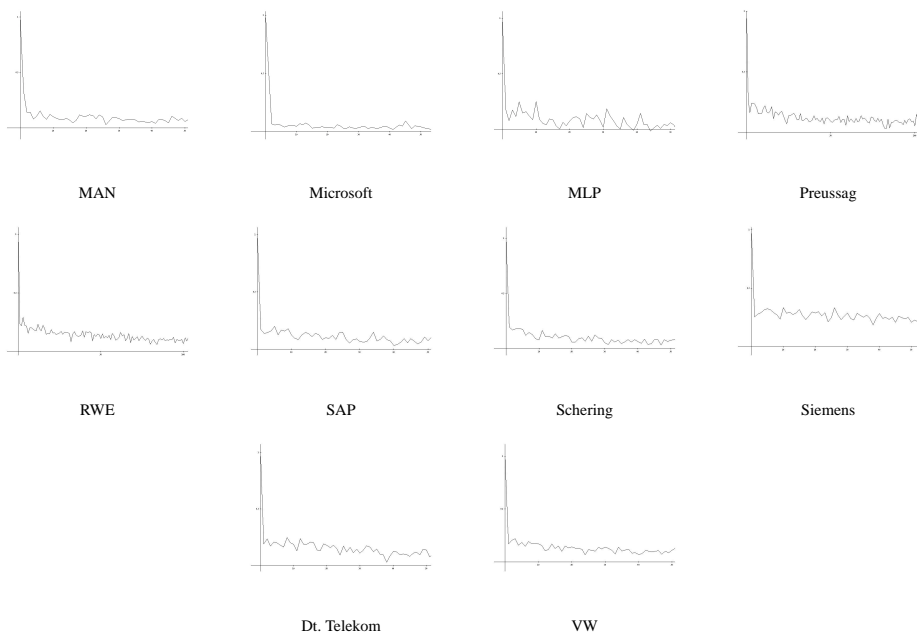


Tabelle B.3: Autokorrelationsfunktionen der absoluten Renditen II

Dank

Dieser Arbeit entstand in enger Zusammenarbeit mit den Mitgliedern der Fachgruppe von Prof. Dr. R.-D. Reiss. Vor allem danke ich Herrn Prof. Dr. R.-D. Reiss für die Unterstützung und die kritische Würdigung meiner Dissertation. Durch ihn hatte ich die Möglichkeit diese Arbeit zu erstellen. Außerdem danke ich den anderen Mitgliedern dieser Fachgruppe, Dr. M. Thomas und PD. Dr. E. Kaufmann für ihre Diskussionsbereitschaft und Hilfestellungen an kritischen Stellen meiner Arbeit. Ich danke unserer Sekretärin Frau Lange für ihre Unterstützung, mit der sie mir immer den Rücken frei hielt.

Ich danke allen, die an der Korrektur mitgearbeitet und mir Mut und Unterstützung entgegen gebracht haben.

Vor allem danke ich meinen Eltern, die mir das Studium ermöglicht haben und ohne deren Unterstützung ich die Dissertation nie hätte vollenden können.

„Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer, nicht angegebener Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.“

Siegen, den 12. Juli 2004

Literaturverzeichnis

- [1] Acerbi, C., Nordio, C., Sirtori, C., *Expected shortfall as a tool for finance risk management*. Working paper, 2001, www.gloriamundi.org/var/wps.html.
- [2] Acerbi, C., Tasche, D., *Expected Shortfall: A natural coherent alternative to Value at Risk*, 2001, www-m4.mathematik.tu-muenchen.de/m4/pers/tasche/wilmott.pdf
- [3] Acerbi, C., Tasche, D., *On the coherence of Expected Shortfall*, 2002, www-m4.mathematik.tu-muenchen.de/m4/pers/tasche/shortfall.pdf.
- [4] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., *Coherent measures of risk*, Math. Fin. Band 9 (3), S. 203-228, 1999.
- [5] Bams, D., Wielhouwer, J. , *Empirical Issues in Value-at-Risk*, Dezember, 2000.
- [6] Barbe, P., Genest, P., Ghoudi, K., Remillard, B. , *On Kendall's process*, Journal of Multivariate Analysis, Band 58, S. 197-229, 1996.
- [7] Blattberg, R. C., Gonedes, N. J. , *A comparison of the stable and Student distributions as statistical models for stock prices*, Journal of Business, Band 47, S. 224-280, 1974.
- [8] Bollerslev, T. ,*Generalized autoregressive conditional heteroskedasticity*, J. Econometrics 31, S. 307-327, 1986.
- [9] Bornhüter, P. , *Diplomarbeit: Statistische Modellierung von Finanzportefeuilles*, Siegen, 2002 (nicht veröffentlicht).

- [10] Breymann, W., Dias, A., Embrechts, P., *Dependence Structures for Multivariate High-Frequency Data in Finance*, Department of Mathematics, ETHZ, 2003.
- [11] Bouye, E., Durrleman V., Nikeghbali A., Riboulet G., Roncalli T., *Copulas for finance; A Reading Guide and Some Applications*. Financial Econometrics Research Centre, City University Business School London, Groupe de Recherche Operationnelle, Credit Lyonnais, Paris, 2000.
- [12] Bowman, K. O., Shenton, L. R., *Properties of estimators for the gamma distribution*, Statistics: Textbooks and Monographs, Marcel Dekker, New York / Basel, 1988.
- [13] Breidt, F. J., Davis, R. A., *Extremes of stochastic volatility models*, Ann. Appl. Probab. Band 8, S. 664–675, 1998.
- [14] Brockwell, P. J., Davis, R. A., *Time Series: Theory and Methods*, Springer Series in Statistics, 1987.
- [15] Brose, G., Vogel, A., Duddy, K., *JAVA Programming with CORBA*, OMG Press, 2001.
- [16] Cizek, P., Jeske, R., Klinke, S., Symanzik, J., *XploRe Beginners Guide*, www.i-xplore.de.
- [17] Clemente, A. D., Ramono, C., *Working Paper: Beyond Markowitz: Building the optimal portfolio using non-elliptical asset return distributions*, Centro Interdipartimentale sul Diritto e l’Economia dei Mercati, 2003.
- [18] Cleveland, W. S., *Visualizing Data*, HOBART Press, 1993.
- [19] Coles, St., Heffernan, J., Tawn, J., *Dependence Measures for Extreme Value Analyses*, Department of Mathematics and Statistics, Lancaster University, 1999.
- [20] *Dokumentation Java+RMI*, java.sun.com/1.4.2/docs/guide/rmi/.
- [21] Durrleman, V., Nikeghbali, A., Roncalli, T., *Which copula is the right one?*, Groupe de Recherche Operationnelle, Credit Lyonnais, 2000.

- [22] Embrechts, P., Lindskog F., McNeil A., *Modelling Dependence with Copulas and Applications to Risk Management*, Department of Mathematics ETHZ, 2001, www.math.ethz.ch/finance.
- [23] Embrechts, P., McNeil, A. J., Strautmann, D. , *Modelling Extremal Events*, Heidelberg, 1997.
- [24] Embrechts, P., McNeil, A., Strautmann, D., *Correlation and dependency in risk management: properties und pitfalls*, Paper, 1998, www.math.ethz.ch/finance.
- [25] Engle, R. F., *Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation*, *Econometrica* Band 50, S. 987-1007, 1982.
- [26] Fahrmeir, L., Künstler, R., Pigeot, I., Tutz, G., *Statistik: Der Weg zur Datenanalyse*, Berlin-Heidelberg-New York (4. Auflage), 2003.
- [27] Falk, M., Becker, R., Marohn F., *Angewandte Statistik mit SAS*, Springer Lehrbuch, 1995.
- [28] Fama, E. F., *The behavior of stock market prices*, *Journal of Business*, Band 39, 34-105, 1965.
- [29] Freisleben, B., Baumgart, J., *Vorlesung: Grafischen Datenverarbeitung*, Universität Siegen, SS 1999 .
- [30] *Gabler Wirtschafts Lexikon*, 13. vollständig überarbeitete Auflage, 1994.
- [31] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns*, Addison-Wesley, 2000.
- [32] Genest, C., Mackay, J., *The joy of copulas: Bivariate distributions with uniform marginals*, *American Statistician*, Band 40, S. 280-283, 1986.
- [33] Genest, C., Rivest, L., *Statistical inference procedures for bivariate Archimedean copulas*. *Journal of the American Statistical Association*, Band 88, S. 1034-1043, 1993.
- [34] Geyer, A. L. J., *Berechnung und Eigenschaften von mehrperiodigen Renditen*, Institut für Operations Research, Wirtschaftsuniversität Wien.

- [35] Glasserman, P., Heidelberg, P., Shahabuddin, P., *Portfolio value-at-risk with heavy tailed risk factors*, Working paper, Columbia University New York, 2000.
- [36] Gosling, J., Joy, B., Steele, G., Brache, G., *The Java Language Specification Second Edition*, Addison-Wesley, 2000.
- [37] Graf, F., *Vorlesung: Wahrscheinlichkeitsrechnung I*, Universität Erlangen-Nürnberg, 2003.
- [38] Groth, A., *Estimation of periodicity in time series by ordinal analysis with an application to speech*, Department of Mathematics and Informatics, Ernst Moritz Arndt University of Greifswald.
- [39] Hamilton, J. D., *Time Series Analysis*, Princeton University Press, 1994.
- [40] Härdle, W., Hlavka, Z., Klinke, S., *XploRe Application Guide*, www.i-xplore.de .
- [41] Härdle, W., Klinke, S., Müller, M., *XploRe Learning Guide*. www.i-xplore.de .
- [42] Härter, E., *Wahrscheinlichkeitsrechnung, Statistik und mathematische Grundlagen*, Vandenhoeck & Ruprecht, 1987.
- [43] Henning, M., Vinoski, St., *Advanced CORBA Programming with C++*, Addison-Wesley, 2001.
- [44] Hofmann, J., Jobst, F., Schabenberger, R., *Programmieren mit COM und CORBA*, Hanser, 2001.
- [45] Huisman, R., Koedijk, K., Kool, C., Palm, F., *Fat tails in small samples*, Working paper, Limburg Institute of Financial Economics, Universität Maastricht, 1997.
- [46] Huisman, R., Koedijk, K., Pownall, R., *VaR-x: Fat tails in financial risk management*, Journal of Risk, Band 1 (1), S. 46-61, 1998.
- [47] Irmischer, K., *Vortrag RMI*, www.informatik.uni-leipzig.de/rnvs/lehre/vs/ .

- [48] Joe, H., *Multivariate Models and Dependence Concepts*, Chapman & Hall, London 1997.
- [49] Johnson, N. L., Kotz, S., *Distributions in Statistics: Continuous Univariate Distributions–1, –2*, Boston, 1970.
- [50] Johnson, N. L., Kotz, S. , *Distributions in Statistics: Continuous Multivariate Distributions*, Wiley, New York, 1972.
- [51] Jorion, P., *Value at risk: the new benchmark for managing financial risk*, McGraw-Hill, Second Edition 2000.
- [52] Koschel, A., Hoffmann, D., Wipf, M., *Kombination von CORBA und Java für verteilte Client/Server Anwendungen im Intranet/Internet*, FZI Forschungszentrum Informatik Karlsruhe.
- [53] Katritzke, F., Merzenich, W., Reiss, R.-D., Thomas, M., *A Pascal-based Approach towards Statistical Computing*, Universität Siegen, 2002.
- [54] Lindskog, F., *Modelling dependence with copulas and applications to risk management*, Master thesis, ETH Zürich und KTH Stockholm, 2000, <http://www.risklap.ch/Papers.html#MTLindskog> .
- [55] Lindskog, F., McNeil, A., Schmock, U., *Kendall's tau for elliptical distributions*, Research report, RiskLab Switzerland, 2001, www.risklab.ch/Papers.html#KendallsTau .
- [56] Magnus, J. R., Neudecker, H., *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York, 1988.
- [57] Makevergene, Y., Sornett, D. , *Testing the Gaussian copula hypothesis for financial assets dependences*, Research Paper, Quantitative Finance, Band 3, S. 231-250, 2003.
- [58] Mandelbrot, B., *The variation of certain speculative prices*, Journal of Business, Band 36, S. 394-419, 1963.
- [59] Manzotti, A., Peres, F.J., Quiroz, A. J. , *A statistic for testing the null hypothesis of elliptical symmetry*, Multivariate Analysis, Band 81, S. 274-285, 2002.

- [60] Markowitz, H. , *Portfolio Selection*, The Journal of Finance, Vol. VII, No. 1, S. 77-91, 1952.
- [61] Mashal, R. , Zeevi, A., *Beyond Correlation: Extreme Comovements Between Financial Assets*, Columbia University, Working Paper, 2002.
- [62] McLachlan, G. J., Kristnan, T., *The EM Algorithm and Extensions*, Wiley, New York, 1997.
- [63] Melchiori, R. M. , *Which Archimedean Copula is the right one?*, Universidad Nacional del Litoral, 2003.
- [64] Nagel, M., Benner, A., Ostermann, R., Henschke, K., *Grafische Datenanalyse*, Gustav Fischer Verlag, 1996.
- [65] Nelsen, R. B., *An introduction to copulas*, Springer, 1999.
- [66] Pant, V. , Chang, W., *An empirical comparison of methods for incorporating fat tails into value-at-risk models*, Journal of Risk, Band 3(3), S. 99-119, 2001.
- [67] Patton, A. J., *Estimation of copula models for time series of possibly different lengths*, University of California, San Diego, Department of Economics, 2001.
- [68] Poon, S.-H., Rockinger, M., Tawn, J., *Extreme-Value Dependence in Financial Markets: Diagnostics, Models and Financial Implications*, 2003.
- [69] Pöpsel, J., Claussen, U., Klein, R.-D., Plate, J., *Computergrafik: Algorithmen und Implementierung*, Springer-Verlag 1994.
- [70] Rankin, J. R., *Computer Graphics Software Construction*, Prentice Hall, 1989.
- [71] Reiss, R.-D., *Approximate Distributions of Order Statistics*, Springer-Verlag, 1989.
- [72] Reiss, R.-D., Spillmann, T., *A Class of Stochastic Volatility Series with Stationary Student Distributions*. Working Paper, Uni-Siegen (noch nicht veröffentlicht), 2003-2004.

- [73] Reiss, R.-D., Thomas, M., *Statistical Analysis of Extreme Values (Second Edition)*, Birkhäuser Verlag, 2000.
- [74] Rockafellar, R. T., Urgasev, S., *Conditional Value-at-Risk for General Loss Distributions*, Research Report 2001-5, ISE Dept., University of Florida, 2001, www.ise.ufl.edu/urgasev/cvar2.pdf.
- [75] Rockafellar, R. T., Urgasev, S., *Optimization of Conditional Value-at-Risk*, The Journal of Risk. Vol.2, No. 3, S. 21-41, 2000, www.ise.flu.edu/urgasev/cvar.pdf.
- [76] Rohatgi, V. K., *An Introduction to Probability Theory and Mathematical Statistics*, Wiley-interscience publication, John Wiley & Sons, 1976.
- [77] Romano, C., *Calibrating and simulating copula functions: an application to the italian stock market*, Working Paper, Centro Interdipartimentale sul Diritto e l'Economia dei Mercati, 2002.
- [78] Schäfer, W., *Maximum-Likelihood Schätzungen der Funktionsparameter der Gammaverteilung bei vollständigen und Typ II zensierten Stichproben*, Verlag der Ferber'schen Universitätsbuchhandlung, 1984.
- [79] Schmidt, R., *Dependencies of Extreme Events in Finance*, Dissertation, Universität Ulm, 2003.
- [80] Segal, M., Akeley, K., *The OpenGL Graphics System: A Specification (Version 1.5)*, www.opengl.org/documentation/specs/version1.5/glspec15.pdf.
- [81] Segal, M., Akeley, K., *The Design of the OpenGL Graphic Interface*, Silicon Graphics Computer Systems.
- [82] Slusallek, P., *Computergraphik -OpenGL-*, graphics.cs.uni-sb.de/Courses/ws0001/cg/folien/Opengl.pdf.
- [83] Spillmann, T., *Eine CORBA-basierte Client/Server-Architektur für statistische Visualisierungen*, Uni-Siegen, 2001 (nicht veröffentlicht).
- [84] Stroustrup, B., *The C++ Programming Language (Third Edition)*, Addison-Wesley, 2002.
- [85] Taylor, S. , *Modelling Financial Time Series*, Wiley, Chichester, 1986.

- [86] Schierenbeck, H., Wiedemann, A. , *Marktwertrechnungen im Finanzcontrolling*, Schäffer-Poeschel Verlag, 1996.
- [87] Yamia, Y., Yoshiba, T., *Comparative analysis of expected shortfall and value-at-risk under market stress*, Bank of Japan, 2002.

Index

- 1-q-Quantil, 59
- 2D-Bildraum, 11, 13
- 2D-Grafik, 27
- 2D-Koordinaten, 157
- 2D-Plot, 37
- 2D-Übergangsraum, 13, 157
- 2D-Weltraum, 13
- 2D-Window, 13
- 3D-Grafik, 27
- 3D-Grafiken, 27
- 3D-Grafikobjekte, 18, 35, 36, 160
- 3D-Objekte, 13
- 3D-Plot, 159
- 3D-Welten, 31, 32
- 3D-Weltraum, 13
- 3D-Window, 13

- Abhängigkeit, 98, 99, 119
- Abhängigkeitsmaß, 98
- absolute Stetigkeit der Copula, 124
- absoluten Renditen, 81, 94
- ACF, 81, 183
- ActiveX, 25, 27
- Aktien, 45
- Aktienkurs, 64
- Algorithmus, 5
- Ali-Mikhail-Haq-Copuladichte, 200
- Ali-Mikhail-Haq-Copula, 199

- allgemeiner Simulationsalgorithmus für
eine Copula, 133
- Anderson-Darling-Abstand, 141
- Anforderungen, 20
- Anlage, 45
- Applet, 33
- Arbeitsgemeinschaft, 21
- Archimedische Copula, 106, 142
- Architektur, 3, 15, 16, 20, 36
- arithmetische Rendite, 49
- Attribute, 40
- Audiodaten, 27
- Aufgabenbereich, 15
- Ausgabe, 16, 34
- Ausgabefenster, 17
- Ausschnitt, 11
- Auswahl der richtigen Copula, 138
- Auswahl einer Copula, 137
- Autokorrelation, 49
- Autokorrelationsfunktion, 48, 81, 93,
183
- Autokovarianzfunktion, 78, 183

- Balkendiagramm, 5, 6
- Bandbreite, 7
- BB1-Copula, 206
- BB2-Copula, 207
- BB3-Copula, 207
- BB4-Copula, 208

BB5–Copula, 209
 BB6–Copula, 209
 BB7–Copula, 209
 bedingte Dichte, 110
 bedingte Erwartung, 111
 bedingte Normalverteilung, 65
 bedingte Verteilung, 110, 111
 bedingter Erwartungswert, 110, 184
 bekanntes α , 109, 112
 Benutzer, 43
 Benutzer–Interface, 31
 Berechnungen, 17
 Beta–Kern, 92
 Betrachter, 1, 5, 16, 17, 31
 Betrachtungsintervall, 47
 Betriebssystem, 2, 23, 26, 31
 Bildkoordinatensystem, 11
 Bildraum, 11, 17, 35, 157
 Bildschirm, 36
 Bivariate Copula, 144
 Bivariate Copulas, 185
 Bivariate Gumbel–Logistic–Copula, 197
 Bivariate Kerndichte, 8
 Bivariate Studentsche Verteilung, 97
 BoxPlot, 6
 Boxplot, 5
 Browser, 3, 33
 Bytecode, 25

 C++, 21, 23, 26, 34
 Callbacks, 18, 19, 31, 41–43
 Canonical Maximum Likelihood Methode, 132
 Cap, 45
 CDR, 22
 Cholesky–Zerlegung, 134, 135

 Clayton Copuladichte, 206
 Clayton–Copula, 147, 194
 Clayton–Copuladichte, 195
 Client, 3, 16, 18, 19, 25, 26, 33–36, 38, 42
 Client/Server–Architektur, 2, 3, 15, 16, 19, 30–34, 36, 40, 43, 163
 Clipping, 12–14, 35, 157
 clone, 164
 cloneObject, 34
 CML–Methode, 132
 COBOL, 23
 coherentes Risikomas, 58
 Collar, 45
 Collection Service, 24
 COM, 25, 26, 31
 COM+, 27
 COM–Standard, 26, 27
 Common Data Representation, 22
 Common Objekt Request Broker Architecture, 21
 Component Object Model, 25
 Concurrency Control Service, 24
 Cook–Johnson–Copula, Simulation, 135
 Copula, 98, 99, 101, 104, 118–120, 121, 131
 Copula Parameter, 215
 Copula, empirisch, 125
 Copula, Farlie–Gumbel–Morgenstern, 136
 Copula, Frank, 134
 Copuladichte, 124, 126, 127
 Copuladichte, Ali–Mikhail–Haq, 200
 Copuladichte, Clayton, 195
 Copuladichte, Eigenschaften, 127

Copuladichte, empirisch, 128
 Copuladichte, Farlie–Gumbel–Morgenstern, 199
 Copuladichte, Frank, 201
 Copuladichte, Galambos, 203
 Copuladichte, Gauss, 187
 Copuladichte, Gumbel, 202
 Copuladichte, Gumbel–Exponential, 202
 Copuladichte, Joe–Clayton, 206
 Copuladichte, Logistic Extreme, 191
 Copuladichte, Marshall–Olkin–Copula, 201
 Copuladichte, Plackett, 193
 Copuladichte, Student, 188
 CORBA, 21, 23, 26, 31, 33
 CORBA–Implementierung, 33
 CORBA–Objekt, 21, 22, 38
 CORBA–Schnittstelle, 34, 37, 166
 CORBA–Services, 23
 CORBA–Services Schnittstellen, 23
 CORBA–Standard, 21
 Cuadras–Auge–Copula, 197

 Datenanalyse, 5
 Datenbank, 18
 Datenmengen, 2, 3
 Datensatz, 5, 6, 19, 34
 Datensicherheit, 3, 19
 Dax, 107
 DCOM, 25, 31
 DCOM Systemarchitektur, 27
 Delegation, 165
 Design–Pattern, 39, 163
 destroy, 34
 Dichte, 7, 110
 Dienste, 26
 Dienstspezifikationen, 23
 Digamma–Funktion, 112
 DII, 22
 DirectX, 27, 32
 DirectX–Bibliotheken, 31
 Display–Server, 3, 14, 16–19, 33, 35, 39, 40, 42
 Distributed–COM, 26
 Diversifikation, 45
 draw, 34
 dreidimensional, 10
 DSI, 22
 Dynamic Financial Analysis, 73
 Dynamic Invocation Interface, 22
 Dynamic Skeleton Interface, 22
 Dynamisches Laden, 23

 E–Learning, 2, 31, 33, 43
 E–Schritt, 111, 112, 185
 E–Step, 185
 Effekte, 56
 eigene Bedürfnisse, 18
 Eigenschaften, 5, 6
 Eigenschaften der Projektion, 13
 Eigenschaften des Bildraumes, 35
 Eigenschaften des Display–Server, 35
 Eigenschaften Risikomaß, 58
 elliptische Copula, 139
 elliptische Symmetrie, 139
 elliptische Verteilungen, 59, 101
 Elternschnittstelle, 37
 EM–Algorithmus, 109, 112, 115, 184
 EM–Schätzer, 108
 empirische Copula, 125, 147
 empirische ES, 63

empirische Kurtosis, 70
 empirische Quantilfunktion, 132
 empirische Tail-Dependence, 126
 empirische TCE, 62
 empirische VaR, 62
 empirische Varianz, 178
 empirische Verteilung, 64
 empirische Verteilungsfunktion, 132
 empirischen Tail-Dependence-Funktionen, 146
 empirischen Verteilungsfunktion, 132
 Empirischer Expected Shortfall, 63
 empirischer Kendall's τ , 102, 133
 empirischer Mittelwert, 178
 empirisches Pearson's ρ , 99
 empirisches Spearman's ρ , 100
 Entfernen von verdeckten Kanten und Linien, 35
 Entfernen von verdeckten Linien, 18
 Entwurfsmuster, 39, 163
 Erwartungsschritt, 185
 erwartungstreuer Schätzer, 178
 Erwartungswert, 53
 Erzeuger, 106, 142
 Erzeugung eines Objektes, 39
 ES, 61
 Event Service, 24
 Event-Objekt, 40
 Eventliste, 40
 Events, 17–19, 31, 40, 42, 43
 Excel, 26
 Expected Shortfall, 61
 Externalization Service, 24
 Extremwerte, 54
 Exzedenten-Verteilungsfunktion, 60
 Exzess-Verteilungsfunktion, 60
 Factory, 17, 34, 38, 39, 163
 Farben, 8
 Farlie-Gumbel-Morgenstern-Copula, 136, 147, 198
 Farlie-Gumbel-Morgenstern-Copuladichte, 199
 fehlende Daten, 109
 Fensterbreite, 7
 Fenstermanager, 35
 Fernsehen, 1
 Finanzdaten, 8, 73, 107
 Finanzdatenanalyse, 4
 Finanzsektor, 1
 Finanzwerte, 45, 47, 55, 56, 64, 98
 Flanken, 52, 53, 64, 71, 103, 144
 Flankenabhängigkeit, 54
 Flexibilität, 20, 151
 Floor, 45
 Fréchet-Copula, 196
 Fréchet-Hoeffding Schranken, 122
 Fréchet-Randverteilungen, 105
 Frank-Copula, 134, 200
 Frank-Copuladichte, 201
 Funktionale, 5, 15
 Galambos-Copula, 203
 Galambos-Copuladichte, 203
 Gamma-Dichte, 66, 108
 Gamma-Funktion, 65, 179
 gamma-verteilt, 74, 111
 Gamma-Verteilung, 66, 108
 Gamma-Verteilung, Dichte, 65
 Gauss-Copuladichte, 187
 Gauss-Copula, 129, 140, 147, 185

Gauss–Copula, CML–Methode, 132
 Gauss–Copula, IFM–Methode, 132
 Gauss–Copula, Simulation, 134
 Gauss–Kern, 92
 Gesamtrisiko, 46
 Geschwindigkeit, 20
 Gewichte, 108
 gewichtetes Stichprobenmittel, 111
 Gewinn, 45
 Gewinn+Verlust–Verteilung, 56, 58,
 64, 97
 Gleichverteilung, 177
 Globally Unique Identifier, 25
 Grafik, 5, 9
 Grafik–Hardware, 30
 Grafikbibliothek, 2, 5, 17, 18
 Grafiken, 1
 Grafikfactory, 164
 Grafikobjekte, 9,10, 11–14, 16–18, 34,
 37, 38, 42
 Grafikprimitiven, 9, 16, 18, 19, 30,
 35, 36, 235
 grafische Datenverarbeitung, 9
 Graustufen, 8
 GUID, 25
 Gumbel–Copula, 137, 202
 Gumbel–Copula, Simulation, 137
 Gumbel–Copuladichte, 202
 Gumbel–Exponential–Copula, 202
 Gumbel–Exponential–Copuladichte, 202

H–Volumen, 121
 Hedging, 45
 High–Level–Design, 3, 20
 highleveldesign, 15
 Hill–Schätzer, 105,106

 Histogramm, 6
 His
H–Volumen, 121
 historische Daten, 47
 historische Daten, 47, 64
 homogene Koordinaten, 158, 159
 HTTP, 23
 Huessler–Reiss–Copula, 204

 IDL, 21, 23
 IDL–Compiler, 21, 37
 IDL–Syntax, 21
 IFM–Methode, 131, 132
 IIOP, 22
 Implementation, 22, 25
 Implementierung, 3, 21
 Implementierungsdetails, 34
 Industriestandard, 30
 Inference–Functions–Margins–Methode,
 131
 Informatik, 9
 Informationszeitalter, 1
 Inter–ORB Protocol, 22
 Interaktion, 2, 17, 18, 27, 31, 40
 interaktiv, 43
 interaktiver grafischer Benutzerumge-
 bung, 31
 Interaktivität, 33
 Interface, 21
 Interface Definition Language, 21
 Internet, 1, 22
 Inte
H–Volumen, 121
 Iqrquartil Range, 6
 Intranet, 2, 31
 Invarianz der Copula, 122
 IQR, 6

JacORB, 33
 JAVA, 21, 23, 26, 33
 JAVA Virtual Machine, 33
 JAVA–Client, 31
 JAVA
 H–Volumen, 121–Klassen, 23
 JAVA–Objekte, 23
 java.rmi.*, 23
 Joe–Clayton– Copuladichte, 206
 Joe–Copula, 205

Kameratransformationen, 30
 Kendall’s τ , 100, 102, 126
 Kendall’s τ , Copulas, 102
 Kendall’s τ , Eigenschaften, 101
 Kennnummer, 25
 Kern, 7, 77, 93
 Kern–Dichteschätzer, 7
 Kerndichte, 6, 7, 117
 Klasse, 16, 39
 Klassenfabriken, 27
 Klasseninformationen, 23
 Klonen, 34, 38
 Kolmogorovabstand, 141
 Kommunikation, 16–23, 26, 30, 36, 40
 Kommunikationsprotokoll, 22
 Komponenten, 3, 15, 16, 18, 20, 21, 25, 30, 33
 Koordinatenachse, 159
 Koordinatensystem, 7, 11, 17, 36
 Koordinatenursprung, 12
 Korrelationskoeffizient, 8, 99, 177
 Kovarianz, 80
 Kovarianz der absoluten Renditen, 83

Kovarianz–Matrix, 99, 107, 109, 111, 134, 135, 141
 Kreis, 9
 Kreisausschnitt, 9
 Kreisdiagramm, 5, 6
 Kurswerte, 47
 Kurtosis, 51, 69
 Kurtosis der Normalverteilung, 51

Langsame Variation, 180
 Language Mapping, 21, 23
 Laufzeit, 23, 163, 164
 leptokurtisch, 52
 Licensing Service, 24
 Licht, 30
 Life Cycle Service, 24
 Likelihood–Funktion, 110, 128
 linearer Korrelationskoeffizient, 99
 Linie, 9
 Liniestyles, 8
 Linux, 23
 log–Likelihood–Funktion, 109, 110, 184
 log–Renditen, 49, 58, 59, 64, 73
 logarithmische Rendite, 49
 Logistic–Extreme–Copula, 190
 Logistic–Extreme–Copuladichte, 191
 Lokationsmaß, 6
 Lokationsparameter, 178
 low–level API, 27
 low–level Standard, 30

M–Schritt, 111–113, 185
 M–Step, 185
 Manager, 38–40, 163
 Mardia–Copula, 196

Markowitz, 45, 98
 Marktpreis, 56
 Marshall–Olkin–Copula, 201
 Marshall–Olkin–Copuladichte, 201
 Mathematische Grundlagen, 177
 Matrixdarstellung der WVP, 158
 Maus–Event, 40
 Maximierungsschritt, 185
 Maximum–Anziehungsbereich der Fréchet–
 Verteilung, 68, 180
 Maximum–Likelihood–Methode, 128
 Maximum–Likelihood–Schätzer, 109,
 184
 MDA der Studentschen Verteilung, 68
 Median, 6
 Medizin, 1
 Mehrfachvererbung, 165
 Meteorologie, 1
 Methoden, 34, 37, 40
 Methodenaufrufe, 39
 MICO, 34
 Microsoft, 25, 27, 31
 Microsoft Interface Definition Language,
 25
 Middleware, 3, 20, 21, 30, 33
 midl, 25
 Mischung, 66, 67, 74, 108, 179
 Mittelwertsvektor, 111
 MLE der Clayton–Copula, 129
 MLE der Gauss–Copula, 129
 MLE der Student–Copula, 130
 Modellbildung, 47
 Momentenschätzer, 70
 Morgenstern–Copula, Simulation, 136
 MTS, 27
 Multimedia, 33
 multimediale Erweiterung, 27
 multivariate Studentsche Verteilung,
 108
 multivariate t–Verteilung, 108
 multivariate Verteilung, 107
 multivariate Verteilungsfunktion, 120
 multivariates Modell, 47, 65, 97, 107
 name(), 34, 38
 Naming Service, 24
 Netzwerk, 21
 Netzwerkfähigkeit, 27
 Neuberechnung der Daten, 19
 Newton–Raphson Methode, 184
 Newton–Raphson Verfahren, 109
 nichtparametrischen Schätzer, 7
 nichtvollständige Daten, 184
 Normalverteilung, 66, 99, 108
 Normalverteilung, Dichte, 178
 Normalverteilung, Eigenschaften, 178
 Normalverteilungsmodell, 50, 64, 73
 Notification Service, 24
 Nullhypothese, 142
 Object Management Group, 21
 Object Request Broker, 21, 38, 43
 Object Serialization, 25
 Object Transaction Service, 25
 Objekt–Schnittstelle, 22
 Objekt–Server, 3, 14, 16–19, 33–36,
 38–40, 42, 43, 164
 Objektadapter, 21
 Objektaktivierung, 21
 Objektdeaktivierung, 21
 Objekte, 12, 21, 23, 25, 35, 39

Objekteigenschaften, 10
 Objekterzeugung, 163, 164
 objektorientiert, 16, 20, 23, 163
 objektorientierte statistische Programmiersprache, 31
 objektorientierte Struktur der Grafikobjekte, 37
 objektorientierten Aufbau, 21
 objektorientiertes Design, 16
 Objektreferenz, 22
 Objektverwaltung, 163
 OLE, 25
 OMG, 21
 OpenGL, 30–32
 ORB, 21, 38
 ordinale Merkmale, 6
 Ordnungsstatistik, 61, 125

 P+L–Verteilung, 55, 56, 64
 Parameter, 5, 19, 40
 Parameter der Copulas, 215
 Parameterwahl, 41
 Pearson's ρ , 99
 Persistent Object Service, 24
 Physik, 1
 Plackett–Copula, 192
 Plackett–Copuladichte, 193
 Plot, 5, 11, 18, 34, 38
 Plot–Objekt, 35, 165
 Polygon, 9
 Portfolio, 45, 47, 55, 57, 58, 97, 107, 119
 Portfolio Selection, 45
 Portfolio–Theorie, 98
 Positive Homogenität, 58
 Positivität, 58

 Preis, 55
 Primitiven, 17
 Printmedien, 1
 Profit+Loss–Verteilung, 64
 Programmiersprachen, 3, 20, 21, 23, 26, 33, 34
 Programmierstrukturen, 163
 ProjektionEB, 13, 14, 30, 35, 160
 Projektionsebene, 160
 Projektionskoordinaten, 160
 Property Service, 24
 Protokoll, 30
 Prototyp, Design Pattern, 164
 Prototyp–Manager, 164
 Prototypem, 3, 33, 34, 36, 38, 163
 Proxie, 22, 23, 26
 Punkt, 9
 Python, 26

 Quadranten, 145
 Quantilfunktion, 62, 122, 177
 Query Service, 24

 Randverteilungen, 98, 119, 120, 131
 Randverteilungen der Aktien, 116
 Räumliche Darstellung, 8
 Räumliches Sehen, 8
 Ray–Tracing, 30
 rechtshändiges Koordinatensystem, 160
 Referenz, 25, 35, 39
 Reflexionsmodelle, 30
 Remote Method Invocation, 23
 Remote–Objekt, 25
 Renditen, 48, 50, 55, 64
 Request, 16
 Risiko, 45, 47, 50, 55, 107

Risikoanalyse, 55
 Risikoberechnung, 119
 Risikodiversifikation, 58
 Risikokalkulation, 47
 Risikomaß, 58, 64, 98
 Risikomessung, 73, 97
 Risikominimierung, 98
 RMI, 23
 RMI-Registry, 23
 Rotation, 18, 35, 40, 41, 159, 160

 Scatterplot, 5, 7, 19, 47
 Schätzer, 5, 31, 98, 184
 Schätzer für Tail-Dependence $\bar{\chi}$, 106
 Schiefe, 54
 Schnittstelle, 16, 17, 21, 23, 25, 30, 37, 39
 Schnittstellen, 23
 SCM, 26
 Security Service, 25
 Semantik, 22
 Servant, 22, 37, 38, 43
 Server, 16, 26, 43
 Server-Komponenten, 26
 Service Control Manager, 26
 setData, 34, 38, 164
 Sicherheit Dienste, 27
 Sicherheitsrichtlinien, 26
 Simulation Cook-Johnson-Copula, 135
 Simulation einer Copula, 133
 Simulation Gauss-Copula, 134
 Simulation Morgenstern-Copula, 136
 Simulation Student-Copula, 135
 Simulationsalgorithmus, 91
 Singularität der Copula, 124
 Skalenparameter, 178

 Skalierung, 12, 158
 Skeleton, 22, 23
 Skew, 54, 69
 Sklar's Theorem, 121
 Slider, 42
 Smalltalk, 23
 Sound, 27
 Spearman's ρ , 100, 126
 Spezielle Algorithmen zur Stichprobenerzeugung, 134
 Sprung-Kern, 93
 SSL, 19
 Standard-Normalverteilung, 178
 Standard-Normalverteilung, Dichte, 178
 standardnormal-verteilt, 74
 Startschätzer, 185
 statische Plots, 31
 statistische Algorithmen, 1
 statistische Eigenschaften, 5, 10
 statistische Funktionale, 9
 statistische Modelle, 41, 43
 statistische Plots, 8, 33, 34
 statistische Software, 31
 statistische Visualisierungen, 1, 3-5, 9, 10, 15-17, 33
 statistischer Algorithmus, 5
 statistischer Plot, 5, 9, 10, 13, 15-18, 32, 36, 41
 stetige Verteilung, 7
 Stichprobenkovarianzmatrix, 111
 Stichprobenmittelwert, 178
 Stichprobenvarianz, 100, 178
 stochastische Volatilitätsserie, 74
 Stochastischer Prozess, 182
 Strategie, 45

Streuungsmaß, 6
 Struktur der Grafikobjekte, 37
 Stub, 22, 23, 25
 Student–Copula, 142, 188
 Student–Copula, rekursiver Schätzer
 , 132
 Student–Copula, Schätzer ρ , 133
 Student–Copula, Simulation, 135
 Student–Copuladichte, 188
 Studentsche Dichte, 65
 Studentsche Modellierung, 64
 Studentsche Verteilung, 53, 64–66, 73,
 97, 107, 110, 184
 Studentsche Verteilung, multivariate
 Dichte mit einem Alpha, 107
 Studentsche Zeitreihe, 65, 73
 Studentsches Zeitreihenmodell, 73
 Stylized Facts, 47, 56, 64, 73, 94
 Sub–Copula, 120
 Subadditivität, 58, 63
 Survival Copula, 121
 Survivorfunktion, 121

 tail, 52
 Tail Conditional Expectation, 60
 Tail–Dependence χ , 104
 Tail–Dependence $\bar{\chi}$, 105
 Tail–Dependence–Funktion, 144
 TCE, 60
 TCP/IP–Protokoll, 30
 TDisplayServerDragEvent, 40, 41
 TDisplayServerEvent, 40, 41
 TDisplayServerShutdownEvent, 41
 TDisplayServerSliderChangedEvent,
 41
 Test auf Gauss–Copula, 140

 Tests, 31
 Textelement, 9
 Texturen, 30
 TGraphObject2D, 37
 TGraphObject3D, 37
 Theorem von Sklar, 120
 Three–Tier–Lösung, 15
 Time Service, 25
 Tools, 23
 Trading Service, 25
 Transaktionen, 27
 Transformationen, 12, 30
 Transformationssatz, 66
 Transformationssatz für Dichten, 179
 Translation, 12, 157
 Translationsinvarianz, 58
 Typüberprüfung, 40

 UML, 163
 Umsetzung, 20, 30, 33
 Unabhängigkeit, 131
 unbekanntes α , 112
 Unified Modelling Language, 163
 univariate Modelle, 47, 65
 univariate Modellierung, 64
 Unix, 23
 unkorreliert, 49
 Unkorreliertheit, 56

 Value–at–Risk, 59
 VaR, 59
 VaR(T,q), 59
 Varianz, 50, 51, 98
 VBA, 26
 Verbindung, 19
 Vererbung, 16, 40

Verlust, 45, 57
 Verlust–Verteilung, 55, 59, 60, 98
 Verteilung, 6
 Verteilungsfunktion, 120, 126
 verteilungstransparent, 21
 Verteilungstransparenz, 20
 Vertrag, 16
 Verwaltung, 35, 39
 Verwaltung der Grafikobjekte, 34
 Verwaltung von Systemkomponenten,
 25
 Viewport, 11, 12, 158
 Virtual Machine, 23
 Visualisierungen, 8, 43
 VM, 23
 Volatilität, 48, 50
 Volatilitätscluster, 65, 73
 vollständige Daten, 184

Web, 15, 31
 Web–Anwendung, 2
 Web–Browser, 3
 Web–Seiten, 33
 Weltkoordinatensystem, 10, 12
 Weltraum, 10, 12, 17, 34, 35, 157
 Werteverlauf, 48
 Window, 10, 12
 Window–Viewport–Transformation, 11–
 13, 17, 19, 36, 157, 160
 Windows, 23
 Windows–Plattformen, 25
 Windows–Registry, 26
 WSH, 26
 WVP, 13, 19
 www.omg.org, 23

XploRe, 31

Zeitabhängigkeit, 73
 Zeithorizont, 75
 Zeitinformation, 64
 Zeitreihe, 8, 48, 56, 64, 183
 Zentralprojektion, 160, 162
 zweidimensional, 10
 zweidimensionale Copula, 121
 zweidimensionale Punkte, 7