

# Modellierung und Simulation Systemischer Risiken mittels Probabilistischer Programmierung auf Basis eines integrativen Rahmenmodells

Von der Fakultät für Wirtschafts- und Sozialwissenschaften der Universität  
Stuttgart zur Erlangung der Würde eines Doktors der Philosophie  
(Dr. phil.) genehmigte Abhandlung

Vorgelegt von  
Jens Kouros  
aus Tübingen

**Hauptberichter:** Prof. Dr. Ortwin Renn

**Mitberichter:** Prof. Dr.-Ing Dr.h.c. Dr.h.c. Prof.E.h.  
Michael M. Resch

**Tag der mündlichen Prüfung:** 17.10.2019



## Danksagung

An dieser Stelle möchte ich mich bei all denen bedanken, die mich während der Anfertigung meiner Dissertation unterstützt haben.

Zunächst gebührt mein Dank Herrn Prof. Dr. Ortwin Renn, der meine Arbeit betreut hat. Ohne seine Förderung und das Vertrauen, das er von Anfang an in mich und mein Dissertationsvorhaben gesetzt hat, hätte diese Arbeit nicht entstehen können. Die Offenheit, die er mir und meinen Ideen entgegen gebracht hat, ist nicht selbstverständlich. Seine Zuversicht und Gelassenheit haben mir besonders über die schwierigen Phasen hinweg geholfen. Für seine hilfreichen Anregungen und die konstruktive Kritik möchte ich mich herzlich bedanken.

Ebenfalls gilt mein Dank Herrn Prof. Dr.-Ing Dr.h.c. Dr.h.c. Prof.E.h. Michael Resch, dem Leiter des Höchstleistungsrechenzentrums Stuttgart, das während der Arbeit an meiner Dissertation mein Arbeitsplatz war. Dass ich als Doktorand der Sozialwissenschaften in den Genuss eines Arbeitsplatzes an einem Rechenzentrum gekommen bin, ist seiner Offenheit zu verdanken.

Weiterhin gilt mein Dank meinen Abteilungsleitern, Herrn Uwe Küster und seinem Nachfolger Herrn Dr.-Ing Ralf Schneider, die mich in der Abteilung für Numerische Methoden herzlich willkommen geheißen haben und mir und meiner Arbeit immer aufgeschlossen gegenüber standen. Es war für mich ein Glücksfall, von ihrer Expertise im Bereich der Computersimulation profitieren zu können.

Bedanken möchte ich mich weiterhin bei vielen Kollegen und Freunden, die mich auf ganz unterschiedliche Weise unterstützt haben. Da wären zunächst Dr. Sophia Becker, Dr. Birgit Mack und Karolin Tampe-Mai, die mir den Einstieg in die sozialwissenschaftliche Forschungsarbeit ermöglicht haben und so den Grundstein für meine weitere Laufbahn gelegt haben; Frederick Roth und Till Lorentzen, die eine große Hilfe in Fragen bezüglich der Software Entwicklung waren; Michael Herrmann, Nico Formanek, Hiltrun Lampe, Alena Umbach und Andreas Kaminski, mit denen ich zahlreiche Gespräche und Diskussion zu verschiedenen Aspekten meiner Dissertation und dem wissenschaftlichen Arbeit insgesamt führen durfte; Philip Offenhäuser und Dmitry Khaby, die eine angenehme Arbeitsatmosphäre in unserem gemeinsamen Büro haben entstehen lassen; bei allen anderen, die ich namentlich noch nicht genannt habe und die mir auf meinem Weg geholfen haben.

Mein letzter und größter Dank gilt meiner Familie. Dass ich überhaupt einen Weg einschlagen konnte, der zur Anfertigung einer Dissertation führt, habe ich meinen Eltern, Gabriele und Bijan Kouros, zu verdanken. Sie haben mir außerdem ganz konkret geholfen, in dem sie das gesamte Manuskript durchgesehen und an vielen Stellen wertvolle Hinweise zur Verbesserung gegeben haben. Meiner Frau, Kornelia

Timm, danke ich dafür, dass sie mich in allem, was ich tue, unterstützt. Die Anfertigung einer Dissertation ist mitunter eine zeitintensive Aufgabe, die nicht viel Raum für anderes lässt. Ohne ihr Verständnis und ihre Geduld wäre es nicht möglich gewesen, die Dissertation anzufertigen. Und schließlich danke ich meinem Bruder Kristian Kouros und seiner Frau Burcu Zimmerling für die vielen fachlichen und nicht-fachlichen Gespräche, die immer eine große Quelle der Inspiration waren.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>I</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Zusammenfassung</b>	<b>VII</b>
<b>1 Einleitung</b>	<b>1</b>
<b>I Systemische Risiken aus theoretischer Perspektive</b>	<b>7</b>
<b>2 Überblick über verschiedene Ansätze zur Betrachtung Systemischer Risiken</b>	<b>9</b>
2.1 Ökonomie und Finanzwissenschaften . . . . .	10
2.2 Risk Governance . . . . .	13
2.3 Soziologie . . . . .	15
2.4 Weitere Definitionen . . . . .	16
<b>3 Wiederkehrende Aspekte bei der Betrachtung Systemischer Risiken</b>	<b>19</b>
3.1 Globalität . . . . .	20
3.2 Dynamische Wechselwirkungen . . . . .	21
3.3 Ansteckung und Ausbreitung . . . . .	22
3.4 Dynamische Risiken . . . . .	24
<b>4 Entwicklung eines Rahmenmodells für Systemische Risiken</b>	<b>27</b>
4.1 Differenzierung der Merkmale nach Betrachtungsebenen . . . . .	27
4.1.1 Ebenen der Betrachtung . . . . .	27
4.1.2 Differenzierung der Merkmale . . . . .	29
4.2 Integration der Merkmale durch Abstraktion . . . . .	30
4.2.1 Abstraktion . . . . .	31
4.2.2 Rekursion als abstraktes Merkmal . . . . .	31

4.3	Wesentliche Merkmale Systemischer Risiken . . . . .	32
4.3.1	Globalität . . . . .	32
4.3.2	Rekursivität . . . . .	33
4.3.3	Dynamik . . . . .	33
4.4	Formulierung des Rahmenmodells . . . . .	33
4.5	Systemische Risiken und Resilienz . . . . .	35
 <b>II Modellierung Systemischer Risiken mittels Probabilistischer Programmierung</b>		<b>39</b>
<b>5</b>	<b>Die Methode der Probabilistischen Programmierung</b>	<b>41</b>
5.1	Formale Repräsentation von Wahrscheinlichkeitsverteilungen . . . . .	43
5.2	Inferenz . . . . .	44
5.3	Expressivität . . . . .	45
5.4	Einordnung ins Ebenenmodell . . . . .	46
<b>6</b>	<b>Fallbeispiel 1: Ausbreitung von Influenza</b>	<b>47</b>
6.1	Allgemeine Beschreibung des SIR-Modells . . . . .	48
6.2	Beschreibung des probabilistischen SIR-Modells . . . . .	50
6.2.1	Entwicklung des I-Kompartiments . . . . .	51
6.2.2	Entwicklung des R-Kompartiments . . . . .	58
6.2.3	Entwicklung des S-Kompartiments . . . . .	59
6.3	Beschreibung der empirischen Daten . . . . .	59
6.4	Auszüge aus dem Probabilistischen Programm . . . . .	61
6.5	Ergebnisse . . . . .	65
6.5.1	Simulation ohne Einbindung von Daten . . . . .	65
6.5.2	Simulation mit Einbindung von Daten . . . . .	68
6.6	Diskussion . . . . .	74
<b>7</b>	<b>Fallbeispiel 2: Soziale Diffusion und Fake News</b>	<b>77</b>
7.1	Beschreibung des Modells . . . . .	79
7.2	Ergebnisse . . . . .	81
7.2.1	Simulation auf Basis eines Newman-Watts-Graphen . . . . .	81
7.2.2	Simulation auf Basis eines Barabási-Albert-Graphen . . . . .	87
7.3	Diskussion . . . . .	90
<b>8</b>	<b>Fallbeispiel 3: Soziale Ungleichheit und vernetzte Risiken</b>	<b>95</b>
8.1	Beschreibung des Modells . . . . .	96
8.2	Ergebnisse . . . . .	97
8.3	Diskussion . . . . .	106

<b>Inhaltsverzeichnis</b>	<b>III</b>
<hr/>	
<b>9 Diskussion</b>	<b>111</b>
<b>Literaturverzeichnis</b>	<b>119</b>
<b>A Quellcode zu Kapitel 6</b>	<b>129</b>
<b>B Quellcode zu Kapitel 7</b>	<b>161</b>
<b>C Quellcode zu Kapitel 8</b>	<b>177</b>
<b>Eigenständigkeitserklärung</b>	<b>188</b>



# Abbildungsverzeichnis

4.1	Unterschiedliche Ebenen bei der Betrachtung Systemischer Risiken . . .	36
4.2	Grafische Darstellung des Rahmenmodells zu Systemischen Risiken . . .	37
6.1	Kompartimente des SIR-Modells . . . . .	49
6.2	Progressionen von Kohorten . . . . .	50
6.3	Histogramme für Poisson-Verteilungen mit Parameter 1.0 bzw. 2.0. . .	54
6.4	Histogramme der A-priori Verteilungen von $\lambda_1$ und $\lambda_2$ . . . . .	55
6.5	Grafik aus dem Saisonbericht 2015/16 des RKI . . . . .	60
6.6	Ergebnisse von 10000 Simulationen des SIR-Modells ohne Daten . . .	67
6.7	Primäre und sekundäre Neuinfektionen nach 10000 Simulationsläufen ohne Daten . . . . .	68
6.8	Verteilungen der Neuinfektionen pro Woche ohne Daten . . . . .	69
6.9	Verteilungen der Neuinfektionen pro Woche ohne Daten . . . . .	70
6.10	Ergebnisse von 10000 Simulationsläufen des SIR-Modells mit Daten . .	71
6.11	Primäre und sekundäre Neuinfektionen nach 10000 Simulationsläufen unter Einbezug von Daten . . . . .	72
6.12	Verteilungen der Neuinfektionen pro Woche unter Eibezug von Daten . .	73
6.13	A-posteriori Verteilungen der primären und sekundären Ansteckungs- parameter $\lambda_1$ und $\lambda_2$ . . . . .	74
7.1	Newman-Watts-Graph mit 500 Knoten . . . . .	82
7.2	Verteilung der Knotengrade im Newman-Watts-Graphen . . . . .	84
7.3	Ausbreitung von Fake News im Newman-Watts-Graphen . . . . .	85
7.4	Verteilungen der Ausbreitung von Fake News im Newman-Watts- Graphen nach Zeitschritten . . . . .	86
7.5	Barabasi-Albert Graph mit 500 Knoten . . . . .	88
7.6	Verteilung der Knotengrade im Barabási-Albert-Graphen . . . . .	89
7.7	Ausbreitung von Fake News im Barabasi-Albert Graphen . . . . .	90
7.8	Verteilungen der Ausbreitung von Fake News im Barabási-Albert- Graphen nach Zeitschritten . . . . .	91

8.1	Initialisierter Newman-Watts-Graph. . . . .	98
8.2	Verteilung der Knotengrade von Sim-1 . . . . .	99
8.3	Verteilung der Knotengrade nach Sim-1 . . . . .	100
8.4	Verteilung des Guts vor und nach Sim-1 . . . . .	101
8.5	Graph nach Sim-1 . . . . .	102
8.6	Initialisierter Barabási-Albert-Graph. . . . .	103
8.7	Verteilung der Knotengrade vor Sim-2 . . . . .	104
8.8	Verteilung des Guts vor und nach Sim-2 . . . . .	105
8.9	Verteilung der Knotengrade nach Sim-2 . . . . .	106
8.10	Graph nach Sim-2 . . . . .	107

## Abstract

Systemic risk is investigated from two perspectives. The first part of this work reviews different conceptions and definitions from different fields, revealing a diverse picture. A number of reoccurring aspects is identified and taken as the basis for an overarching framework for systemic risk. The framework is qualitative in that it does not specify any mathematical formalization. This abstract nature enables it to integrate the different conceptions found in the literature. Furthermore it offers a way to distinguish conventional and systemic risk.

In the second part three examples for systemic risk are identified based on this framework. For each of these, a computer simulation is developed by means of probabilistic programming, a development from the fields of machine learning and artificial intelligence. The use of probabilistic programming for modelling and simulating systemic risk is a focus of this work. The first two case studies deal with influenza and epidemics, and with social diffusion and the spreading of *fake news*. In the third case study, a rudimentary model for social inequality is developed, in which the structure of a graph is changed as a consequence of the interaction of two different mechanisms.

## Zusammenfassung

Die vorliegende Arbeit beschäftigt sich aus zwei unterschiedlichen Perspektiven mit Systemischen Risiken. Im ersten Teil werden verschiedene Ansätze und Definitionen Systemischer Risiken aus verschiedenen Disziplinen zusammengetragen. Aus diesen zum Teil sehr unterschiedlichen Ansätzen werden einige häufig auftretende Merkmale extrahiert und nach einem weiteren Abstraktionsschritt zu einem übergeordneten Rahmenmodell in Beziehung gesetzt. Es handelt sich um ein qualitatives Modell, bei dem bewusst auf eine konkrete formale Ausgestaltung verzichtet wurde. Der abstrakte Charakter des Rahmenmodells ermöglicht die Integration der verschiedenen Ansätze und Definitionen Systemischer Risiken. Des Weiteren wird durch die Formulierung eines übergeordneten Rahmenmodells eine Abgrenzung zwischen konventionellen und Systemischen Risiken möglich.

Auf Basis des integrativen Rahmenmodells wurden im zweiten Teil der Arbeit drei Beispiele für Systemische Risiken identifiziert, für die mit Hilfe der Probabilistischen Programmierung Computersimulationen entwickelt wurden. Die Probabilistische Programmierung ist eine Entwicklung aus der künstlichen Intelligenzforschung, die in den letzten Jahren verstärkt Interesse erfährt. Die Verwendung dieser Methode zur Modellierung und Simulation Systemischer Risiken stellt einen Fokus dieser Arbeit dar. Die ersten zwei Beispiele befassen sich mit der Ausbreitung von Influenza und einer möglichen Epidemie sowie mit dem Konzept der sozialen Diffusion, das am Beispiel der Ausbreitung von *Fake News* behandelt wird. Im dritten Fallbeispiel wird ein rudimentäres Modell zur sozialen Ungleichheit entwickelt, in dem sich ein soziales Netzwerk strukturell aufgrund der Interaktion zweier verschiedener Mechanismen verändert.

# Kapitel 1

## Einleitung

Risiken lauern überall. Sich mit ihnen auseinanderzusetzen, ist eine Notwendigkeit. Das gilt sowohl im Privatleben als auch in der Politik, wo Entscheidungen von großer Reichweite getroffen und Weichen für die Zukunft gestellt werden. Doch während diese Auseinandersetzung im Privaten oft implizit und unbewusst ist, müssen Risiken und Chancen in Bezug auf politische Entscheidungen sorgfältig abgewogen werden. Und auch wenn die Feststellung trivial ist, die Aufgabe ist es nicht. Denn die subjektive Wahrnehmung von Risiken stimmt bei weitem nicht immer mit der Realität überein (siehe z.B. Renn, 2015) und kann außerdem kognitiven Verzerrungen unterliegen (Tversky & Kahneman, 1981).

Aus diesem Grund hat sich in der wissenschaftlichen Politikberatung ein Feld entwickelt, welches das Verständnis um Risiken fördern und so den Umgang mit ihnen erleichtern soll. Unter dem Namen *Risk Governance* werden dabei vor allem institutionalisierte Anstrengungen des Risikomanagements zusammengefasst, die in Bezug auf sogenannte *konventionelle* Risiken in vielen Bereichen große Erfolge verzeichnen können. So hat sich beispielsweise der Prozentsatz der Todesfälle am Arbeitsplatz pro Jahr seit den 1960er Jahren um etwa 90% verringert (Renn, 2016).

Spätestens seit Veröffentlichung eines Berichts der OECD (2003) wird eine neue Klasse von Risiken diskutiert, die als *Systemische Risiken* bezeichnet werden. Was genau unter Systemischen Risiken zu verstehen ist, wird in den folgenden Kapiteln noch ausführlich diskutiert. Als Beispiele, die zugleich die Relevanz deutlich machen, seien an dieser Stelle zunächst Finanzkrisen, Epidemien oder der Klimawandel ge-

nannt. Es ist klar, dass diese Phänomene “uns alle” betreffen. Es geht um globale Systeme mit einem hohen Grad an Vernetzung, um nicht-lineare Wechselwirkungen und um die Möglichkeit des Zusammenbruchs dieser Systeme, von denen das Funktionieren oder gar der Fortbestand unserer Gesellschaft abhängt. Man kann sich leicht vorstellen, dass der Umgang mit solchen Risiken noch schwieriger ist, als das schon bei den konventionellen Risiken der Fall war. Bereits Dörner (1989) hat eindrucksvoll die Probleme beim Planen und Entscheiden in komplexen Systemen aufgezeigt, in denen nicht-lineare dynamische Wechselbeziehungen am Werke sind. Das Risikomanagement sieht sich hier mit ganz neuen Herausforderungen konfrontiert, was sich unter anderem in der Entstehung einer Institution wie dem *International Risk Governance Council (IRGC)* zeigt, die sich der Förderung des Verständnis und des Managements von Systemischen Risiken verschrieben hat.

Das Ziel einer solchen Institution ist es letztlich, der Politik Handlungsempfehlungen zu geben, um durch ein Eingreifen in ein System Systemische Risiken möglichst gering zu halten oder ganz zu beseitigen. Ideal wäre es, könnte man zu diesem Zweck “Stellschrauben” benennen, mit denen das Verhalten eines Systems gesteuert werden kann. Das Wissen darum, wann ein Eingriff nötig ist, würde ein Frühwarnsystem liefern, das immer dann anschlägt, wenn das System droht, “aus dem Ruder zu laufen”. Dazu müssen die kausalen Wirkmechanismen innerhalb des Systems bekannt sein, was im Falle von sozialen Systemen, bzw. in Systemen, die unter anderem auch einen sozialen Aspekt haben, für gewöhnlich nicht der Fall ist<sup>1</sup>. Nun könnte man einwenden, dass es zu diesem Zweck ja die Sozialwissenschaften gibt, deren Gegenstand doch gerade das Erforschen solcher Zusammenhänge ist. Wenn die zu erforschenden Systeme aber globaler Natur sind, dann sieht sich die empirische Sozialforschung mit einem Problem konfrontiert. Denn der Königsweg der empirischen Forschung, das Experiment, lässt sich hier nicht anwenden. Das hat schon ganz praktische Gründe, aber selbst wenn diese gelöst werden könnten, ein Experiment am Finanz- oder Klimasystem verbietet sich ganz offensichtlich.

Ein Ausweg, so scheint es, besteht in der Simulation der Systeme. Und tatsäch-

---

<sup>1</sup>Gemeint sind etwa soziotechnische oder sozial-ökologische Systeme. An dieser Stelle sei der Diskussion vorgegriffen, ob Systemische Risiken immer auch eine soziale Komponente haben. Der Überblick über die Literatur, der später gegeben wird, legt dies nah. Aber selbst wenn man diese Frage verneint: in der vorliegenden Arbeit werden eventuelle Systemische Risiken ohne sozialen Bezug nicht behandelt.

---

lich erlebt die Simulation in den Sozialwissenschaften in der jüngeren Vergangenheit einen Aufschwung. Davon zeugt unter anderem das Erscheinen des “Handbuch Modellbildung und Simulation in den Sozialwissenschaften” (Braun & Saam, 2015), aber auch ein Sammelband zur Simulation in den Sozialwissenschaften, der zum Zeitpunkt der Erstellung dieses Textes in Arbeit ist und an dem auch der Autor dieser Arbeit beteiligt ist. Die kausalen Strukturen müssen natürlich immer noch gefunden werden. Ist das aber erst geschafft, dann können die wichtigen Elemente und Zusammenhänge in ein Simulationsmodell überführt werden. Man könnte dann einfach die Simulation beobachten und die gewonnenen Erkenntnisse auf das reale System übertragen. Diese Vorstellung ist allerdings so verführerisch wie naiv. Was genau aus Computersimulationen im Allgemeinen gelernt werden kann, ist Gegenstand aktueller Forschung in der Philosophie der Computersimulation (Beisbart, 2017). Was sozialwissenschaftliche Simulationen im Speziellen angeht, so wird der oben erwähnte Sammelband interessante Einblicke über deren Status gewähren. Gottschalk-Mazouz (2012) etwa bezeichnet viele sozialwissenschaftliche Simulationsmodelle als *toy models*, die “schwerlich als Modellierung von Realitätsausschnitten verstanden werden” können (Gottschalk-Mazouz, 2012, S.1). Auch wenn sich wohl Abstufungen im Vergleich unterschiedlicher Simulationen machen lassen, insgesamt ist festzuhalten, dass der Realismus, der mit sozialwissenschaftlichen Simulationen erreicht werden kann, nicht mit dem naturwissenschaftlich orientierter Simulationen vergleichbar ist.

Daraus soll nun aber nicht geschlossen werden, dass man Simulationsmodelle für Systemische Risiken bzw. in den Sozialwissenschaften allgemein besser nicht mehr entwickeln sollte. Auch Gottschalk-Mazouz (2012) stellt die Sinnhaftigkeit sozialwissenschaftlicher Simulationen nicht in Frage, solange anerkannt wird, dass ein Simulationsmodell kein Abbild der Realität ist, sondern eine indirekte Repräsentation eines Phänomens darstellt. Saam (2015) legt dar, dass mit ihnen “grundlegende Defizite im Stand der Forschung in den Sozialwissenschaften [...] adressiert” (Saam, 2015, S.61) und Impulse für zukünftige Forschung gewonnen werden können. Und nicht zuletzt kann und sollte auch versucht werden, sozialwissenschaftliche Simulationsmodelle durch immer bessere theoretische und empirische Fundierung Stück für Stück ein kleines bisschen realistischer zu machen: ein bisschen weniger *toy*, ein bisschen mehr *model*. In diesem Licht sind dann auch die Simulationsmodelle zu verstehen, die in dieser Arbeit vorgestellt werden. Denn es ist klar, dass die Probleme,

die von Saam (2015) und Gottschalk-Mazouz (2012) erläutert werden, auch auf sie zutreffen. Sie sind, das sei in aller Deutlichkeit gesagt, *toy models*, wenngleich sie sich im Verhältnis von *toy* zu *model* unterscheiden. Bis hin zu Stellschrauben und Frühwarnsystemen ist es noch ein weiter Weg. Vielmehr stellen die Modelle Startpunkte der Simulation Systemischer Risiken dar, die in zukünftigen Anstrengungen überarbeitet und verbessert werden müssen.

Dabei war es von Anfang an das Ziel, Möglichkeiten der Modellierung und Simulation Systemischer Risiken zu finden, die auch die Berücksichtigung probabilistischer Zusammenhänge erlauben. Es ist vermutlich unstrittig zu behaupten, dass es deterministische Zusammenhänge in den Sozialwissenschaften<sup>2</sup>, wenn überhaupt, nur selten gibt, und dies gilt auch für Systemische Risiken. Der Luxus von Naturgesetzen, wie sie etwa die Newton'sche Mechanik hervorgebracht hat, wird für die Sozialwissenschaften vielleicht immer ein Traum bleiben. Die Prämisse der vorliegenden Arbeit ist es deshalb, dass Theorien und Modelle in den Sozialwissenschaften immer mit Unsicherheit behaftet sind. Diese Unsicherheit mag je nach Fragestellung größer oder kleiner ausfallen und im besten Fall wird ihr Ausmaß durch Forschung reduziert. Eine seriöse Herangehensweise muss dieser Unsicherheit Rechnung tragen und ein Weg das zu tun, liegt in der Verwendung probabilistischer Methoden. Die Frage, ob soziale Phänomene ihrem Wesen nach probabilistisch sind oder "nur" nicht ausreichend verstanden werden, soll dabei vernachlässigt werden. Als ein besonders vielversprechender Kandidat erschien die *Probabilistische Programmierung*, die dann auch in den Fallstudien dieser Arbeit zur Anwendung kommt.

Bei den Recherchen zu dieser Arbeit hat sich aber früh herausgestellt, dass die Entwicklung eines probabilistischen Modells erst der dritte Schritt sein kann. Der zweite Schritt bestand in der Identifikation geeigneter Beispiele für Systemische Risiken, die im Rahmen einer Promotion handhabbar sind. Diese Feststellung mag trivial erscheinen, ist aber insofern wichtig, als sich bei dieser Suche ein ganz grundsätzliches Problem offenbarte. Denn um geeignete Beispiele zu finden, musste zunächst geklärt werden, was genau denn ein Systemisches Risiko ist. Eine allgemein akzeptierte Definition existiert nicht, wie die Ausführungen in Kapitel 2 zeigen, wohl aber

---

<sup>2</sup>Die Sozialwissenschaften werden in dieser Arbeit in einem weiten Sinn verstanden als alles, was nicht den Naturwissenschaften zuzuordnen ist. Dazu zählen etwa die Soziologie, Psychologie, Ökonomie und Politikwissenschaften.

eine große Zahl an unterschiedlichen Ansätzen und Definitionen, die sich zum Teil beträchtlich unterscheiden. Um entscheiden zu können, ob ein bestimmtes Phänomen ein Systemisches Risiko darstellt, gab es zwei Möglichkeiten. Es wäre möglich gewesen, sich für eine der Definitionen zu entscheiden, die dann Kriterien zur Identifikation liefert. Das würde aber auch bedeuten, alle anderen Definitionen abzulehnen. Eine sorgfältige wissenschaftliche Herangehensweise verlangt dabei selbstverständlich nach einer transparenten und argumentativ begründeten Entscheidung. Das “Aussortieren” so vieler Definitionen erschien aber, um es vorsichtig auszudrücken, als anmaßend. Ganz zu schweigen von dem Problem, die “eine richtige” Definition zu finden.

Aus diesen Gründen wird in der vorliegenden Arbeit ein anderer Weg gegangen. Der erste Teil dieser Arbeit widmet sich dem Versuch, ein übergeordnetes, integratives Rahmenmodell für Systemische Risiken zu entwickeln, mit dem die unterschiedlichen Ansätze zu Systemischen Risiken gewissermaßen “unter einen Hut” gebracht werden können. Die grundsätzliche Vorgehensweise besteht darin, von den speziellen Eigenheiten der unterschiedlichen Ansätze zu abstrahieren und allgemeine Prinzipien oder Merkmale ausfindig zu machen, die allen oder möglichst vielen Ansätzen gemein sind. Diese abstrakten Merkmale können dann zu einem ebenfalls abstrakten Modell zusammengeführt werden. Bei der Betrachtung konkreter Beispiele Systemischer Risiken müssen die abstrakten Merkmale dann wieder durch konkrete, spezifische Phänomene ersetzt werden.

Die Arbeit ist folgendermaßen aufgebaut: In Kapitel 2 wird ein Überblick über verschiedene Definitionen Systemischer Risiken gegeben. Ein erster Abstraktionsschritt wird dann in Kapitel 3 gemacht, wenn verschiedene Merkmale herausgehoben werden, die in den unterschiedlichen Definitionen immer wieder und zum Teil in verschiedenen Kombinationen zum Vorschein kommen. In Kapitel 4 werden diese Merkmale in Bezug auf unterschiedliche Betrachtungsebenen differenziert. Anschließend wird ein weiterer Abstraktionsschritt durchgeführt, wodurch drei Merkmale extrahiert werden können, auf deren Basis dann das integrative Rahmenmodell formuliert wird. Im zweiten Teil der Arbeit werden Simulationsmodelle für drei Beispiele von Systemischen Risiken vorgestellt. Das am weitesten fortgeschrittene Modell behandelt die Ausbreitung von Influenza in Deutschland und wird in Kapitel 6 erörtert. Ein Grund für die Wahl dieses Beispiels war, dass vom *Robert Koch-Institut* Daten

zur Ausbreitung von Influenza erhoben werden, die bei Verwendung der Probabilistischen Programmierung berücksichtigt werden können. Die Methode der Probabilistischen Programmierung wird in Kapitel 5 vorgestellt. In Kapitel 7 wird die Idee der sozialen Diffusion in Zusammenhang mit Systemischen Risiken gebracht. Soziale Diffusion ist ein abstraktes Konzept, das sich auf viele unterschiedliche Beispiele beziehen lässt. Hier wurde die Ausbreitung von *Fake News* gewählt. Im Gegensatz zu dem Modell aus Kapitel 6 ist dieses Modell als rudimentär zu betrachten. Interessant ist es nach Ansicht des Autors trotzdem, unter anderem weil ein Graph als Ausgangspunkt dient. In Kapitel 8 schließlich wird die graphenbasierte Modellierung noch einen Schritt weiter getrieben, dahingehend, dass der Graph sich während der Simulation verändert. Dieses Modell wird in Bezug auf soziale Ungleichheit diskutiert, die dann für die Veränderung der Struktur des Graphen ausschlaggebend ist.

# Teil I

## Systemische Risiken aus theoretischer Perspektive



## Kapitel 2

# Überblick über verschiedene Ansätze zur Betrachtung Systemischer Risiken

Im Folgenden wird ein Überblick über verschiedene Definitionen Systemischer Risiken gegeben. Die einzige fächerübergreifende Zusammenschau, die dem Autor zum Zeitpunkt der Erstellung des vorliegenden Textes bekannt ist, stammt von Neitzke (2007). Seine Arbeit ist trotz des Datums nach wie vor empfehlenswert und das nicht nur, weil sie die einzige ihrer Art ist. Die Ausführungen in diesem Kapitel folgen deshalb in weiten Teilen denen von Neitzke (2007) und werden, wo es notwendig ist, durch die Berücksichtigung von neuerer Literatur ergänzt. Die unterschiedlichen Disziplinen, die sich mit Systemischen Risiken beschäftigen, werden dabei zu den folgenden drei Themenbereichen zusammengefasst: *Ökonomie und Finanzwissenschaften*, *Risk Governance* und *Soziologie*.

Der Überblick soll zum einem interessierten Lesern als Einstieg in die Literatur zu Systemischen Risiken dienen. Er stellt aber auch eine wichtige Vorarbeit für die folgenden Kapitel dieser Arbeit dar. Wie sich zeigen wird, ist die Verwendung der Bezeichnung *Systemische Risiken* alles andere als einheitlich. Unterschiede bestehen sowohl zwischen als auch innerhalb der Disziplinen. Die Literaturrecherche zu Systemischen Risiken offenbart deshalb ein Problem bei der Beschäftigung mit Systemischen Risiken, nämlich die Frage, was genau denn unter einem solchen zu

verstehen ist. Dieses Problem hat auch Neitzke (2007) gesehen und eine Definition von Systemischen Risiken vorgeschlagen, auf die später noch eingegangen wird.

## 2.1 Ökonomie und Finanzwissenschaften

Laut Neitzke (2007) kommt die Bezeichnung *Systemisches Risiko* ursprünglich aus der Ökonomie und wird dort verwendet, um Finanzkrisen zu beschreiben. Die ersten Veröffentlichungen, in deren Titel die Systemischen Risiken vorkommen, stammen aus den 1980er Jahren. Diese frühen Arbeiten entstanden aber laut Dwyer (2009) im Kontext von Politik und Politikberatung. Von der professionellen wissenschaftlichen Ökonomie wurde das Thema erst etwas später aufgegriffen. Eine der ersten Definitionen von Systemischen Risiken stammt von Rochet und Tirole (1996):

Systemic risk refers to the propagation of an agent's economic distress to other agents linked to that agent through financial transactions. (Rochet & Tirole, 1996, S.733)

Eine andere Definition kommt von De Bandt und Hartmann (2000):

We define a systemic event in the narrow sense as an event, where the release of bad news about a financial institution, or even its failure, or the crash of a financial market leads in a sequential fashion to considerable adverse effects on one or several other financial institutions or markets, e.g. their failure or crash. [...] A systemic event in the narrow sense is strong, if the institution(s) affected in the second round or later actually fail as a consequence of the initial shock, although they have been fundamentally solvent ex ante, or if the market(s) affected in later rounds also crash and would not have done so without the initial shock. We denote these strong instances of systemic events in the narrow sense as contagion. [...] Systemic risk can be defined as the risk of experiencing systemic events in the strong sense. (De Bandt & Hartmann, 2000, S.10f)

Beide Definitionen stellen einen direkten Zusammenhang zwischen Systemischen Risiken und einem Mechanismus der Ansteckung her und betonen damit zumindest

implizit die Vernetzung zwischen den Akteuren. Den größten Teil der Definition von De Bandt und Hartmann (2000) nimmt allerdings die Beschreibung bzw. Definition sogenannter *Systemischer Ereignisse* ein, die den Ansteckungsmechanismus in Gang setzen. Ein Systemisches Risiko wird dann definiert als das Risiko, dass ein System ein solches Ereignis erfährt. Der Kern eines Systemischen Risikos steckt nach dieser Definition im auslösenden Ereignis, während bei Rochet und Tirole (1996) die Ausbreitung des “economic distress” selber im Zentrum steht. Vor allem auf die Größenordnung des potentiellen Schadens heben Ferguson et al. (2001) ab:

Systemic financial risk is the risk that an event will trigger a loss of economic value or confidence in, and attendant increases in uncertainty about, a substantial portion of the financial system that is serious enough to quite probably have significant adverse effects on the real economy. Systemic risk events can be sudden and unexpected, or the likelihood of their occurrence can build up through time in the absence of appropriate policy responses. The adverse real economic effects from systemic problems are generally seen as arising from disruptions to the payment system, to credit flows, and from the destruction of asset values. (Ferguson et al., 2001, S.126)

Auch in dieser Definition findet sich ein auslösendes Ereignis, das aber nur insofern näher beschrieben wird, als dass die materiellen und immateriellen Schäden, die durch das Ereignis ausgelöst werden, einen “substantiellen Teil des Finanzsystems” betreffen und damit “ernst genug” sind, um “wahrscheinlich signifikante Effekte auf die Realwirtschaft” zu haben. Das auslösende Ereignis wird nicht näher beschrieben, ist aber dennoch von großer Bedeutung für die Definition von Ferguson et al. (2001). Denn ohne ein solches Ereignis scheint es Systemische Risiken nicht zu geben. Ferguson et al. (2001) bleiben also, wie auch von Dwyer (2009) hervorgehoben wird, insgesamt vage. Interessant ist allerdings die Bemerkung, dass Systemische Risiken unerwartet auftreten können und dass die Wahrscheinlichkeit eines solchen Auftretens sich mit der Zeit verändern kann. Eine weitere Definition findet sich bei Kaufman und Scott (2003):

Systemic risk refers to the risk or probability of breakdowns in an entire system, as opposed to breakdowns in individual parts or components,

and is evidenced by comovements (correlation) among most or all of the parts. (Kaufman & Scott, 2003, S.371)

Das Hauptaugenmerk in dieser Definition liegt, ähnlich wie bei Ferguson et al. (2001), auf der Größe des Schadens. Interessant ist hier, dass nicht zwischen Risiko und Wahrscheinlichkeit eines Zusammenbruchs unterschieden wird. Für Kaufman und Scott (2003) sind dies also einfach unterschiedliche Bezeichnungen. Eine weitere und nach Neitzke (2007) weit verbreitete Definition stammt von Davis (2003):

We prefer to define systemic risk, financial instability or disorder as entailing heightened risk of a financial crisis - ‘a major collapse of the financial system, entailing inability to provide payments services or to allocate credit to productive investment opportunities.’ (Davis, 2003, S.2)

Auch hier liegt der Fokus auf der Reichweite des Schadens, der mit Begriffen der Finanzwirtschaft beschrieben wird, so wie das auch bei einigen anderen Definitionen der Fall war. Eine ganz andere Herangehensweise findet sich bei Hendricks, Kambhu und Mosser (2007). Sie stellen zwar auch einen Zusammenhang zwischen Systemischen Risiken und dem Wirtschafts- und Finanzsystem her. Die eigentliche Definition benutzt aber Begriffe aus der Mathematik.

[...] the key characteristic of systemic risk is the movement from one stable (positive) equilibrium to another stable (negative) equilibrium for the economy and the financial system. (Hendricks et al., 2007, S.65)

Hier werden Systemische Risiken mit dem Begriff des Gleichgewichts in Verbindung gebracht. Dieses Konzept findet vor allem in der Betrachtung nicht-linearer Systeme seinen Einsatz (siehe Strogatz, 2018, für eine Einführung in die Theorie nicht-linearer Systeme). Gänzlich aus ökonomischer Perspektive formuliert dagegen Schwarcz (2008) und bemüht sich um eine detaillierte Sicht, deren zentrales Element wieder ein auslösendes Ereignis darstellt. Seiner Meinung nach ist ein Systemisches Risiko

[...] the risk that (i) an economic shock such as market or institutional failure triggers (through a panic or otherwise) either (X) the failure of

a chain of markets or institutions or (Y) a chain of significant losses to financial institutions, (ii) resulting in increases in the cost of capital of decreases in its availability [...]. (Schwarcz, 2008, S.204)

## 2.2 Risk Governance

Die Definitionen, die im vorigen Abschnitt vorgestellt wurden, stammen aus dem Bereich der Ökonomie bzw. der Finanzwissenschaften. Im Jahr 2003 wurde der Begriff der Systemischen Risiken dann von der OECD aufgenommen und auf andere Themenfelder ausgeweitet. Der Bereich der Finanzmärkte wurde dabei explizit ausgeklammert. Stattdessen wurden fünf Kategorien von Risiken genannt, denen gemeinsam ist, dass sie nicht einzelne Individuen, sondern die Gesellschaft als Ganzes betreffen: Naturkatastrophen, Industrieunfälle, infektiöse Krankheiten, Terrorismus und Lebensmittelsicherheit. Aus dieser Beschreibung wird klar, dass Systemische Risiken hier vor allem mit Blick auf deren Reichweite definiert werden, wie das auch schon in einigen Definitionen aus Kapitel 2.1 der Fall war. Im Unterschied zu letzteren ist aber das System im Fokus nicht mehr das Finanzsystem. Der OECD geht es vielmehr um Systeme, die für das Funktionieren der Gesellschaft notwendig sind. Als Beispiele für solche Systeme werden das Gesundheitssystem, Transportsysteme, die Umwelt oder Telekommunikationssysteme benannt. In ihrer Definition von Systemischen Risiken stellt die OECD sie den konventionellen Risiken gegenüber:

Risk refers to the combination of two factors: the probability that a potentially harmful event will occur; and the potential damage such an occurrence would cause. [...] A systemic risk, in the terminology of this report, is one that affects the systems on which society depends - health, transport, environment, telecommunications, etc. (OECD, 2003, S.30)

Die OECD betont außerdem die Notwendigkeit einer Veränderung bei der Modellierung und (statistischen) Analyse von Risiken und insbesondere von Systemischen Risiken. In der Regel, so die OECD, würden einerseits hauptsächlich lineare Modelle verwendet, die andererseits Phänomene isoliert betrachten. Die neu aufkommenden Systemischen Risiken erfordern aber eine systemische Perspektive, in der auch nicht-lineare Zusammenhänge berücksichtigt werden sollen.

Die geforderte systemische Perspektive findet sich auch bei Renn und Klinke (2004) wieder. Sie betonen, dass Risiken für Gesundheit und Umwelt in einen größeren Zusammenhang eingebettet sind, der soziale, finanzielle und ökonomische Risiken und Chancen beinhaltet. Diese Sichtweise wird von Renn (2005) für das *International Risk Government Council (IRGC)* weiter ausgearbeitet und im Sinne einer Definition formuliert:

The IRGC has as one of its primary responsibilities the provision of expertise and practical advice in dealing with [...] ‘systemic risks’ (OECD, 2003). This term denotes the embeddedness of any risk to human health and the environment in a larger context of social, financial and economic consequences and increased interdependencies both across risks and between their various backgrounds. Systemic risks are at the crossroads between natural events [...], economic, social and technological developments and policy-driven actions, both at the domestic and the international level. (Renn, 2005, S.19)

Auch Renn (2005) betont außerdem die Notwendigkeit von neuen, ganzheitlichen Methoden (nicht nur) bei der Modellierung und Analyse von Systemischen Risiken. Explizit mit Blick auf die Vernetzung der Systemelemente untereinander formuliert das *Global Risk Network* seine Definition von Systemischen Risiken:

Systemic risks are risks that originate in an identifiable event that threatens predictable harm to one element of the system and which, due to links between the different systemic components is (or has the potential to be) amplified in either magnitude or direction, leading to substantial damage to the system as a whole. (Global Risk Network, 2006, zitiert von Neitzke, 2007)

Diese Definition hebt sich insofern von den übrigen ab, als dass hier Wechselwirkungen zwischen den Elementen des Systems offen angesprochen werden. Sie hat aber auch Ähnlichkeiten zu einigen der bisher genannten Definitionen aus Kapitel 2.1. Denn auch das Global Risk Network geht von einem auslösenden Element aus, das dem Systemischen Risiko vorausgeht.

## 2.3 Soziologie

Zusätzlich zu den bisher vorgestellten Ansätzen gibt es auch innerhalb der Soziologie einige Autoren, die sich mit Systemischen Risiken auseinandersetzen. Die Arbeiten von Helbing (2012, 2013) fokussieren auf methodische Aspekte, wobei seine Konzeption von Systemischen Risiken sich mit der Zeit etwas verändert. So beschreibt er Systemische Risiken zuerst als

[...] risks that can trigger unexpected large-scale changes or imply uncontrollable large-scale threats to it, (Helbing, 2012, S.1)

und bringt sie in einen Zusammenhang mit seiner Vorstellung von sozialen und sozio-ökonomischen Systemen als *komplexe Systeme*. In einer späteren Arbeit kann man dann folgende Definition lesen:

[...] systemic risk is the risk of having not just statistically independent failures, but interdependent, so-called 'cascading failures in a network of  $N$  interconnected system components. That is, systemic risks result from connections between risks ('networked risks'). (Helbing, 2013, S.51)

Hier fallen einige Punkte auf. Zunächst scheint es in der früheren Definition auch ein auslösendes Ereignis zu geben, welches aber mit dem Risiko gleich gesetzt wird. Das Risiko selbst *ist* also das auslösende Ereignis für die Veränderung oder Bedrohung des Systems. In der späteren Definition findet sich dann wieder der Aspekt der Ansteckung, hier explizit mit Bezug auf ein Netzwerk. In dieser Definition wird von dem Risiko des Versagens mehrerer Systemkomponenten gesprochen. Das Risiko ist hier also nicht mehr selbst das Ereignis. Das Systemische Risiko besteht nach dieser Lesart vielmehr darin, dass sich mehrere einzelne Risiken zu einem Netzwerk aus Risiken verbinden. Dieser Aspekt tritt in keiner der bisher vorgestellten Definitionen so deutlich hervor und steht sogar in einem Widerspruch zur Definition der OECD (2003). Das Systemische am Systemischen Risiko wird nach dieser Lesart nicht auf das System bezogen, in dem es auftritt, sondern auf ein System von Risiken, die sich wiederum auf ein komplexes System beziehen.

Im Gegensatz dazu ist Renn (2016, siehe auch Renn, Lucas, Haas und Jaeger, 2017) konsistent in seiner Definition von Systemischen Risiken. Er charakterisiert sie durch das Vorhandensein von vier Merkmalen:

Systemic risks can be characterized by four major properties [...]: they are (1) global in nature, (2) highly interconnected and intertwined leading to complex causal structures, (3) non-linear in the cause-effect relationships showing often unknown tipping points and (4) stochastic in their effect structure. (Renn et al., 2017)

Man kann Gemeinsamkeiten zwischen den Definitionen von Helbing und Renn ausmachen. So sprechen auch Renn et al. (2017) an anderer Stelle von komplexen Systemen und auch hier sind es Risiken, die miteinander in Wechselwirkung stehen. Die Betonung von nicht-linearen Zusammenhängen steht sowohl in Einklang mit der Definition der OECD (2003) als auch mit der Sichtweise von Hendricks et al. (2007). Und schließlich betonen Renn et al. (2017) die stochastische Natur Systemischer Risiken, ein Punkt, der zwar auch von Ferguson et al. (2001) und Kaufman und Scott (2003) erwähnt wird, aber dort nicht die gleiche Bedeutung erfährt.

## 2.4 Weitere Definitionen

Neitzke (2007) hat ebenfalls eine Definition von Systemischen Risiken vorgeschlagen. Auf Basis seiner Literaturrecherche macht er drei unterschiedliche Klassen von Definitionen bzw. den darin beschriebenen Risiken aus. Die erste Klasse von Risiken sind solche, bei denen das potentielle Ausmaß des Schadens besonders groß ist, evtl. sogar menschliche Opfer fordert, in jedem Fall aber mit hohen Kosten verbunden ist. Als Beispiel für diese Klasse von Risiken nennt Neitzke Erdbeben oder Hochwasser in dicht besiedelten Gebieten. Die zweite Klasse von Risiken bezieht sich auf kritische Infrastrukturen, also Infrastrukturen, von denen die Gesellschaft abhängig ist. Beispiele hier sind Stromversorgung oder Transportsysteme. Die dritte Klasse schließlich bezieht sich auf solche Risiken, die "erst aufgrund der Komplexität sozial-ökologischer oder technischer Systeme entstehen bzw. die sich aufgrund der Wechselwirkungen in komplexen Systemen 'aufschaukeln'" (Neitzke, 2007, S.1).

Auf diese Klasse von Risiken fokussiert Neitzke (2007) in seiner Definition von Systemischen Risiken. Während die ersten beiden Klassen schlicht und einfach "große" Risiken beschreiben, weil sie ein ganzes System betreffen, entstehen die Risiken der dritten Klasse erst durch die Wechselwirkungen der Systemelemente. Das System

ist hier gleichermaßen Verursacher und Betroffener. Seine Definition lautet folgendermaßen:

Ein Systemisches Risiko liegt dann vor, wenn sich eine Entwicklung in einem System aufgrund der dynamischen Wechselwirkungen zwischen den Elementen des Systems auf das System als Ganzes negativ auswirken kann. (Neitzke, 2007, S.12)

Er unterscheidet außerdem Systemische Risiken im engeren und im weiteren Sinne. Erste sind solche, die seiner Definition genügen, letztere sind solche, die den ersten zwei Klassen und damit auch der Konzeption der OECD (2003) folgen. Der Vollständigkeit halber sei an dieser Stelle auch die Definition genannt, die auf Wikipedia zu sehen ist, auch wenn der entsprechende Artikel den Titel *Systemrisiken* trägt. Die Definition lautet:

Das Systemrisiko (Systemisches Risiko) ist ein Risiko, das die Funktion oder das Fortbestehen eines ganzen Systems beeinträchtigen kann. Das Systemrisiko ist zu unterscheiden von spezifischen Risiken, von denen immer nur bestimmte Systemteilnehmer betroffen sind, ohne das System als Ganzes zu gefährden. (“Systemrisiko”, 2018)

Unter den Einzelnachweisen findet sich auch ein Link zur Quelle dieser Definition. Dieser Link führt auf eine Seite des Weltwirtschaftsforums, die allerdings zum Zeitpunkt des Abrufs “currently unavailable” war. Auch wenn gewisse Ähnlichkeiten zur Definition des Global Risk Network erkennbar sind (das vom Weltwirtschaftsforum ins Leben gerufen wurde), so scheint es sich um eine frühere und weniger ausgearbeitete Definition zu handeln. Ähnlich wie in der frühen Definition von Helbing (2012) sind hier die Risiken selbst aktiv in dem Sinne, als dass sie ein System “beeinträchtigen”. Abgesehen davon zielt diese Definition, wie schon einige andere, vor allem auf die Größe des Risikos ab. Im Sinne von Neitzkes Unterscheidung handelt es sich bei dieser Definition eindeutig um Systemische Risiken im weiteren Sinne.



# Kapitel 3

## Wiederkehrende Aspekte bei der Betrachtung Systemischer Risiken

Die Ausführungen des vorangehenden Kapitels haben gezeigt, dass es keinen Konsens bezüglich der Definition Systemischer Risiken gibt, weder im intra- noch im interdisziplinären Vergleich. Der Vorschlag von Neitzke (2007) stellt den Versuch einer fächerübergreifenden Definition dar. Bei genauem Hinsehen zeigt sich aber, dass seine Definition keine Abgrenzung zwischen Systemischen und konventionelle Risiken leisten kann. Denn auch wenn er die dynamischen Wechselwirkungen innerhalb des Systems betont, geht es immer noch um ein als negativ bewertetes Ereignis, das aufgrund einer Entwicklung entstehen kann. Zum Vergleich hier die Definition von konventionellen Risiken der *Society for Risk Analysis (SRA)*:

We consider a future activity [interpreted in a wide sense to also cover, for example, natural phenomena], for example the operation of a system, and define risk in relation to the consequences (effects, implications) of this activity with respect to something that humans value. (SRA, 2015, S.3)

Der grundlegende Unterschied zwischen dieser Definition und der von Neitzke liegt darin, dass Neitzke von einer “Entwicklung in einem System aufgrund dynamischer Wechselwirkungen” spricht, während die SRA “zukünftige Aktivitäten im weiteren Sinne” als Ursache für einen eventuellen Schaden verantwortlich machen.

Es lässt sich darüber streiten, inwiefern Neitzkes Definition über die der SRA hinausgeht, eine klare Abgrenzung zwischen konventionellen und Systemischen Risiken ist aber angesichts des direkten Vergleichs mit Neitzkes Vorschlag nicht zu erreichen.

Aus diesem Grund wird nun ein weiterer Versuch unternommen, eine theoretische Fundierung für den Umgang mit Systemischen Risiken zu schaffen. Dazu werden in den folgenden Unterkapiteln verschiedene Aspekte hervorgehoben, die in den unterschiedlichen Definitionen immer wieder formuliert werden und dem Autor deshalb als besonders wichtig erscheinen. Diese Aspekte sind Ausgangspunkte für die Entwicklung eines übergeordneten, integrativen Rahmenmodells, mit dem die im letzten Kapitel vorgestellten unterschiedlichen Ansätze vereint werden sollen.

### 3.1 Globalität

Ein Systemisches Risiko, hier sind sich wohl alle Ansätze einig, betrifft ein ganzes System oder zumindest weite Teile davon. Autoren aus dem Bereich der Ökonomie haben hier vor allem das Finanzsystem im Blick, ein System, das die ganze Welt umspannt und folglich auch für die gesamte Welt relevant ist. Im Bereich der Risikoforschung und der Sozialwissenschaften werden meist andere Beispiele erörtert, deren Reichweite aber keineswegs geringer ist. Klimawandel, Epidemien, soziale Ungleichheit, soziale Unruhen, Lebensmittel- und Wasserknappheit und *cyber security* sind typische Beispiele, die diskutiert werden (OECD, 2003; Renn et al., 2017; Helbing, 2013; Renn, Jovanovic & Schröter, 2011). Systemische Risiken betreffen demnach gesamtgesellschaftlich relevante Systeme (OECD, 2003; Helbing, 2013; Lucas, Jaeger & Renn, 2017).

Aus der Reichweite folgt die Größenordnung des Schadensausmaßes. Systemische Risiken betreffen ganze Gesellschaften und evtl., wie beim Beispiel des Klimawandels, sogar die gesamte Menschheit. Der Schaden besteht im Zusammenbruch des Systems (Kaufman & Scott, 2003). Neitzke (2007) weist darauf hin, dass manche Autoren Systemische Risiken deshalb als “besonders große Risiken” verstehen und darin gar das wesentliche Definitionsmerkmal sehen. Auch ohne so weit zu gehen, kann man Systemische Risiken aber als global bezeichnen. Der Begriff der Globalität beinhaltet dabei eine Mehrdeutigkeit, die wohlwollend in Kauf genommen wird. Zum einen betrifft ein Systemisches Risiko das ganze System, das betrachtet wird. Zum

anderen sind die betrachteten Systeme häufig weltumspannend, so dass tatsächlich der gesamte Globus betroffen ist.

## 3.2 Dynamische Wechselwirkungen

Der Aspekt der dynamischen Wechselwirkungen findet sich gleich zweifach in der Literatur wieder: zum einen in Bezug auf die Interaktion zwischen Systemelementen, zum anderen in Bezug auf die Interaktion zwischen Risiken.

Ersteres ist oft dann der Fall, wenn Systemische Risiken im Zusammenhang mit *nicht-linearen dynamischen Systemen* diskutiert werden (Hendricks et al., 2007; Helbing, 2013; Lucas et al., 2017; Haldane & May, 2011). Das Interesse liegt dabei vor allem auf sogenannten Gleichgewichtszuständen, auch Fixpunkten genannt, in denen sich ein System befinden bzw. aus denen es “herauskatapultiert” werden kann. Das kann geschehen, wenn das System eine *Perturbation* erfährt. Ferner können Gleichgewichtszustände stabil oder instabil sein, was sich auf das Verhalten des Systems nach einer Perturbation auswirkt. Zu einem stabilen Fixpunkt kehrt das System nach der Perturbation zurück, zu einem instabilen Fixpunkt nicht. Ein Phänomen, das auch in die populärwissenschaftliche Literatur eingezogen ist und so einer breiteren Öffentlichkeit zugänglich wurde, ist der *tipping point* (Gladwell, 2006). Damit gemeint ist ein Zustand, der, wenn er einmal erreicht ist, einen Zustandswechsel des Systems unausweichlich nach sich zieht. Ein solcher *point of no return* könnte also einen Wechsel in ein instabiles Gleichgewicht markieren.

Die Anwendung der Theorie der dynamischen Systeme auf Systemische Risiken besteht letztlich in der Formulierung eines Differentialgleichungssystems, das die wechselseitige Beziehung der Systemelemente in Abhängigkeit von der Zeit beschreibt. Ein denkbares Risiko besteht dann darin, dass das System sich in einem unerwünschten Gleichgewichtszustand befindet bzw. sich in die Richtung eines solchen Zustands bewegt. Ein Systemisches Risiko könnte auch in der Veränderung eines oder mehrerer Parameter liegen, die zu einer qualitativen Veränderung führt. So könnten etwa mögliche Gleichgewichtszustände verschwinden. Da es sich bei einem solchen Vorgehen um mathematische Modellierung handelt, sind Arbeiten dieser Art methodisch sehr anspruchsvoll und durch einen vergleichsweise hohen Grad an Formalisierung gekennzeichnet. Sie setzen außerdem ein Maß an Wissen über das

betrachtete System voraus, dass die Formulierung eines Systems von Differentialgleichungen erlaubt.

Nicht nur im Gegensatz dazu, sondern auch in Kombination damit werden Wechselwirkungen zwischen verschiedenen Risiken diskutiert. Helbing (2013) spricht in diesem Zusammenhang auch von vernetzten Risiken. Auffallend ist, dass dieser Aspekt vor allem in der sozialwissenschaftlich orientierten Literatur präsent ist (Renn, 2005; Lucas et al., 2017; Renn et al., 2017; Helbing, 2013). Die Feststellung etwa, dass Risiken für Gesundheit und Umwelt soziale, ökonomische und finanzielle Konsequenzen haben können (Renn, 2005), stellt ohne Zweifel eine Verknüpfung von verschiedenen Risiken dar. Als ein konkretes Beispiel nennen Lucas et al. (2017) den Klimawandel, der Dürre und Hungersnöte auf der einen Seite, aber auch die Zunahme von extremen Wetterereignissen und ansteigenden Meeresspiegel auf der anderen Seite zur Folge haben kann. Beides kann wiederum weitere Folgen wie Flucht und Migration oder soziale Unruhen nach sich ziehen.

### 3.3 Ansteckung und Ausbreitung

Im Zusammenhang mit Epidemien bzw. Pandemien sind es offensichtlich Mechanismen der Ansteckung und Ausbreitung, die bei der Modellierung im Vordergrund stehen. Aber auch in der ökonomisch orientierten Literatur spielen diese Aspekte eine Rolle. Häufig stellt man sich das Finanzsystem als ein Netzwerk aus Banken und anderen finanziellen Institutionen vor, die aufgrund von Geschäftsbeziehungen untereinander in vielfältiger Art und Weise miteinander verbunden sind (Bianchi, Billio, Casarin & Guidolin, 2015; Goldin & Vogel, 2010; May, Levin & Sugihara, 2008). Wenn nun eine Bank in Schwierigkeiten gerät, so kann das auch Auswirkungen auf ihre Geschäftspartner haben. Die “kranke” Bank kann ihre Geschäftspartner “anstecken” (Eine tiefere Auseinandersetzung mit dem Ansteckungsbegriff findet sich beispielsweise bei Forbes & Rigobon, 2001). Aufgrund des hohen Vernetzungsgrads kann sich ein ursprünglich lokal auftretendes Problem dann ausbreiten und weite Teile des Netzwerks erfassen. Die Folge ist eine globale Krise, die im schlimmsten Fall zum Zusammenbruch des Finanzmarkts führt.

War es im vorigen Abschnitt die Theorie der dynamischen Systeme, so ist es jetzt die Graphentheorie, die methodisch im Vordergrund steht. Dabei werden häu-

fig strukturelle Analysen des Netzwerks durchgeführt, um Schlüsse über Eigenschaften und Verhalten des Systems machen zu können. Eine wichtige Kennzahl ist der *Grad* eines Knotens, also die Anzahl seiner Verbindungen. Anhand der Verteilung der Knotengrade und der damit zusammenhängenden topologischen Eigenschaften werden Netzwerke in unterschiedliche Kategorien eingeteilt, wobei vor allem *random networks*, *small world networks* und *scale free networks* untersucht werden.

Random networks (deutsch: *Zufallsnetze*) (Erdos & Rényi, 1960) bestehen aus einer festgelegten Anzahl an Knoten und einer zufälligen Anzahl von Kanten, die durch einen Parameter gesteuert werden. Die sich ergebende Verteilung der Knotengrade ist zufällig, der Vernetzungsgrad des entstehenden Graphen hängt aber stark von diesem Parameter ab: ist er klein, so hat der resultierende Graph nur wenig Kanten, ist er groß, so sind die Knoten untereinander stark vernetzt.

Small-World-Netzwerke (Watts & Strogatz, 1998) sind stärker strukturiert. Ihre Verteilung ist flach, alle Knoten haben einen ähnlichen Grad. Knoten sind mit ihren nächsten Nachbarn verbunden, aber es gibt immer wieder Verbindungen zu weiter entfernten Knoten. Durch diese Topologie entsteht die typische Eigenschaft von Small-World-Netzwerken: jeder Knoten kann von jedem anderen Knoten über einige wenige Zwischenschritte erreicht werden. Erste empirische Evidenz für die Existenz der Small-World Eigenschaft in sozialen Netzwerken wurde bereits von Travers und Milgram (1967) vorgelegt. Neuen Aufschwung erlebte diese Art der Netzwerkforschung dann durch die Arbeiten von Watts und Strogatz (1998). Watts (1999) konnte für verschiedene Netzwerke zeigen, dass sie Small-World Eigenschaften besitzen.

Die sogenannten *skalenfreien* Netzwerke, manchmal auch *Preferential Attachment Networks* genannt, wurden von Barabási und Albert (1999) in die Literatur eingebracht. Netzwerke dieser Klasse besitzen zwar ebenfalls Small-World Eigenschaften, unterscheiden sich aber hinsichtlich ihrer Topologie von den oben genannten Netzwerken. Die meisten Knoten haben hier einen relativ geringen Knotengrad, sind also nur wenig vernetzt. Einige wenige Knoten aber haben einen sehr hohen Knotengrad und sind demnach stark vernetzt. Barabási und Albert (1999) konnten für einige Netzwerke wie das World Wide Web und ein Netzwerk wissenschaftlicher Zitationen zeigen, dass ihre Topologie der der skalenfreien Netzwerke folgt.

Man kann nun untersuchen, inwiefern sich Netzwerke unterschiedlicher Topologi-

en hinsichtlich Ansteckung und Ausbreitung verhalten. Dezsó und Barabási (2002) fanden, dass in skalenfreien Netzwerken vor allem die stark vernetzten Knoten eine große Rolle spielen, wenn es um die Ausbreitung und Weiterverbreitung innerhalb eines Netzwerks geht. Colizza, Barrat, Barthélemy und Vespignani (2006) untersuchten die Ausbreitung von ansteckenden Krankheiten unter Berücksichtigung des weltweiten Flugverkehrsnetzwerks. Häufig steht bei dieser Art der Modellierung ein Ausfall bestimmter Knoten im Vordergrund. Buzna, Peters und Helbing (2006) legen ein allgemeines Modell zur Analyse der Ausbreitung des Versagens von Knoten in Netzwerken vor.

### 3.4 Dynamische Risiken

Der letzte hier angesprochene Aspekt markiert vielleicht den größten Unterschied zu konventionellen Risiken: Systemische Risiken sind nicht konstant. Am deutlichsten wird das bei Ferguson et al. (2001), wenn sie davon sprechen, dass die Wahrscheinlichkeit eines systemischen Ereignisses, das ihrer Konzeption nach Auslöser eines Systemischen Risikos ist, sich im Laufe der Zeit aufbauen kann. Der Eintritt eines solchen Ereignisses kann dann plötzlich und unerwartet erscheinen, weil die immer weiter steigende Eintrittswahrscheinlichkeit nicht bemerkt wurde. Eine solche dynamische Veränderung ist gemeint, wenn in dieser Arbeit von dynamischen Risiken die Rede ist. Auch wenn sich dieser Punkt so direkt nur selten in der Literatur findet, so ist er bei genauerem Hinsehen implizit oft vorhanden. Wenn beispielsweise im Zusammenhang mit Systemischen Risiken von Tipping Points die Rede ist, dann ist damit die Veränderung in Richtung eines Zustands gemeint, an dem bestimmte Folgen nicht mehr verhindert werden können (Renn et al., 2017; Lucas et al., 2017; Georg, 2013; Helbing, 2012). Das impliziert aber, dass vor Erreichen dieses Punkts eine Abwendung dieser Folgen durchaus möglich ist. Wir haben es also mit einer dynamischen Veränderung des Zustands eines Systems zu tun, das sich etwa in Richtung eines instabilen Fixpunkts bewegt (Hendricks et al., 2007).

Aber auch in methodisch anderweitig orientierter Forschung findet dieser Aspekt wieder. So verstehen etwa Bianchi et al. (2015), deren Arbeit auf Netzwerke und Graphentheorie fokussiert, Systemische Risiken als dynamische Veränderung der Konnektivität innerhalb eines Netzwerks. Ein ähnlicher Ansatz findet sich bei Diebold

und Yilmaz (2014). Colizza et al. (2006) kombinieren das in der Epidemiologie häufig verwendete SIR-Modell (Kermack & McKendrick, 1927) mit einem netzwerk-basierten Ansatz, um die Entwicklung einer globalen Epidemie zu modellieren.

Die Betonung der Einbettung von Risiken für Gesundheit und Umwelt in einen größeren Zusammenhang, der auch soziale, wirtschaftliche und finanzielle Konsequenzen beinhaltet (Renn, 2005), kann ebenfalls unter diesem Gesichtspunkt betrachtet werden. Wenn etwa Lucas et al. (2017) die sozialen Folgen von Extremwetterereignissen betonen, die durch den Klimawandel begünstigt werden, der wiederum unter anderem durch Industrieproduktion und die damit verbundenen CO<sub>2</sub>-Emissionen begünstigt wird, dann wird schnell klar, dass statische Modelle hier nur begrenzt sinnvoll sind. Wollte man in dieser Situation dem Problem des CO<sub>2</sub>-Ausstoßes dadurch begegnen, dass man die Industrieproduktion zurückfährt, dann müssen beispielsweise auch soziale und ökonomische Auswirkungen für die davon betroffenen Arbeiter berücksichtigt werden.

Tatsächlich drängt sich der Eindruck auf, dass die Dynamik ein inhärentes Merkmal von Systemischen Risiken ist, denn bei genauer Betrachtung ist es in allen hier angesprochenen Ansätzen zu finden. So deutet schon die Anwendung der Theorie der dynamischen Systeme auf die Betrachtung eines dynamischen Phänomens hin. Auch Ansteckung und Ausbreitung sind kaum ohne die Berücksichtigung einer wie auch immer gearteten Dynamik denkbar, was am Beispiel der Epidemie besonders deutlich hervortritt. Es erscheint deshalb verwunderlich, dass dieser Aspekt nicht stärker betont wird, um eine Abgrenzung zwischen Systemischen und Nicht-Systemischen Risiken zu erreichen.



# Kapitel 4

## Entwicklung eines Rahmenmodells für Systemische Risiken

### 4.1 Differenzierung der Merkmale nach Betrachtungs- ebenen

#### 4.1.1 Ebenen der Betrachtung

Die Identifizierung dieser häufig genannten Merkmale ist ein erster Schritt auf dem Weg zu einem Rahmenmodell. Eine Integration der verschiedenen Ansätze ist mit ihnen jedoch noch nicht möglich. Eine Hilfestellung dafür bietet ein Artikel von Resch (2017). Er beschreibt die Schritte auf dem Weg zu einer Computersimulation, die zunächst auf Basis eines mathematischen Modells entsteht. Das mathematische Modell wiederum entsteht auf Basis eines physikalischen Modells, welches aus der Realität abgeleitet wird. Oder anders herum: von der Realität gelangt man über ein physikalisches zu einem mathematischen Modell und von da zu einem Simulationsmodell. (Das Erstellen eines Simulationsmodells besteht selbst wieder aus einer Reihe von verschiedenen Schritten, die Resch (2017) detailliert benennt.) Resch (2017) beschäftigt sich zwar nicht mit Systemischen Risiken, sondern geht einer ganz anderen Frage nach. Trotzdem sind seine Überlegungen auch hier wertvoll. Betrachtet man die Merkmale vor diesem Hintergrund, so fällt auf, dass sie verschiedenen Ebenen zuzuordnen sind, die denen von Resch (2017) entsprechen bzw. an sie angelehnt sind.

### **Ebene realer Phänomene**

Wenn im Zusammenhang mit Systemischen Risiken über den Klimawandel oder über Finanzkrisen gesprochen wird, dann sind das Phänomene, von deren Folgen “wir”, also die allgemeine Bevölkerung, mehr oder weniger direkt betroffen sind. Sie begegnen uns im Alltag auf verschiedene Art. Forschung innerhalb der Sozialwissenschaften interessiert sich letztlich für Phänomene auf dieser Ebene, die als *Ebene realer Phänomene* bezeichnet werden soll.

### **Ebene theoretischer Abstraktion**

Um ein reales Phänomen zu verstehen, zu erklären oder durch Intervention zu verändern, ist die Kenntnis zumindest einiger relevanten Größen sowie der kausalen Beziehungen zwischen ihnen nötig. Sie zu finden ist letztlich die Aufgabe der Wissenschaft und die Entwicklung von Theorien und Modellen ist die Art und Weise, wie das geschieht. Dabei gehört es zum Grundverständnis, dass eine Theorie oder ein Modell niemals eine exakte Kopie der Realität sein kann, sondern immer nur einen Ausschnitt daraus darstellt. Es muss hier also von der Komplexität der realen Phänomene abstrahiert werden (vgl. dazu auch Kapitel 4.2.1). In den Sozialwissenschaften ist es nicht unüblich, dass solche Theorien nicht formalisiert, sondern qualitativer Natur sind. Aufgrund der großen Komplexität sozialer Phänomene ist die theoretische und abstrakte Auseinandersetzung mit dem Forschungsgegenstand auf qualitativer Ebene möglich und sinnvoll. Ist eine Theorie einmal auf der *Ebene theoretischer Abstraktion* formuliert, kann auch eine formale Betrachtung erfolgen.

### **Ebene formaler Repräsentationen**

Die Entwicklung von formalisierten Modellen geschieht typischerweise durch die Anwendung mathematischer Ideen. Ein soziales Netzwerk kann etwa als Graph beschrieben werden, dessen Knoten die Personen und dessen Kanten die Beziehungen zwischen ihnen sind. In der Analyse dieses Graphen können dann Begriffe zum Einsatz kommen, die innerhalb der Graphentheorie wohl definiert sind. Die *mittlere Weglänge*, zum Beispiel, beschreibt die durchschnittliche Anzahl an Knoten auf dem Weg zwischen zwei beliebigen Knoten. Ist sie vergleichsweise kurz, dann kann das im Bezug auf das soziale Netzwerk so *interpretiert* werden, dass zwei beliebige Personen

des sozialen Netzwerks durch einige wenige Zwischenstationen miteinander in Kontakt treten können (Ein Ergebnis, dass durch Watts, 2004a, unter der Bezeichnung *six degrees of separataion* in die Literatur eingegangen ist).

Wichtig ist nun die Feststellung, dass die mittlere Weglänge ein Begriff aus der Graphentheorie ist und seine Geltung auf die mathematische Graphenstruktur beschränkt ist. Ein soziales Netzwerk ist aber kein Graph, auch wenn die formale Repräsentation durch einen Graphen intuitiv verständlich erscheint. Stattdessen kann ein Graph zur Repräsentation eines sozialen Netzwerks verwendet werden. In diesem Zug können dann auch Begriffe wie die mittlere Weglänge zur Analyse des Graphen herangezogen werden. Was die mittlere Weglänge dann aber im Bezug auf das soziale Netzwerk bedeutet, ist eine Frage der Interpretation. Im sozialen Netzwerk selber, das ja aus Personen und deren Beziehungen untereinander besteht, existiert so etwas wie eine Weglänge nicht. Die Ebene, auf der reale Phänomene formal repräsentiert werden, soll im Folgenden als *Ebene formaler Repräsentation* bezeichnet werden.

### **Ebene der Implementierung**

Spätestens bei der Entwicklung einer Computersimulation muss eine weitere Unterscheidung gemacht werden. Auch wenn man ein Simulationsprogramm selbst als formale Spezifikation verstehen kann, es wird zunächst auf Basis eines formalen Modells entwickelt. Dabei können unterschiedliche Simulationsansätze gewählt werden, auf die weiter unten noch eingegangen wird. Wichtig ist an dieser Stelle, dass es sich um eine weitere Ebene handelt, was sich auch dadurch zeigt, dass ein formales Modell mit verschiedenen Ansätzen in eine Computersimulation überführt werden kann. Diese Ebene soll im Folgenden als *Ebene der Implementierung* bezeichnet werden.

### **4.1.2 Differenzierung der Merkmale**

Ein großer Teil der Probleme bei der Entwicklung eines übergeordneten Rahmenmodells löst sich nun auf, wenn man die Merkmale den unterschiedlichen Ebenen zuordnet. Auf der Ebene realer Phänomene unterscheiden sich die verschiedenen Disziplinen naturgemäß, was den Gegenstand ihrer Forschung betrifft. In der Ökonomie bzw. in den Finanzwissenschaften liegt das Interesse auf der Analyse von Finanzkrisen, in der Soziologie dagegen werden soziale Aspekte wie soziale Ungleichheit in

den Vordergrund gerückt.

Die Vernetzung verschiedener Risiken und die dadurch neu entstehenden Risiken dagegen gehören zur die Ebene theoretischer Abstraktion, denn hier werden bereits Wirkmechanismen zwischen verschiedenen Phänomenen thematisiert, die theoretisch begründet werden können und müssen. Wie schon weiter oben angesprochen wurde, ist die Komplexität hier bereits so groß, dass Forschung ausschließlich auf dieser Ebene betrieben werden kann und muss.

Bei der Entwicklung formaler Repräsentationen der realen Phänomene können natürlich ganz unterschiedliche Wege gegangen werden. Es überrascht deshalb nicht, dass es auch auf der Ebene formaler Repräsentationen unterschiedliche Herangehensweisen gibt. Zunächst muss herausgearbeitet werden, welches System betrachtet wird. Dabei müssen zum einen die Elemente des Systems benannt, zum anderen die Beziehungen zwischen ihnen spezifiziert werden. Wie bereits weiter oben betont wurde, dominieren hier besonders zwei Herangehensweisen: Wenn auf Ansteckung und Ausbreitung fokussiert wird, dann werden eher Graphentheoretische Modelle benutzt, wenn auf dynamische Wechselwirkungen fokussiert wird, arbeitet man eher mit Differentialgleichungssystemen. Die Globalität ist ebenfalls auf dieser Ebene verortet. Hier kommt zum Ausdruck, dass das gesamte System betroffen ist.

Man kann diskutieren, ob die Berechnung der Lösung bzw. der Lösungen eines Gleichungssystems, die Bestimmung eines tipping points oder auch verschiedener statistischer Kennwerte eines Graphen noch zur Ebene formaler Repräsentationen oder schon zur Ebene der Implementierung gehört. Spätestens wenn aber ein Simulationsmodell entwickelt werden soll, dann geschieht das auf der Ebene der Implementierung. Wie schon weiter oben gesagt, gibt es auch hier gibt es eine Vielzahl unterschiedlicher Ansätze. Braun und Saam (2015) unterscheiden fünf Ansätze, zu denen auch die *Agentenbasierte Modellierung* gehört, die sich in den Sozialwissenschaften großer Beliebtheit erfreut (für eine Einführung siehe Macal & North, 2009).

## 4.2 Integration der Merkmale durch Abstraktion

Durch die Unterscheidung der Ebenen wird eine Differenzierung der Merkmale erreicht. Für ein übergeordnetes Rahmenmodell müssen die Merkmale und Ansätze aber gleichsam integriert werden. Die Herausforderung dabei liegt vor allem in der

Integration der Ansätze, die auf Differenzialgleichungssystemen basieren und der Graphentheoretischen Ansätze. Mit Blick auf die Betrachtungsebenen handelt es sich bei diesen zwei Ansätzen um konkrete formale Repräsentationen, deren Integration auf einer höheren Abstraktionsebene, der Ebene theoretischer Abstraktion, erreicht werden kann.

### 4.2.1 Abstraktion

Eine prägnante Beschreibung dessen, was im Kontext der vorliegenden Arbeit unter Abstraktion zu verstehen ist, findet sich bei Cartwright (1994): “strip away all that is irrelevant to the concerns of the moment to focus on some single property or a set of properties”. Eine ähnliche Sichtweise hat Jones (2005), der Abstraktion als “omission of a truth”, als das Weglassen von etwas bezüglich eines Modells bezeichnet. Jones geht auch auf die Entwicklung von Metamodellen ein, die er “models of kinds” nennt. Ein solches Metamodell ist eine Art Zusammenfassung verschiedener, konkreter Modelle des gleichen Systems oder Prozesses zu einem übergeordneten Modell, das einige Eigenschaften der betrachteten Modelle beinhaltet und einige eben nicht. Die berücksichtigten Eigenschaften sind nach Weisberg (2007) kausale Faktoren, die den Kern des Modells ausmachen. Er spricht auch von denjenigen Faktoren, die “den Unterschied machen”.

### 4.2.2 Rekursion als abstraktes Merkmal

Betrachten wir also die beiden konkurrierenden Ansätze vor diesem Hintergrund etwas genauer. Dynamische Wechselwirkungen im Sinne der Theorie der dynamischen Systeme werden formal und mathematisch als System von Differenzialgleichungen formuliert (Strogatz, 2018). Eine Differenzialgleichung ist eine Funktion, die die Veränderung einer Variablen in Abhängigkeit ihres Wertes - und evtl. weiterer Variablen wie der Zeit - beschreibt. Man kann also sagen, dass hier auf mathematische Weise beschrieben wird, inwiefern eine Variable und deren Entwicklung von sich selbst abhängt.

Bei der Modellierung von Ansteckungs- und Ausbreitungseffekten auf Graphen werden zunächst einige wenige Knoten infiziert, die dann ihre Nachbarn anstecken. Die neu infizierten Nachbarn stecken dann wiederum ihre Nachbarn an und es kommt

zu einer Ausbreitung der Infektion. Wie schnell sich die Infektion ausbreitet, hängt von verschiedenen Faktoren ab, die an dieser Stelle nicht weiter betrachtet werden sollen. Wichtig ist die Feststellung, dass die Entwicklung der Anzahl der betroffenen Knoten von der Anzahl der betroffenen Knoten abhängt.

In beiden Ansätzen tritt also der Selbstbezug von etwas hervor. Die Ausprägung eines Phänomens hängt von der Ausprägung eben dieses Phänomens ab. Es ist dieses abstrakte Merkmal, das die Integration der Ansätze möglich macht. Verschiedene Autoren benutzen verschiedene Begriffe, die jeweils auf den Kontext zugeschnitten sind. Im Zusammenhang mit Infektionen wird von Ansteckung und Ausbreitung gesprochen, im Zusammenhang von kritischen Infrastrukturen eher von Kaskaden. Auch Feedback bzw. Wechselwirkungen sind denkbare Begriffe. Um hier einen allgemeineren Begriff zu haben, werde ich im weiteren Verlauf der Arbeit von *Rekursivität* sprechen.

## 4.3 Wesentliche Merkmale Systemischer Risiken

Auf Basis dieser Vorüberlegungen ist es nun möglich, wesentliche Merkmale Systemischer Risiken zu benennen, mit denen die Formulierung eines übergeordneten, integrativen Rahmenmodells möglich wird. Es treten drei Merkmale zum Vorschein: *Globalität*, *Rekursivität* und *Dynamik*.

### 4.3.1 Globalität

Die Zweideutigkeit des Globalitätsaspekts wurde in 3.1 bereits herausgestellt. Dass ein Systemisches Risiko global ist, kann einerseits heißen, dass der gesamte Globus betroffen ist. Es kann andererseits bedeuten, dass ein System als Ganzes betroffen ist. Mit Blick auf das Rahmenmodell, und um eine saubere Trennung der unterschiedlichen Betrachtungsebenen zu erreichen, soll an dieser Stelle vor allem Letzteres betont werden. Durch das Merkmal Globalität soll also zum Ausdruck kommen, dass ein Zusammenbruch des System als Folge einer Entwicklung droht. Damit ist die Globalität, so wie die folgenden Merkmale auch, auf der Ebene theoretischer Abstraktion verortet. Es sei aber nochmals betont, dass auf der Ebene realer Phänomene typischerweise solche Phänomene betrachtet werden, die tatsächlich weltumspannend

sind.

### 4.3.2 Rekursivität

Im Zusammenhang mit dem betrachteten System gibt es immer mindestens ein Element mit Selbstbezug, wie in Kapitel 4.2.2 herausgearbeitet wurde. In Bezug auf die Betrachtungsebenen ist dieses Merkmal auf der Ebene formaler Repräsentationen verortet. Auf welche Art diese rekursive Beziehung zum Ausdruck kommt, hängt dementsprechend von den Details der Formalisierung ab. Die Rekursivität kann sich auf ein einzelnes Systemelement beziehen oder aber auf ein aggregiertes Merkmal. Letzteres ist bei der Fokussierung auf Ansteckung und Ausbreitung der Fall, ersteres bei der Modellierung eines nicht-linearen, dynamischen Systems.

### 4.3.3 Dynamik

Das dritte Merkmal von Systemischen Risiken folgt eigentlich aus der Rekursivität: Systemische Risiken sind spezielle Vertreter der Gruppe der dynamischen Risiken. Wenn die Entwicklung eines Merkmals von diesem Merkmal selbst abhängt, dann hat man es mit einem dynamischen Phänomen zu tun. Auch die Dynamik gehört zur Ebene formaler Repräsentationen, was schon allein durch die Beziehung zur Rekursivität bedingt ist. Die Redundanz, die durch die Benennung der Dynamik entsteht, kann durchaus kritisiert werden. Vielleicht werden weitere Überlegungen zeigen, dass man sie nicht als wesentliches Merkmal explizit benennen sollte. Die Betonung der Dynamik erlaubt aber eine Abgrenzung von konventionellen Risiken, was eine explizite Hervorhebung rechtfertigt.

## 4.4 Formulierung des Rahmenmodells

Die drei wesentlichen Merkmale können nun in einem Rahmenmodell zueinander in Beziehung gebracht werden. Die Formulierung erfolgt in Anlehnung an die allgemeine Definition konventioneller Risiken der SRA (2015) (siehe auch Aven, 2016). Ähnlich wie im Fall der konventionellen Risiken soll so eine allgemeine Formulierung erreicht werden, die unterschiedliche Konzeptionen integriert und mit Ansätzen auch

aus verschiedenen Disziplinen vereinbar ist. Bei dem nun folgenden Vorschlag handelt es sich um ein qualitatives Rahmenmodell, dass in Bezug auf die verschiedenen Betrachtungsebenen aus Kapitel 4.1.1 auf der Ebene theoretischer Abstraktion angesiedelt ist.

**Rahmenmodell.** *Sei  $\mathcal{S}$  ein System, bestehend aus einer Menge von Elementen, die sich gegenseitig beeinflussen können. Die Elemente von  $\mathcal{S}$  können beliebig sein, insbesondere kann es sich bei ihnen wieder um Systeme handeln.  $\mathcal{S}$  ist nicht statisch, sondern unterliegt im Laufe der Zeit Veränderungen, die sich als Konsequenz aus der Wechselwirkung der Elemente untereinander ergeben. Die Veränderungen können sowohl die Elemente als auch deren Beziehungen untereinander betreffen. Mindestens eins der Systemelemente steht dabei in einer Wechselwirkung mit sich selbst. Dieses rekursive Element, das mit  $\mathcal{R}$  bezeichnet sei, kann selbst wieder ein System aus mehreren Elementen sein, die sich gegenseitig rekursiv beeinflussen. Ein Systemisches Risiko besteht dann, wenn sich das ursprüngliche System  $\mathcal{S}$  potentiell in einem solchen Maß verändert, dass aus ihm ein neues System  $\mathcal{S}'$  wird, das nicht mehr als äquivalent zu  $\mathcal{S}$  angesehen wird.  $\mathcal{S}$  wird dabei als das wertvollere System erachtet, dessen Bestehen gesichert werden soll.*

Der Versuch einer bildlichen Darstellung des Rahmenmodells ist in 4.2 zu sehen. Abgebildet ist die Veränderung eines Systems über die Zeit hinweg, die (unter anderem) durch die selbstbezogene Wechselwirkung des rot eingezeichneten rekursiven Elements stattfindet. Durch die Grafik soll angedeutet werden, dass die Veränderung sowohl die Elemente als auch deren Beziehungen untereinander betreffen kann. So können etwa Elemente verschwinden, oder auch neue hinzukommen. Auch eine qualitative Veränderung ist möglich, die allerdings in der Grafik nicht zum Ausdruck kommt. Gleiches gilt für die Beziehungen der Elemente untereinander, die durch Pfeile angedeutet sind.

Dieses sparsame Rahmenmodell reicht aus, um die verschiedenen Ansätze zu integrieren und um einen Startpunkt für die Modellierung zu geben. Es ist offen in Bezug auf die realen Phänomene, die betrachtet werden sollen. Es ist ebenso offen in Bezug auf die formale Repräsentation und deren Implementierung. Man beachte, dass im Rahmenmodell nicht gesagt wird, *was* ein Systemisches Risiko ist. Auch darin besteht eine Gemeinsamkeit zur Definition der SRA (2015), in der konventionelle

Risiken nicht selber definiert, sondern in Beziehung zu Konsequenzen einer Aktivität gesetzt werden. Stattdessen werden durch das Rahmenmodell die wesentlichen Merkmale von Systemischen Risiken benannt und deren Zusammenwirken beschrieben. Dieses Vorgehen ist angelehnt an die Konzeption von Renn et al. (2017), die Systemische Risiken durch das Vorhandensein wichtiger Komponenten definieren.

## 4.5 Systemische Risiken und Resilienz

Im Zusammenhang mit Systemischen Risiken kann auch die Beschäftigung mit dem Konzept der *Resilienz* fruchtbar sein. Dieser Begriff unterlag im Laufe der Zeit einigen Veränderungen. Laut Bach, Bouchon, Fekete, Birkmann und Serre (2013) wird er aktuell von der *Resilience Alliance* definiert als

The ability to absorb disturbances, to be changed and then to re-organise and still have the same identity (retain the same basic structure and ways of functioning). (Bach et al., 2013, S.2)

Vergleicht man diese Definition mit dem im vorigen Kapitel vorgestellten Rahmenmodell zu Systemischen Risiken, dann kann man den Eindruck bekommen, dass hier aus unterschiedlichen Perspektiven über ähnliche Dinge geschrieben wird. Wenn Resilienz als die Fähigkeit eines Systems angesehen wird, Veränderungen zu erfahren und dabei trotzdem seine Identität zu behalten, dann könnte man Systemische Risiken und Resilienz ohne Probleme in einen Zusammenhang stellen. Ein System ist dann resilient, wenn es mit einem Systemischen Risiko umgehen kann. Die Veränderung des Systems, die unter anderem durch das rekursive Element zustande kommt, führt dann eben nicht dazu, dass das System sich in einer Weise verändert, dass man nicht mehr vom gleichen System sprechen kann.

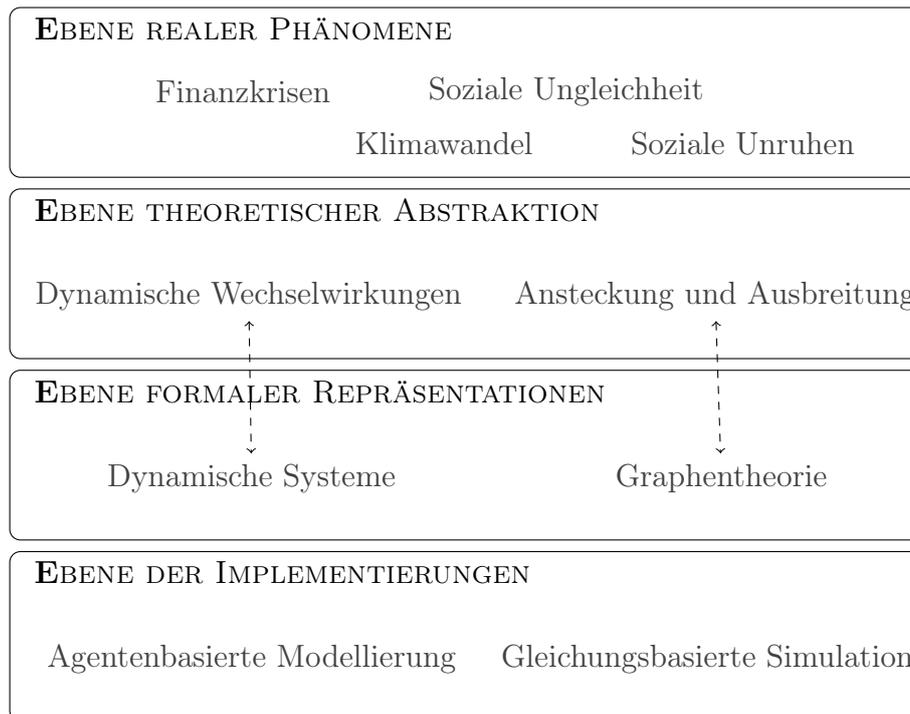


Abbildung 4.1: Hierarchie der Ebenen, auf denen Systemische Risiken betrachtet werden können. Die Reihenfolge spiegelt unterschiedliche Phasen des Forschungsprozesses wieder, an dessen Anfang ein reales Phänomen steht, das Gegenstand der Forschung ist. Auf der Ebene theoretischer Abstraktion findet zunächst eine eher qualitative Auseinandersetzung statt, die in der Formulierung einer Theorie münden kann. Im nächsten Schritt kann diese Theorie formalisiert und anschließend in Form einer Computersimulation implementiert werden.

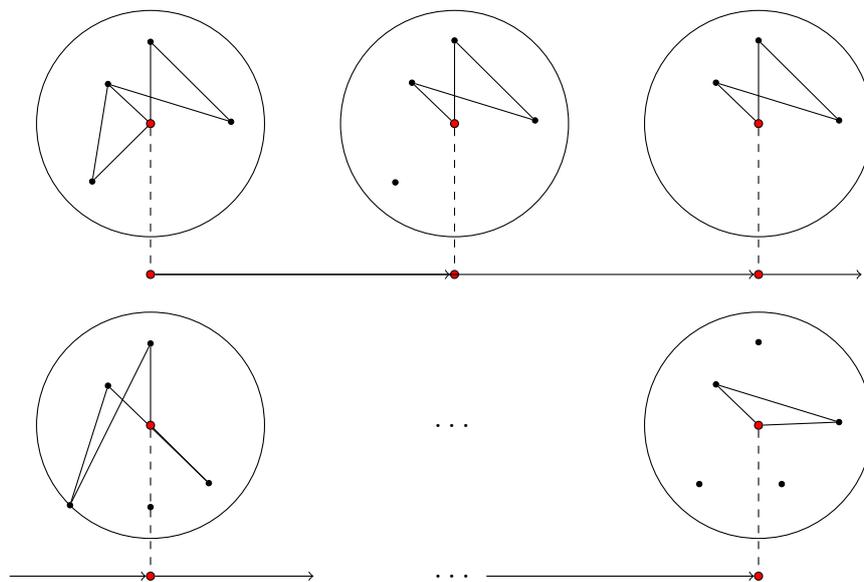


Abbildung 4.2: Grafische Darstellung der zeitlichen Veränderung eines Systems. Rot eingezeichnet ist das rekursive Element, das sich über die Zeit hinweg selbst beeinflusst. Unter anderem durch diese rekursive Beziehung verändert sich das System im Laufe der Zeit sehr stark.



## Teil II

# Modellierung Systemischer Risiken mittels Probabilistischer Programmierung



# Kapitel 5

## Die Methode der Probabilistischen Programmierung

Bei der Entwicklung des Rahmenmodells wurde von den Details der Modellierung und Implementierung abstrahiert. Dadurch sollte unter anderem ein Ausgangspunkt für die Modellierung von Systemischen Risiken geschaffen werden, ohne die zu verwendenden Methoden zu restringieren. Basierend auf dem Rahmenmodell können nun auf einer theoretisch abstrakten Ebene Beispiele für Systemische Risiken identifiziert und anschließend unter der Verwendung geeigneter Methoden formalisiert werden. In Bezug auf die Formalisierung liegt der Fokus der vorliegenden Arbeit dabei auf der Entwicklung probabilistischer Modelle. Die Notwendigkeit probabilistischer Modelle im Zusammenhang mit Systemischen Risiken wird besonders von Renn et al. (2017) hervorgehoben, die stochastische Ursache-Wirkungsstrukturen als definierendes Merkmal von Systemischen Risiken verstehen. Auch wenn das oben entwickelte Rahmenmodell diesen Punkt nicht beinhaltet, diese Einschätzung der Bedeutung probabilistischer Modelle in sozialwissenschaftlich orientierter Forschung wird auch vom Autor der vorliegenden Arbeit geteilt. Denn im Gegensatz zu manchen Bereichen der naturwissenschaftlichen Forschung liegt hier, selbst in Bezug auf als gut bestätigt geltenden Theorien, immer ein gewisses Maß an Unsicherheit vor. Eine Möglichkeit, dieser Unsicherheit bei der Modellbildung Rechnung zu tragen, ist die Verwendung probabilistischer Methoden.

Bei den Fallstudien, die im weiteren Verlauf dieser Arbeit vorgestellt werden,

wird dazu die Methode der *Probabilistischen Programmierung* benutzt. Dabei handelt es sich um ein noch relativ junges Feld aus dem Bereich des maschinellen Lernens und insbesondere der künstlichen Intelligenz. Leider wird diese Bezeichnung nicht von allen Autoren in der gleichen Weise benutzt, ähnlich wie das schon bei Systemischen Risiken der Fall war. Eine einfache Suche im World Wide Web liefert mehr als 11000000 Suchergebnisse und schon auf der ersten Seite kann man Unterschiede beobachten. Es lassen sich aber zwei Gruppen ausmachen: die eher aus der Praxis kommenden *data scientists*, deren Fokus auf der Datenanalyse liegt, und die aus dem universitären Umfeld kommenden Forscherinnen und Forscher. Während die erste Gruppe aber Probabilistische Programmierung in einem engen Sinn als die Anwendung verschiedener statistischer Modelle, insbesondere aus der Klasse probabilistischer graphischer Modelle, betrachtet, hat die zweite Gruppe eine allgemeinere Sichtweise. Eine Probabilistische Programmiersprache wird hier als eine formale Sprache zur Repräsentation beliebiger Wahrscheinlichkeitsverteilungen verstanden. Ein Probabilistisches Programm muss hier nicht mehr einem bestimmten statistisches Modell folgen. Der Programmtext kann alles beinhalten, was in der formalen Sprache ausgedrückt werden kann. Aus diesem Grund erscheint es gerechtfertigt, zwischen Probabilistischer Programmierung in einem engeren bzw. in einem weiteren Sinn zu unterscheiden. Die vorliegende Arbeit beschäftigt sich mit Probabilistischer Programmierung im weiteren Sinn.

Häufig wird auch von *Probabilistischen Programmiersystemen* gesprochen, die aus einer *Probabilistischen Programmiersprache* und einer sogenannten *inference engine* bestehen. Letztere ist ein Programm, das in der Lage ist, spezielle Inferenzalgorithmen auf ein Probabilistisches Programm anzuwenden und so Werte aus der Verteilung zu ziehen, die das Programm repräsentiert. Mit Bezug auf die Betrachtungsebenen befinden sich Probabilistische Programmiersprachen auf der Ebene formaler Repräsentation, die Programme zur Ausführung der Inferenzalgorithmen dagegen auf der Ebene der Implementierung. Durch diese Herangehensweise wird eine Trennung von Modellbildung auf der einen Seite und Inferenz auf der anderen Seite erreicht. Für Letzteres werden Algorithmen aus der Familie der *Markov Chain Monte Carlo Simulation* verwendet (siehe dazu auch Kapitel 5.2).

Die Anzahl der Probabilistischen Programmiersprachen, die auf der Webseite [www.probabilistic-programming.org](http://www.probabilistic-programming.org) aufgelistet werden, ist beachtenswert. Zur Zeit

werden 30 solcher Sprachen genannt, von denen allerdings die meisten in die erste Gruppe gehören. Sie sind nur für bestimmte Arten von probabilistischen Modellen anwendbar und machen keine Trennung zwischen Formalisierung und Inferenz. Es existieren mittlerweile aber auch einige universelle Probabilistische Programmiersysteme. Das Erste dieser Art namens *Church* wurde von Goodman, Mansinghka, Roy, Bonawitz und Tenenbaum (2012) entwickelt. Als Weiterentwicklungen davon sind *Venture* (Mansinghka, Selsam & Perov, 2014) und *Anglican* (Tolpin, van de Meent, Yang & Wood, 2016) zu betrachten. Mit *WebPPL* wurde ein Probabilistisches Programmiersystem entwickelt, das besonders für die Anwendung auf Webseiten geeignet ist (Goodman & Stuhlmüller, 2014). Der Taxidienst *uber* ging 2018 mit einer eigens entwickelten Probabilistischen Programmiersprache namens *Pyro* an die Öffentlichkeit (Bingham et al., 2018). Die Beispiele in dieser Arbeit wurden in *Anglican* geschrieben. Das gesteigerte Interesse an der Probabilistischen Programmierung zeigt sich auch darin, dass sie Einzug in die universitäre Lehre erhält. Vorreiter ist fast schon traditionell das *MIT*, aber auch am *KAIST* in Südkorea und an der *LMU* in Deutschland werden Veranstaltungen angeboten. 2018 fand außerdem die erste internationale Konferenz zur Probabilistischen Programmierung statt.

## 5.1 Formale Repräsentation von Wahrscheinlichkeitsverteilungen

Eine Probabilistische Programmiersprache erlaubt also die Formalisierung beliebiger Modelle in Form von Wahrscheinlichkeitsverteilungen. Die Namensgebung ist insofern irreführend, wie Gordon, Henzinger, Nori und Rajamani (2014) anmerken, als dass “normale” Programme geschrieben werden, um ausgeführt zu werden. Probabilistische Programme dagegen werden geschrieben, um Wahrscheinlichkeitsverteilungen zu repräsentieren. Ein Probabilistisches Programm ist genau das: die formale Repräsentation einer Wahrscheinlichkeitsverteilung. Gleichsam kann das Probabilistische Programm dazu benutzt werden, Werte aus der Verteilung zu ziehen, die es beschreibt. Denn die Ausführung eines Probabilistischen Programms liefert einen Wert aus dieser Wahrscheinlichkeitsverteilung (mehr zur Ausführung im nächsten Kapitel). Man wird es deshalb nicht bei einer Ausführung belassen, sondern die Er-

gebnisse vieler Ausführungen betrachten, z.B. in Form eines Histogramms, um sich ein Bild der Verteilung machen zu können.

Interessanterweise können auf diese Art auch bedingte Wahrscheinlichkeitsverteilungen repräsentiert werden. Wenn also Daten vorliegen, die man bezüglich einer oder mehrerer Zufallsvariablen eines Probabilistischen Programms beobachtet hat, dann kann man diese Daten gewissermaßen in das Modell einbinden und auf diese Art die bedingte Wahrscheinlichkeitsverteilung des Modells gegeben die Daten repräsentieren. Mit Begriffen aus der Bayes-Statistik stellt ein Programm ohne Daten die Repräsentation der A-priori Verteilung, ein Programm mit Daten dagegen die A-Posteriori Verteilung eines Modells gegeben eben diese Daten dar.

## 5.2 Inferenz

Probabilistische Programme, so wurde im vorigen Abschnitt gesagt, werden ausgeführt, um einen Wert aus einer Wahrscheinlichkeitsverteilung zu liefern. So zumindest sieht es für einen Benutzer einer Probabilistischen Programmiersprache oberflächlich aus. Tatsächlich ist diese Beschreibung aber nicht korrekt. Denn ein Probabilistisches Programm repräsentiert eine im Voraus nicht bekannte Wahrscheinlichkeitsverteilung. Das Ziehen von Stichproben (englisch: *sampling*) muss daher mittels statistischer Inferenz erfolgen.

Es wird deshalb nicht das Probabilistische Programm selber ausgeführt, sondern ein Inferenzalgorithmus. Die verwendeten Algorithmen kommen aus der Familie der Markov-Chain-Monte-Carlo-Algorithmen (*MCMC*). Man kann zeigen, dass mit diesen Algorithmen unter bestimmten Voraussetzungen tatsächlich eine zunächst unbekannte Verteilung angenähert werden kann (siehe etwa Hastings, 1970, für eine detaillierte mathematische Betrachtung). Auf diese Art stellt das Probabilistische Programmiersystem sicher, dass die resultierenden Werte auch tatsächlich eine Approximation der gewünschten Wahrscheinlichkeitsverteilung darstellen und nicht einfach eine Folge von Zufallszahlen.

Mit Bezug auf Probabilistische Programmierung im engeren Sinn existieren bereits seit Ende der 80er Jahre Inferenzalgorithmen (McGrayne, 2014). Mit der Probabilistischen Programmierung im weiteren Sinn ist man nun nicht mehr an ein statistisches Modell gebunden. Stattdessen können alle verfügbaren Sprachkonstrukte

verwendet werden, um einen beliebigen Programmtext zu erstellen. Da es sich bei Probabilistischen Programmiersprachen um sogenannte *Turing-vollständige* Sprachen handelt, gibt es hier kaum Restriktionen. Alles, was in jeder anderen Programmiersprache geschrieben werden kann, kann auch in ein Probabilistisches Programm eingehen, mit dem Unterschied, dass das Probabilistische Programm an verschiedenen Stellen nicht-deterministische Teile enthält.

Die Inferenzalgorithmen des Probabilistischen Programmiersystems sind in der Lage, Inferenz bezüglich des Quellcodes durchzuführen. An diesem Punkt wird auch deutlich, warum von Probabilistischen Programmiersystemen gesprochen wird. Sie bestehen zum einen aus einer Probabilistischen Programmiersprache und zum anderen aus einem Programm, dass Inferenzalgorithmen auf einen Programmtext anwenden kann. Die Entwicklung von neuen, speziellen Inferenzalgorithmen innerhalb der künstlichen Intelligenzforschung macht die Probabilistische Programmierung im weiteren Sinn möglich. Es existieren bereits einige Varianten (Rainforth et al., 2016; van de Meent, Yang, Mansinghka & Wood, 2015; Tolpin, van de Meent, Paige & Wood, 2015; Tolpin & Wood, 2015; Paige & Wood, 2014).

### 5.3 Expressivität

Wenn von der Repräsentation von Wahrscheinlichkeitsverteilungen die Rede ist, dann sind damit nicht Verteilungen wie die Normalverteilung oder die Binomialverteilung gemeint. Es wäre zwar möglich, diese schon lange bekannten und gut untersuchten Verteilungen mit Probabilistischen Programmen zu beschreiben, aber keineswegs notwendig. Zum einen sind sie mit mathematisch-analytischen Methoden handhabbar, zum anderen existieren hier effiziente Methoden zur Generierung von Werten. Vielmehr geht es um Wahrscheinlichkeitsverteilungen, die aus verschiedenen Zufallsvariablen zusammengesetzt sind und zwischen denen es vielfältige Wechselwirkungen gibt. Dabei sind sowohl Anzahl und Art der Zufallsvariablen als auch der Wechselwirkungen beliebig. Ferner ist es möglich, sollte es sich um die Beschreibung eines Prozesses handeln, dass sich Art und Anzahl verändern, und zwar sowohl in Hinsicht auf die Zufallsvariablen als auch auf deren Wechselwirkungen. Die Entwicklung eines Prozesses könnte etwa in Abhängigkeit der Ausprägung einer Zufallsvariablen einen von mehreren qualitativ unterschiedlichen Verläufen annehmen,

die sich keinerlei Überschneidung hinsichtlich Zufallsvariablen und Wechselwirkungen aufweisen. Solange eine Anwenderin in der Lage ist, den Prozess in Textform oder in Form eines Flussdiagramms zu beschreiben, kann auch ein Probabilistisches Programm geschrieben werden.

Diese Eigenschaft wird in der Literatur unter dem Stichwort *Expressivität* behandelt. Es wird möglich, auch komplexe Modelle zu formalisieren, die mit mathematisch-analytischen Methoden nur schwer oder vielleicht auch gar nicht zu fassen sind. Das macht sie zu einem Kandidaten für die Modellierung in den Sozialwissenschaften insgesamt und für Systemische Risiken im Speziellen.

## 5.4 Einordnung ins Ebenenmodell

Die Ideen, die hinter der Entwicklung von *Probabilistischen Programmiersprachen* stehen, weisen einige Ähnlichkeit mit Aspekten auf, die bei der Differenzierung nach unterschiedlichen Betrachtungsebenen diskutiert wurden. Probabilistische Modellierung wird bisher vor allem in Zusammenhang mit statistischer Datenanalyse betrieben, die in einem weiten Sinn auch die Bereiche des maschinellen Lernens und der künstlichen Intelligenz umfasst. Je nach Fragestellung kommen Verfahren der Regression, der Klassifikation, der Clusteranalyse oder der Dimensionsreduzierung in Betracht, um einige der eher traditionellen statistischen Verfahren zu nennen. In den letzten Jahren sind außerdem Verfahren auf Basis von neuronalen Netzwerken in Mode gekommen (Mansinghka, 2015). Mit Bezug auf die Betrachtungsebenen aus Kapitel 4.1.1 sind diese Verfahren der Ebene der Implementierungen zuzuordnen. Probabilistische Programmiersprachen dagegen gehören zur Ebene formaler Repräsentation, sie können zur formalen Repräsentation beliebiger Modelle herangezogen werden. Die Implementierung erfolgt durch die oben erwähnte *inference engine*, ein Programm, das in der Lage ist, Inferenz bezüglich des Programmtextes zu betreiben. Ein *Probabilistisches Programmiersystem* beinhaltet beides und umspannt in diesem Sinne sowohl die Ebene formaler Repräsentationen als auch die Ebene theoretischer Abstraktion.

# Kapitel 6

## Fallbeispiel 1: Ausbreitung von Influenza

Das erste Probabilistische Programm zur Modellierung eines Systemischen Risikos wird in diesem Kapitel vorgestellt. Am Beispiel der Ausbreitung von Influenza wurde dazu ein Modell entwickelt, das die Verteilung der wöchentlichen Neuinfektionen im Verlauf einer Grippe-Saison repräsentiert. Ausgangspunkt bei der Entwicklung des probabilistischen Modells war das SIR-Modell (Kermack & McKendrick, 1927), das seit seiner Einführung in die Literatur in der Epidemiologie weite Verbreitung gefunden hat (siehe beispielsweise Weidemann et al., 2017; Balcan et al., 2010; Coburn, Wagner & Blower, 2009; Chowell, Miller & Viboud, 2008).

Die Ausbreitung einer ansteckenden Krankheit und, damit einhergehend, die Möglichkeit einer Epidemie steht im Einklang mit dem Rahmenmodell aus Kapitel 4.4. Das System  $S$ , das betrachtet wird, ist hier die Bevölkerung, aufgeteilt in drei disjunkte Kompartimente  $S$ ,  $I$  und  $R$  (Genaueres dazu wird im weiteren Verlauf des Kapitels gesagt). Auch wenn die Influenza prinzipiell überall auf der Welt auftreten kann, liegt der Fokus in dieser Arbeit auf der Ausbreitung innerhalb der deutschen Bevölkerung. Bei Ausbruch einer Epidemie ist in der Regel ein großer Teil der Bevölkerung betroffen. In diesem Sinne ist der Aspekt der Globalität gegeben. Das rekursive Element ist die Anzahl der infizierten Individuen, die durch das  $R$ -Kompartiment repräsentiert werden. Da jedes infizierte Individuum neue Ansteckungen generieren kann und diese Neuinfektionen dann, zu einem späteren Zeitpunkt, zur vorherigen

Anzahl dazukommen, ist hier eine rekursive Beziehung gegeben. Daraus folgt dann auch die Dynamik, die dem Beispiel innewohnt.

Das SIR-Modell wird typischerweise als ein System von Differentialgleichungen formalisiert. Die Repräsentation in Form eines Probabilistischen Programms und damit gleichbedeutend in Form einer Wahrscheinlichkeitsverteilung ist in Bezug auf das Ebenenmodell aus Kapitel 4.1.1 lediglich eine Alternative auf der Ebene formaler Repräsentationen.

## 6.1 Allgemeine Beschreibung des SIR-Modells

### Kompartimente

Das SIR-Modell wurde entwickelt als ein theoretisches Mittel, um den Verlauf einer infektiösen Krankheit innerhalb einer Population zu prognostizieren. Dabei wird die Population in drei disjunkte Gruppen, auch Kompartimente genannt, eingeteilt, die mit  $S$ ,  $I$  und  $R$  bezeichnet werden.  $S$  steht für *susceptible*, auf deutsch: *empfänglich*. Bei dieser Gruppe handelt es sich um alle Individuen der Population, die sich potentiell mit der Krankheit infizieren können.  $I$  steht für *infected* und damit für alle Personen, die bereits infiziert sind. Und schließlich steht  $R$  für *removed*, oder wahlweise auch *recovered*, und bezeichnet all die Individuen, die nicht mehr krank sind, entweder weil sie gesund und nun immun oder aber, weil sie an der Krankheit gestorben sind.

Das SIR-Modell beschreibt nun einen Prozess, der sich zeitlich entfaltet und in dessen Verlauf sich die Anzahl der Individuen in den Kompartimenten verändert bzw. verändern kann. Zu jedem Zeitpunkt wechseln Individuen aus dem  $S$ - in das  $I$ -Kompartiment bzw. aus dem  $I$ - in das  $R$ -Kompartiment. Prinzipiell ist es auch möglich, dass gesundene Individuen nicht immun, sondern wieder empfänglich für die Krankheit werden. In diesem Fall würden sie dann wieder ins  $S$ -Kompartiment wechseln. In der vorliegenden Arbeit wird aber vereinfachend davon ausgegangen, dass dies nicht der Fall ist.

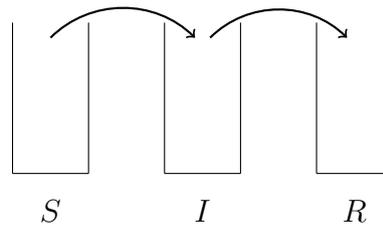


Abbildung 6.1: Die Kompartimente des SIR-Modells. Zu jedem Zeitpunkt erkranken Individuen und wechseln dadurch vom S- in das I-Kompartiment. Gleichzeitig wechseln Individuen vom I- in das R-Kompartiment, weil sie entweder immun geworden oder aber an der Krankheit gestorben sind.

### Progressionen von Kohorten

Das SIR-Modell wird typischerweise, wie oben schon gesagt, als System von Differentialgleichungen formalisiert, wobei für jedes Kompartiment eine Gleichung aufgestellt wird. Die Herleitung dieses Modells wird von Kermack und McKendrick (1927) detailliert beschrieben. Die Einzelheiten der mathematischen Formalisierung sind an dieser Stelle nicht Gegenstand der Betrachtung. Von großer Bedeutung für die vorliegende Arbeit ist folgende grundlegende Idee, auf der die Entwicklung des Modells aufbaut. Kermack und McKendrick (1927) beginnen mit der Festlegung eines initialen Zeitpunkts  $t_0$ , zu dem es bereits eine Anzahl infizierter Individuen gibt. Diese Individuen stecken nun andere, bislang gesunde Individuen an, so dass es zum Zeitpunkt  $t_1$  eine Anzahl von Neuinfektionen gibt. Diejenigen Individuen, die zum Zeitpunkt  $t_1$  neu infiziert wurden, stecken nun ihrerseits weitere Individuen an. Von den Individuen, die zum Zeitpunkt  $t_0$  infiziert wurden, sind einige zu  $t_1$  bereits nicht mehr infiziert. Die Übrigen sind weiterhin potentielle Überträger der Krankheit und generieren ebenfalls zu  $t_1$  neue Fälle. Die Anzahl der Neuinfektionen zum Zeitpunkt  $t_2$  setzt sich deshalb zusammen aus denen, die durch Individuen der ursprünglichen Gruppe und denen, die durch Individuen der zu  $t_1$  infizierten Gruppe generiert werden.

Dieser Prozess setzt sich nun weiter fort, wobei zu jedem Zeitpunkt eine neue Gruppe von frisch infizierten Personen hinzukommt. Die Gruppe der Individuen, die zu einem beliebigen Zeitpunkt  $t_i$  infiziert werden, wird im Folgenden als *Kohorte* bezeichnet. Eine Kohorte “wandert” durch die Zeit und generiert dabei neue Fälle.

Gleichzeitig verliert die Kohorte im Laufe der Zeit auch Mitglieder, entweder weil diese gesund werden oder aber, weil sie sterben. Diese Entwicklung setzt sich so lange fort, bis alle Mitglieder aus einer Kohorte ausgeschieden sind. Eine graphische Darstellung dieses Prozesses zeigt Abbildung 6.2.

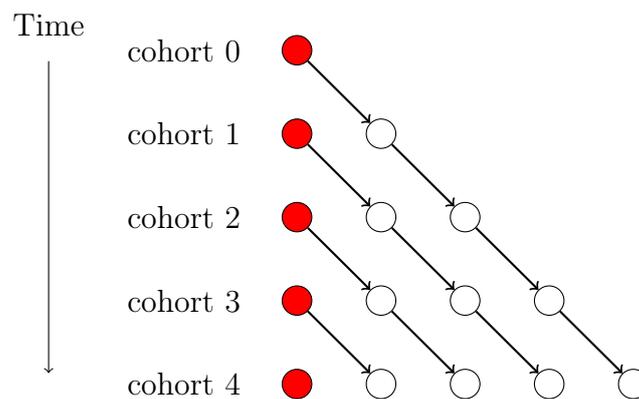


Abbildung 6.2: Progressionen der Kohorten. Zu Beginn des Prozesses gibt es eine Kohorte, bestehend aus bereits infizierten Individuen. Diese Kohorte “wandert” durch die Zeit, wobei zu jedem Zeitpunkt einige Individuen gesund oder sterben und aus der Kohorte fallen. Die Übrigen generieren neue Fälle, so dass zu jedem Zeitpunkt eine neue Kohorte entstehen kann. Die Gesamtzahl der Neuinfektionen zum Zeitpunkt  $t_i$  besteht aus der Summe aller Individuen derjenigen Kohorten, die zu  $t_i$  noch existieren.

Während Kermack und McKendrick (1927) auf Basis dieser Grundüberlegungen ein Differentialgleichungssystem aufstellen, das den Verlauf der Krankheit auf einer kontinuierlichen Zeitachse deterministisch beschreibt, wird hier ein anderer Weg gegangen. Ausgangspunkt sind die Progressionen der Kohorten.

## 6.2 Beschreibung des probabilistischen SIR-Modells

Der Ansatz für das probabilistische SIR-Modell besteht in der Umsetzung der oben beschriebenen Progressionen von Kohorten in Form eines Probabilistischen Programms. Die Anzahl von Individuen, die zu den einzelnen Zeitpunkten den Kompartimenten zugeordnet werden, wird dabei als Zufallsvariablen modelliert. Die Wahl

der Intervalle, in der die Zeit in diesem Modell voranschreitet, ist für die Entwicklung des Probabilistischen Programms grundsätzlich beliebig. Im Einklang mit den Daten, die im Zusammenhang mit der Influenza in Deutschland vom Robert Koch-Institut zusammengefasst und herausgegeben werden, wird bei der Beschreibung des Modells von der Anzahl der neuen Fälle pro Woche die Rede sein. Genau wie bei der Modellierung auf Basis von Differentialgleichungen hängen auch die Zufallsvariablen, die die Kompartimente repräsentieren, gegenseitig voneinander ab.

### 6.2.1 Entwicklung des I-Kompartiments

Die Entwicklung des I-Kompartiments über die Zeit steht im Zentrum des Modells. Sie hängt von zwei Größen ab: der Anzahl der Neuinfektionen pro Woche und der Anzahl derer, die im selben Zeitraum genesen oder sterben und deshalb von diesem Zeitpunkt an dem R-Kompartiment zugeordnet werden müssen. Es müssen also sowohl die Neuzugänge als auch die Abgänge berücksichtigt werden. In diesem Abschnitt wird zunächst detailliert die Modellierung der Neuerkrankungen pro Woche erläutert. Die Modellierung der Abgänge wird in Kapitel 6.2.2 erläutert.

#### Neuerkrankungen pro Individuum pro Woche

**Konzeption als Zufallsvariable** Das Modell wird zunächst aus einer Mikroperspektive heraus entwickelt und betrachtet die Anzahl der neuen Fälle, die ein bereits infiziertes Individuum generiert. Diese wird im Folgenden als  $I_{Ind}$  bezeichnet.  $I_{Ind}$  wird als Zufallsvariable modelliert. Die Gründe hierfür sind vielfältig. So kommt etwa nicht jede Person mit gleich vielen anderen Personen in Kontakt. Und auch bei einem Kontakt kommt es nicht in jedem Fall zur Übertragung der Krankheit. Es ist deshalb notwendig, hier eine gewisse Varianz anzunehmen.

Es ist ebenfalls plausibel, dass sich die Häufigkeit der Übertragung mit der Dauer der Erkrankung verändert. Die Inkubationszeit der Influenza beträgt 1 bis 4 Tage. So lange kann es dauern, bis sich die ersten Symptome zeigen. Während dieser Zeit werden infizierte Personen ihrem Alltag nachgehen, zur Arbeit gehen, sich mit Freunden treffen etc. Da sie das Virus in sich tragen, kann es auch übertragen werden. In diesem Stadium kommt es deshalb zu vielen Kontakten zu anderen Personen und das Virus kann potentiell auf viele andere Personen übertragen werden. Ist bei einer

Person die Krankheit manifestiert, so wird sie eher das Bett hüten und deshalb auch nicht mehr so viele Kontakte zu anderen Personen haben. Die Zahl der durch diese Person generierten Fälle wäre dann geringer.

**Primäre und sekundäre Anzahl neuer Fälle** Aus diesen Überlegungen heraus muss  $I_{Ind}$  in Abhängigkeit des Zeitpunkts der Infizierung bzw. der Dauer der Erkrankung betrachtet werden. Anders gesagt: die Wahrscheinlichkeitsverteilung, der die Zufallsvariable  $I_{Ind}$  folgt, muss sich im Laufe der Simulation verändern. Das hier vorgestellte Modell beschränkt sich dabei auf die Unterscheidung von zwei Zeitpunkten: die Woche, in der die Infizierung stattgefunden hat und die Zeit ab der Woche danach. Die zugehörigen Zufallsvariablen werden mit  $I_{primary}$  und  $I_{secondary}$  bezeichnet.

Der erste Schritt bei der Modellierung besteht nun in der Wahl von geeigneten Wahrscheinlichkeitsverteilungen für  $I_{primary}$  bzw.  $I_{secondary}$ . Überraschend erschien dabei zunächst, dass es zumindest dem Autor nicht möglich war, Literatur zu dieser Frage zu finden. Bei näherer Betrachtung fällt allerdings auf, dass es sich hier um ein schwer zu messendes Phänomen handelt. Denn man müsste den Zeitpunkt der Infektion von mindestens einer Zielperson kennen und anschließend alle Personen, mit denen die Zielperson seitdem in Kontakt war und für die Dauer der Infektion in Kontakt kommt, dahingehend überprüfen, ob sie sich ebenfalls infiziert haben. Es ist nachvollziehbar, dass ein solches Forschungsvorhaben mit vielerlei Problemen behaftet wäre. Selbst wenn man etwa das Problem, den Zeitpunkt der Infektion zu erfahren, als gelöst annimmt, müsste noch ausgeschlossen werden, dass die Infektion auch tatsächlich durch die Zielperson zu Stande kam und nicht auf andere Weise erfolgt ist. Wir haben es hier also mit einer *latenten Variablen* zu tun, die nicht direkt gemessen, sondern nur geschätzt werden kann (zum Begriff der latenten Variablen siehe Borsboom, 2008). Es war aber zumindest möglich, bei der Suche nach einer Wahrscheinlichkeitsverteilung auf Informationen aus verschiedenen Gesprächen mit Medizinern und im Gesundheitswesen tätigen Personen zurückzugreifen.

**Poisson Verteilung** Es stellt sich also zunächst die Frage, durch welche Wahrscheinlichkeitsverteilungen  $I_{primary}$  bzw.  $I_{secondary}$  beschrieben werden können. Klar ist, dass die untere Schranke dieser Zufallsvariablen bei Null liegen muss. Weniger als

niemanden kann ein Individuum nicht infizieren. Man kann ferner annehmen, dass  $I_{secondary}$  tendentiell niedrigere Werte annehmen soll als  $I_{primary}$ . Eine obere Schranke anzugeben ist dagegen schwieriger. Man kann nicht ausschließen, dass unter gewissen Umständen auch eine zweistellige Zahl an neu generierten Fällen auftritt, zumal die neuen Fälle über einen Zeitraum von einer Woche angesteckt werden können. Eine Zahl von beispielsweise 20 neu generierten Fällen erscheint aber auf der anderen Seite als wenig wahrscheinlich. Die Verteilung sollte also solche Werte nicht ausschließen, aber eine höhere Konzentration bei deutlich niedrigeren Werten haben. Was könnte also eine häufige Anzahl von neuen Fällen pro Individuum in der ersten Woche der Erkrankung sein? In Ermangelung von Literaturangaben zu diesem Thema und auf Basis der oben genannten Expertenmeinungen werden in dem hier vorgestellten Modell Werte in einem Bereich von 0 bis 4 als typisch betrachtet, die aber nicht gleichverteilt sein sollten.

Eine Wahrscheinlichkeitsverteilung, die diese Anforderungen erfüllt, ist die *Poisson-Verteilung*. Die Poisson-Verteilung ist eine diskrete Wahrscheinlichkeitsverteilung, die durch einen Parameter bestimmt ist. Dieser Parameter wird im Allgemeinen mit  $\lambda$  bezeichnet und darf nur nicht-negative Werte annehmen. Die Verteilung ordnet den natürlichen Zahlen  $k = 0, 1, 2, 3, \dots$  Wahrscheinlichkeiten zu, Erwartungswert und Varianz der Verteilung werden durch  $\lambda$  beschrieben. Die Poisson-Verteilung ist nach oben nicht beschränkt, je nach Wahl von  $\lambda$  sind hohe Werte aber sehr unwahrscheinlich. So beträgt etwa die Wahrscheinlichkeit, dass die Realisation einer Poisson-verteilten Zufallsvariablen  $X$  mit Parameter  $\lambda = 2$  einen Wert von 10 annimmt,  $P(X = 10) = 0.000038$ .

Damit die Poisson-Verteilung auch tatsächlich zu einer guten Annäherung kommt, muss der Parameter  $\lambda$  so gewählt werden, dass Werte um 2 oder 3 herum am häufigsten auftreten und höhere Werte seltener vorkommen können. Die Verwendung der Poisson-Verteilung bietet den Vorteil, dass ihr Parameter inhaltlich interpretiert werden kann als der arithmetische Mittelwert der neu generierten Fälle. Aufgrund der Eigenschaft, dass die Poisson-Verteilung nach oben nicht beschränkt ist, wird der arithmetische Mittelwert tendentiell etwas höher liegen als der Modalwert, also der am häufigsten auftretende Wert. In Abbildung 6.3 sind Beispiele für Poisson-Verteilungen mit Parametern  $\lambda_1 = 1.0$  bzw.  $\lambda_2 = 2.0$  zu sehen. Man sieht an diesen Abbildungen die Auswirkungen der Wahl des Parameters. Die auftretenden Werte

liegen in beiden Fällen in einem ähnlichen Bereich. Die am häufigsten vorkommenden Werte der Poisson-Verteilung mit Parameter  $\lambda_1 = 1.0$  sind aber gegenüber der anderen Verteilung etwas kleiner.

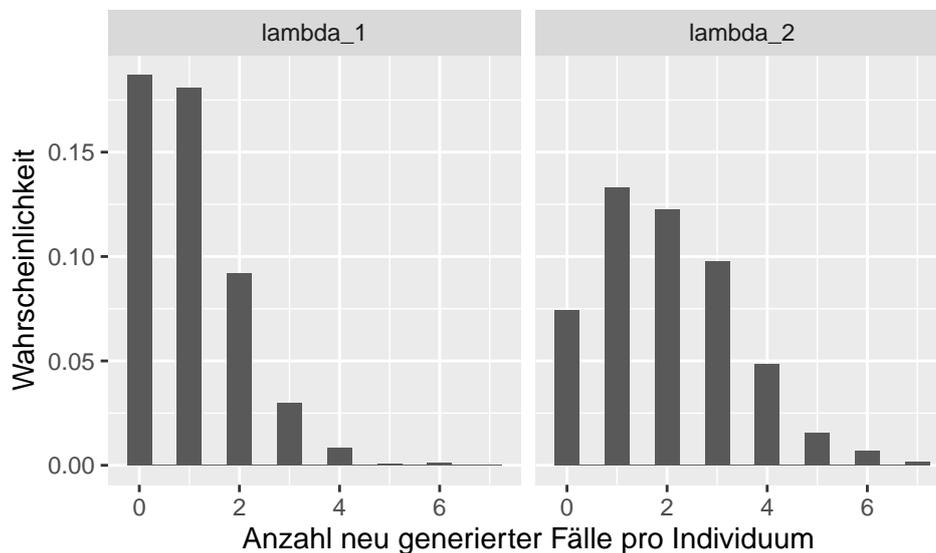


Abbildung 6.3: Histogramme für Poisson-Verteilungen mit Parameter 1.0 bzw. 2.0.

**Verteilung des Parameters** Beide Verteilungen aus Abbildung 6.3 sind Kandidaten für die Verteilungen der Zufallsvariablen  $I_{primary}$  und  $I_{secondary}$ . Sie erfüllen die Anforderungen, die weiter oben genannt wurden. Es stellt sich daher die Frage, welche der Verteilungen in welchem Fall angemessener ist. Eine Festlegung erscheint in gewisser Weise willkürlich und muss es vielleicht sogar sein. Da es keine gesicherten Erkenntnisse über die Häufigkeit der Übertragung gibt, kann man sich nicht auf empirische Befunde stützen. Eine Möglichkeit, mit diesem Problem umzugehen, liegt darin, auch den Parameter der Poisson-Verteilung als Zufallsvariable zu modellieren. Solche Parameter, die eine Verteilung bestimmen und selbst als Zufallsvariable behandelt werden, werden oft als *Hyperparameter* bezeichnet (Gelman et al., 2013).

Damit entsteht nun allerdings ein neues Problem, nämlich die Wahl der Wahrscheinlichkeitsverteilung für die Hyperparameter. Um den Wertebereich des Parameters lediglich einzuschränken, könnte man eine Gleichverteilung wählen, deren obere Grenze so hoch gewählt ist, dass man auf keinen Fall mögliche Werte ausschließt.

Das entspricht einer sogenannten *nicht-informativen* bzw. manchmal auch *schwach informativ* genannten A-priori Verteilung (Gelman et al., 2013). Nicht informativ ist sie deshalb, weil keinerlei Vorwissen in die Spezifikation eingeht. Alle möglichen Werte werden mit der gleichen Wahrscheinlichkeit angenommen. Das Gegenstück ist die *informative* A-Priori Verteilung, die bereits vorhandenes Wissen mit einbezieht. Es kann diskutiert werden, welches Vorgehen das Geeigneterere ist. Auf der Grundlage einiger Testläufe und da eine obere Grenze irgendwo im niedrigeren zweistelligen Bereich in diesem Fall sinnvoll erscheint, fiel die Wahl für die Verteilung der Hyperparameter  $\lambda_1$  und  $\lambda_2$  auf eine Gleichverteilung mit den Grenzen 0.4 und 0.9 bzw. 1.0 und 1.9. Die Verteilungen sind in Abbildung 6.4 zu sehen.

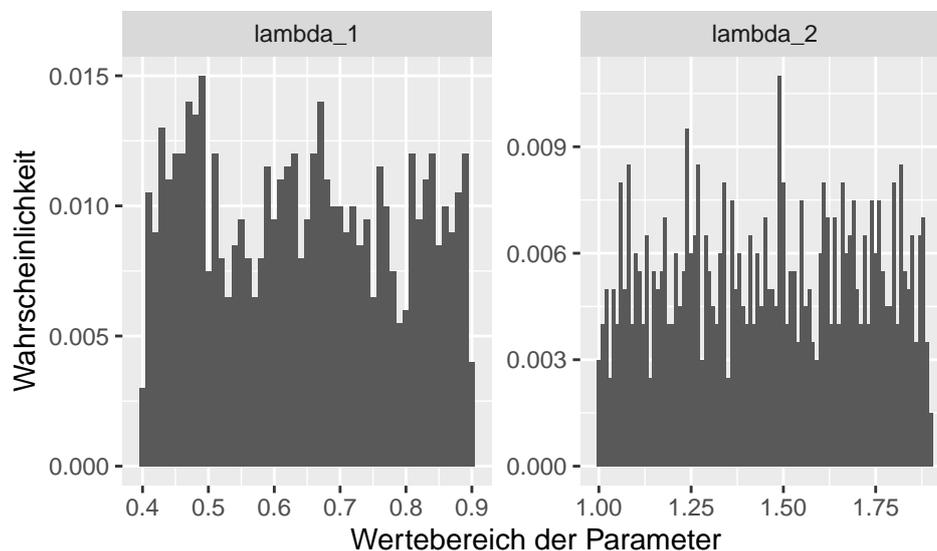


Abbildung 6.4: Histogramme der A-priori Verteilungen von  $\lambda_1$  und  $\lambda_2$ .

### Gesamtzahl der Neuerkrankungen pro Woche

Die Gesamtzahl der Neuerkrankungen zu einem gegebenen Zeitpunkt  $t$ , und damit die Größe der Kohorte dieses Zeitpunkts, wird als  $I_t$  bezeichnet und als Summe der Zufallsvariablen  $I_{Ind}$  über alle Individuen konzipiert, die zum Zeitpunkt  $t$  infiziert sind. Dabei kann eine Eigenschaft der Poisson-Verteilung benutzt werden. Für  $n$  Poisson-verteilte Zufallsvariablen  $Y_i$  mit Parametern  $\lambda_i$  gilt nämlich:

$$\sum_{i=0}^n Y_i = Z \quad (6.1)$$

$$Z \sim \text{Pois}(\lambda_Z) \quad (6.2)$$

$$\lambda_Z = \sum_{i=0}^n \lambda_i. \quad (6.3)$$

In Worten ausgedrückt bedeutet das, dass die Summe mehrerer Poisson-verteilter Zufallsvariablen  $Y_i$  einer Poisson-verteilten Zufallsvariablen  $Z$  entspricht (Gleichungen 6.1 und 6.2), deren Parameter  $\lambda_Z$  die Summe der Parameter aller  $Y_i$  ist (Gleichung 6.3). Diese Eigenschaft macht es möglich, die Gesamtzahl der Neuinfektionen pro Woche  $I_t$  als eine Poisson-verteilte Zufallsvariable zu modellieren. Da neu generierte Fälle der einzelnen Individuen als Poisson-verteilte Zufallsvariablen mit Parametern  $\lambda_{\text{primary}}$  bzw.  $\lambda_{\text{secondary}}$  konzipiert werden, gilt für  $I_t$ :

$$I_t \sim \text{Pois}(\lambda_{I_t}) \quad (6.4)$$

$$\lambda_{I_t} = n_{\text{primary}}\lambda_{\text{primary}} + n_{\text{secondary}}\lambda_{\text{secondary}}. \quad (6.5)$$

Die Summe der neu generierten Fälle aller infizierten Individuen pro Woche  $t$  kann also als eine Poisson-verteilte Zufallsvariable  $I_t \sim \text{Pois}(\lambda_{I_t})$  modelliert werden (Gleichung 6.4), wobei der Parameter  $\lambda_{I_t}$  die Summe der Parameter der einzelnen Individuen ist. Weil im Modell zwischen neu infizierten und seit mindestens einer Woche erkrankten Personen unterschieden wird, muss für alle  $n_{\text{primary}}$  neu infizierten Personen bzw. für die übrigen  $n_{\text{secondary}}$  schon länger Infizierten jeweils der entsprechende Parameter in die Summe eingehen. Dabei muss für die Gesamtzahl  $n$  der infizierten Individuen stets gelten, dass  $n = n_{\text{primary}} + n_{\text{secondary}}$  ist (Gleichung ??).

**Impfquoten** Bei der Modellierung der Übertragung der Infektion müssen auch die Impfquoten beachtet werden. In Deutschland gibt es keine Impfpflicht. Eine jährliche Gripeschutzimpfung ist aber empfohlen und wird kostenlos angeboten. Ein Teil der Bevölkerung nimmt dieses Angebot an. In der Saison 2015/16 lag die bundesweite

Impfquote bei 35.3% (Rieck, Feig, Wichmann & Siedler, 2017). Es gibt nun verschiedene Möglichkeiten, diese im Modell zu berücksichtigen. Im hier vorgestellten Modell läuft der Ansteckungsprozess der einzelnen Individuen zunächst “ungehindert” von möglichen Impfquoten ab. Anschließend wird von der Gesamtzahl der berechneten neuinfizierten Fälle ein gewisser Anteil wieder abgezogen, der die geimpften Personen repräsentiert. Dieser Anteil wird, nicht überraschenderweise, als Zufallsvariable konzipiert, die einer Binomialverteilung folgt. Die Binomialverteilung hat einen Parameter, der die Anzahl derjenigen angibt, die geimpft sind und deshalb von der ursprünglichen Zahl der Infizierten wieder abgezogen werden muss. Der Parameter wird dabei mit 0.2 niedriger als die bundesweite Impfquote angesetzt. Der Grund dafür ist, dass sich die Impfquote auf die Gesamtbevölkerung bezieht und nicht auf den Teil der mit dem Virus in Kontakt gekommenen Personen.

**Einbindung von Daten** Die Festlegung auf einen oder mehrere Parameter, selbst im Fall von als Zufallsvariablen spezifizierten Hyperparametern, stellt eine Annahme dar, die wie alle Annahmen auch falsch sein kann. Mit der Verwendung der Probabilistischen Programmierung hat man nun die Möglichkeit, in gewisser Weise einen Abgleich dieser A-priori Annahmen mit empirischen Daten vorzunehmen. Dabei wird ein Bayes-statistischer Ansatz verwendet. Bezeichnet man das Modell, das durch das Probabilistische Programm beschrieben wird, mit  $\mathcal{M}$  und die Daten mit  $D$ , dann gilt mit dem Satz von Bayes

$$P(\mathcal{M}|D) = \frac{P(D|\mathcal{M})P(\mathcal{M})}{P(D)}. \quad (6.6)$$

Dabei steht  $P(\mathcal{M})$  für die A-priori Verteilung des Modells, also für das Modell inklusive aller Annahmen, die man bei der Formulierung gemacht hat.  $P(\mathcal{M}|D)$  dagegen steht für die A-posteriori Verteilung des Modells gegeben die Daten, für deren Zustandekommen die Ergebnisse der einzelnen Programmläufe anhand der Daten restringiert wurden. Wie genau das geschieht, soll an dieser Stelle nicht weiter vertieft werden, eine Einführung findet sich bei van de Meent, Paige, Yang und Wood (2018).

Wie in Kapitel 5 bereits dargelegt wurde, besteht der Vorteil der Probabilistischen Programmierung gegenüber anderen, inferenzstatistischen Methoden darin,

dass man bei der Formulierung nicht mehr auf statistische Modelle beschränkt ist, für die es spezielle Software gibt. Stattdessen wird das Modell als Quelltext in der Probabilistischen Programmiersprache geschrieben und kann auf alle Konstrukte und Möglichkeiten selbiger zurückgreifen. Dadurch wird es möglich, ein Modell, wie das hier beschriebene, nicht nur ohne Daten als probabilistisches Simulationsmodell zu verwenden, sondern statistische Inferenz auf Basis empirischer Daten zu betreiben.

### 6.2.2 Entwicklung des R-Kompartiments

In jeder Woche werden nicht nur neue Individuen infiziert, sondern bereits infizierte Personen werden gesund oder sterben an den Folgen der Krankheit. Im hier beschriebenen Modell werden diese Fälle nicht unterschieden. Es wird vielmehr eine Zufallsvariable definiert, die all diejenigen Personen identifiziert, die von einem gegebenen Zeitpunkt an nicht mehr infiziert sind. Diese Individuen werden dann dem R-Kompartiment zugeordnet. Bei der Wahl einer Verteilung für diese Zufallsvariable geht es demnach vor allem darum, die Dauer der Erkrankung anzugeben.

Die Anzahl  $R_{t_{cohort_i}}$  der Individuen einer Kohorte, die ab einem gegebenen Zeitpunkt  $t$  nicht mehr dem I-, sondern dem R-Kompartiment zuzuordnen sind, wird gemäß den in Kapitel 6.1 beschriebenen Progressionen auf der Ebene einer Kohorte formuliert.  $R_{t_{cohort_i}}$  wird als Binomial-verteilte Zufallsvariable aufgefasst, deren Parameter auf 0.45 fixiert ist. Ähnlich wie bei der Generierung von neuen Fällen gibt es auch in diesem Punkt keine gesicherten Erkenntnisse. Der Parameter wurde deshalb auf Basis von Gesprächen mit Experten gewählt. Die Gesamtzahl  $R_t$  der Individuen, die in einer Woche  $t$  in das R-Kompartiment wechseln, wird dann durch Summierung über alle Kohorten gebildet. Wenn die Anzahl der Kohorten mit  $n$  bezeichnet wird, dann gilt:

$$R_{t_{cohort_i}} \sim \text{Binomial}(p = 0.45) \quad (6.7)$$

$$R_t \sim \sum_{i=1}^n R_{t_{cohort_i}}. \quad (6.8)$$

### 6.2.3 Entwicklung des S-Kompartiments

Die Anzahl der Individuen innerhalb der Population, die nicht infiziert sind, aber angesteckt werden können, ergibt sich aus dem, was in Kapitel 6.2.1 über die Entwicklung der Anzahl infizierter Individuen dargelegt wurde. Für jede Woche  $t$  verringert sich die Anzahl der Personen im S-Kompartiment dementsprechend um die Anzahl der Neuinfektionen in dieser Woche. Für eine Population der Größe  $N$  ist die Anzahl der empfänglichen Personen  $S_t$  in einer Woche  $t$  durch die Zufallsvariable

$$S_t = S_{t-1} - I_t$$

gegeben, wobei  $S_0 = N$  ist.

## 6.3 Beschreibung der empirischen Daten

Ein Grund für die Wahl der Ausbreitung von Influenza als Beispiel lag darin, dass es in diesem Zusammenhang Daten gibt, die in das Probabilistische Programm eingebunden werden. Die Ergebnisse von Simulationsläufen unter Berücksichtigung von Daten entsprechen dann der A-posteriori Verteilung des Modells gegeben diese Daten. Die Daten selber stammen vom RKI. Das RKI ist das deutsche Bundesinstitut für Infektionskrankheiten und als solches auch für die Beobachtung des Auftretens von Influenza zuständig. Dazu sammelt die Arbeitsgemeinschaft Influenza des RKI Daten zur Entwicklung der Influenzaaktivität in Deutschland aus verschiedenen Quellen und veröffentlicht diese sowohl in wöchentlichen als auch in saisonalen Berichten. In dieser Arbeit wurden Daten aus der Grippesaison 2015/16 verwendet (RKI, 2015)

In den saisonalen Berichten werden die Datenquellen und der Prozess der Datenerfassung detailliert beschrieben RKI (2015). Auf die Einzelheiten dieses komplexen Prozesses wird an dieser Stelle nicht näher eingegangen. Hervorzuheben sind im Zusammenhang mit der vorliegenden Arbeit die Daten zu den Erstkonsultationen von Patienten mit Atemwegserkrankungen (ARE). Bei diesem ARE-Konsultationsindex handelt es sich um diejenigen Patienten mit ARE, die in einer gegebenen Woche zum ersten Mal in einer der an der Datenerfassung beteiligten Arztpraxen vorstellig werden. Von diesen Patienten haben sich allerdings nicht alle tatsächlich mit

der Influenza angesteckt. Um die Anzahl der Influenzafälle zu ermitteln, werden pro Arztpraxis eine Reihe von Abstrichen an das *Nationale Referenzzentrum (NRZ)* geschickt, wo dann die Positivenrate ermittelt wird. Auch die Positivenraten pro Woche werden in den saisonalen Berichten veröffentlicht.

Zur Abschätzung der Influenzaaktivität sind sowohl der ARE-Konsultationsindex als auch die Positivenrate denkbar. Die Positivenrate gibt zwar tatsächliche Infektionen wieder, bezieht sich aber nur auf die eingesandten Proben. Der Konsultationsindex berücksichtigt dagegen auch Patienten, die nicht mit Influenza infiziert sind. Diese Vor- und Nachteile sind bei der Wahl gegeneinander abzuwägen. In Abbildung 6.5 sind die Daten der Grippesaison 2015/16 zu sehen. Der ARE-Konsultationsindex ist als eine durch Punkte verbundene Linie dargestellt. Die nachgewiesenen Influenzafälle sind als rote Balken im Diagramm eingezeichnet.

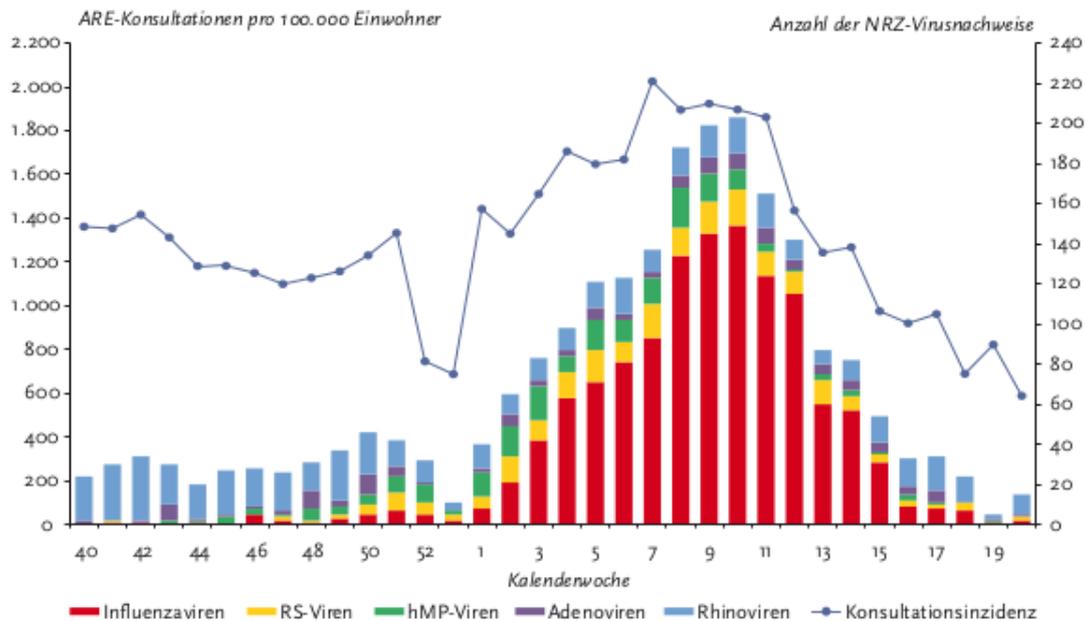


Abbildung 6.5: Grafik aus dem Saisonbericht zur Saison 2015/16 des RKI (2015). Der Konsultationsindex gibt die Anzahl der ARE-Erstkonsultationen pro Woche an. Die roten Balken repräsentieren die nachgewiesenen Influenzafälle.

Leider war es nicht möglich, Die Rohdaten, die Abbildung 6.5 zu Grunde liegen, sind vorerst nicht offen zugänglich. Sie sind im Rahmen der vorliegenden Arbeit

aber insoweit entbehrlich, als das Ziel nicht primär in der Transmissionsmodellierung, sondern vielmehr in der Erprobung der Probabilistischen Programmierung im Zusammenhang mit Systemischen Risiken liegt. Aus diesem Grund erschien es im Kontext dieser Arbeit als vertretbar, die Daten aus Abbildung 6.5 abzulesen.

## 6.4 Auszüge aus dem Probabilistischen Programm

Der vollständige Quellcode des Simulationsprogramms ist in Anhang A abgedruckt. Die Modelle dieser Arbeit wurden wie bereits erwähnt in der Probabilistischen Programmiersprache Anglican geschrieben (Tolpin et al., 2016). Es soll an dieser Stelle keine Einführung in die Probabilistische Programmierung mit Anglican gegeben werden. Trotzdem werden im Folgenden einige Ausschnitte aus dem Programm erläutert. Bei den Erläuterungen wird nicht im Detail auf den Quellcode eingegangen. Der interessierte Leser sei auf den einführenden Artikel von Tolpin et al. (2016) verwiesen. Es ist aber auch ohne Kenntnis der Sprache ist es möglich, einen Eindruck von der Modellierung mit Hilfe der Probabilistischen Programmierung zu bekommen und dieser Zweck wird hier verfolgt.

Zentral für das Simulationsmodell ist der Mechanismus, durch den neue Fälle generiert werden. Wie weiter oben diskutiert, wurde hier ein zweistufiger Mechanismus angenommen, in dem zunächst die bereits infizierten Personen weitere Personen infizieren und diese dann wieder neue Fälle generieren. Diese Anzahlen wurden als Poisson-verteilte Zufallsvariablen modelliert, die sich durch den Parameter unterscheiden. Wahrscheinlichkeitsverteilungen werden in Anglican als spezielle Datentypen realisiert, die zwei Funktionen unterstützen, `sample` und `observe`. Mit Hilfe von `sample` können Werte gezogen werden, mit Hilfe von `observe` kann die sogenannte *log-likelihood* berechnet werden, also der natürliche Logarithmus der Wahrscheinlichkeit für einen Wert aus der Verteilung. Letztere Funktion wird immer dann verwendet, wenn die bedingte Wahrscheinlichkeit in Bezug auf Daten berechnet werden soll und ist notwendig für die Anwendung von MCMC-Algorithmen. Anglican stellt Implementierungen für eine Reihe von Wahrscheinlichkeitsverteilungen bereit, zu denen auch die Poisson-Verteilung gehört.

Um eine bessere Lesbarkeit des Quellcodes zu erzielen, wurden aber im Quellcode nicht zweimal nacheinander die entsprechenden Funktionen mit den jeweiligen Para-

---

```

1 (defdist two-stage-poisson-dist
2   "Custom distribution representing new cases in the two stage poisson
3   model. Expects number of primary and secondary cases and respective
4   parameters as inputs. sample returns a sample from this distribution.
5   When given a number of observed new infections, observe computes the
6   log-likelihood of given the number of infections. Wraps around
7   fast-poisson and new-cases-dist."
8   [N-1 l-1 N-2 l-2]
9   [*l-1 (* N-1 l-1)
10  *l-2 (* N-2 l-2)
11  *lambda (+ *l-1 *l-2)]
12  (sample* [this] (sample* (new-cases-dist 1 *lambda)))
13  (observe* [this value]
14           (if (pos? *lambda)
15               (observe* (poisson *lambda) value)
16               0)))

```

---

Listing 1: Implementierung der Wahrscheinlichkeitsverteilung zur Repräsentation von Neuinfektionen.

metern aufgerufen. Stattdessen wurde eine Möglichkeit ausgenutzt, die in Anglican zur Verfügung steht: das Definieren von eigenen Wahrscheinlichkeitsverteilungen. Als Argumente werden die Anzahlen der primären und sekundären Infizierten sowie die Parameter der Poisson-Verteilungen benötigt. Der Quellcode ist in Listing 1 abgedruckt. `two-stage-poisson-dist` repräsentiert die Anzahl der Neuinfektionen in Abhängigkeit der primären und sekundären Fälle und der jeweiligen Parameter. Innerhalb der Implementierung dieser Verteilung wird natürlich von Anglicans Poisson-Verteilung Gebrauch gemacht. Schon durch die Namensgebung aber, so zumindest die Hoffnung, kann man einen Vorteil für die Lesbarkeit erhalten. Außerdem muss im Quellcode nur noch ein Aufruf der neuen Verteilung erfolgen anstelle von zwei Aufrufen der Poisson-Verteilung, was ebenfalls zur besseren Verständlichkeit beitragen kann.

Die Implementierung der Wahrscheinlichkeitsverteilung `two-stage-poisson-dist` lässt sich, wie an diesem Beispiel zu sehen ist, schon mit relativ wenig Quellcode erreichen. Die obige Implementierung kommt mit 16 Zeilen aus, von denen 6 Zeilen allein die Beschreibung der Verteilung darstellen und somit gar nicht zum eigentlichen

---

```
1 (defdist new-cases-dist
2   "Custom distribution to sample new infections from N infectious
3   individuals with parameter lambda."
4   [N lambda]
5   [*lambda (* N lambda)]
6   (sample* [this]
7     (cond
8       (zero? *lambda) 0
9       (> *lambda 30) (sample* (d/fast-poisson *lambda))
10      :else (sample* (poisson *lambda))))
11  (observe* [this value]
12    (if (pos? N)
13        (observe* (poisson *lambda) value)
14        0)))
```

---

Listing 2: Implementierung der Wahrscheinlichkeitsverteilung `new-cases-dist`.

Quelltext gehören. In Zeile 12 wird das Ziehen aus der `two-stage-poisson-dist` darüber erreicht, dass eine andere Verteilung, `new-cases-dist`, verwendet wird, aus der dann die Werte gezogen werden. Der Quellcode dieser Verteilung ist in Listing 2 zu sehen.

Bei genauer Betrachtung ist zu sehen, dass auch innerhalb der `new-cases-dist` eine andere Verteilung benutzt wird, aus der dann die Werte gezogen werden. Genauer wird entweder eine Verteilung mit Namen `fast-poisson` (Zeile 9) oder aber die von Anglican bereit gestellte `poisson`-Verteilung verwendet (Zeile 10). Dieser Umweg wurde gegangen, weil sich im Laufe der Entwicklung des Modells herausgestellt hat, dass die Laufzeit des Programms bei der Verwendung der `poisson`-Verteilung, die von Anglican bereit gestellt wird, mit größer werdendem Parameter ebenfalls stark ansteigt. Da der Parameter sich unter anderem aus der Anzahl der infizierten Personen berechnet und diese Zahl im Laufe der Simulation bis zu fünf Stellen erreicht, kam es zu langen Laufzeiten. Durch eine alternative Implementierung der `sample`-Funktion konnte eine deutliche Verbesserung erzielt werden. Ein entsprechender Algorithmus wurde von Atkinson (1979) beschrieben. Es wurde deshalb eine weitere Verteilung namens `fast-poisson` definiert, die diesen Algorithmus verwendet. Der Quellcode dieser Funktion ist in Listing 6.4 zu sehen.

---

```

1 (defn sample-fast-poisson [lambda]
2   (let
3     [;; helper functions
4      beta-f (fn [lambda]
5              (/
6               Math/PI
7               (sqrt (* 3 lambda))))
8      alpha-f (fn [lambda b]
9              (* lambda b))
10
11     ;; constants
12     b (beta-f lambda)
13     a (alpha-f lambda b)
14     c (- 0.767 (/ 3.36 lambda))
15     k (- (- (log c) lambda) (log b))
16
17     ;; functions for rejection sampling
18     lhs (fn [x]
19          (let
20            [v (sample* (uniform-continuous 0 1))
21             y (- a (* b x))]
22             (+
23              y
24              (log
25               (/
26                v
27                (pow (+ 1 (exp y)) 2))))))
28
29     rhs (fn [n]
30          (-
31           (+ k (* n (log lambda)))
32           (log-gamma-fn (inc n))))
33
34     ;; rejection sampling
35     (loop [x (propose lambda b a)]
36
37           (if (<= (lhs x) (rhs (floor (+ x 0.5))))
38             (int (floor (+ x 0.5)))
39
40             (recur (propose lambda b a))))))

```

---

Listing 3: Alternative Implementierung der Funktion zum sampeln aus der Poisson-Verteilung. Die Funktion ruft unter anderem eine Funktion namens `propose` auf, deren Implementierung im Anhang A zu finden ist.

Diese Ausführungen relativieren natürlich die frühere Aussage zur relativ knappen Formalisierung einer Wahrscheinlichkeitsverteilung. An diesem Beispiel wird auch deutlich, dass in der Praxis der Entwicklung eines Simulationsprogramms unvermutete Probleme auftreten können, die zudem mit dem eigentlichen Gegenstand des Modells nichts zu tun haben. Sie stellen aber Hemmnisse dar, zu deren Beseitigung zunächst geeignete Lösungen gefunden werden müssen. In diesem Fall war die Verringerung der Laufzeit eine solche Lösung. Auf der anderen Seite ist positiv hervorzuheben, dass auch die Behandlung solcher Probleme möglich ist, die für den praktischen Nutzen eine große Rolle spielen.

Zuletzt soll noch die Implementierung der Progression einer Kohorte gezeigt werden, die in Kapitel 6.1 beschrieben wurde. Für jede Kohorte wird eine Funktion namens `progress` aufgerufen, die in Listing 6.4 zu sehen ist. Als Argumente erwartet diese Funktion den Zeitpunkt `t` der Entstehung der Kohorte, die Anzahl der Individuen in der Kohorte, hier als `cases` bezeichnet, und eine Datenstruktur `coll`, die die Kompartimente enthält. In Zeile 10 wird die Anzahl derer bestimmt, die die Kohorte zu diesem Zeitpunkt wieder verlassen, weil sie entweder gesund oder an der Krankheit gestorben sind. Das sind die Individuen, die vom I- ins R-Kompartiment wechseln. Die übrigen Individuen verbleiben in der Kohorte und werden in Zeile 11 als `remaining` bezeichnet. In Zeile 12 wird die Anzahl der Individuen im S-Kompartiment des vorhergehenden Zeitschritts bestimmt, die später für den aktuellen Zeitpunkt übernommen wird.

## 6.5 Ergebnisse

### 6.5.1 Simulation ohne Einbindung von Daten

Die Ergebnisse von 10000 Simulationsläufen des probabilistischen SIR-Modells einer Population von 100000 Individuen über einen Zeitraum von 40 Wochen sind in Abbildung 6.6 zu sehen. Da bei diesen Programmläufen keine empirischen Daten berücksichtigt wurden, handelt es sich um die A-priori Verteilung des Modells, also die Verteilung, die allein auf den Annahmen beruht, die bei der Formulierung des Modells gemacht wurden. Auf der  $x$ -Achse sind die Zeitpunkte abgetragen, auf der  $y$ -Achse für jede Woche und für jeden Simulationslauf die jeweils vorhergesagte An-

---

```

1 (defm progress
2   [t cases coll]
3
4   (if (= t (count coll))
5     ;; if time's up, remaining cases
6     ;; are discarded
7     coll
8
9     (let
10      [removed (sample (binomial cases 0.45))
11       remaining (- cases removed)
12       susceptible (get-in coll [(dec t) :S])
13
14       ;; during progression a number of things need to happen
15       ;; at each timestep some individuals recover, their number
16       ;; has to be added to [t :R] the rest remains infected,
17       ;; this number has to be added to [t :I] the number of
18       ;; susceptibles has to be copied from (dec t) to t
19       updated-coll (assoc-in
20                    (update-in-ang
21                     (update-in-ang coll [t :R] + removed)
22                     [t :I] + remaining)
23                    [t :S] susceptible)])
24
25     (progress (inc t) remaining updated-coll))))

```

---

Listing 4: Quellcode zur Repräsentation der Progression einer Kohorte. Diese Funktionsdefinition beginnt anders als in Listing 6.4 nicht mit `defn`, sondern mit `defm`. Grund ist, dass diese Funktion eine “echte” Anglican Funktion ist. Im Gegensatz dazu ist die Funktion aus Listing 6.4 eine Clojure Funktion. Da Anglican in Clojure geschrieben ist, ist diese Form der Interoperabilität möglich.

zahl an Neuinfektionen in dieser Woche. Die blau unterlegte Fläche kennzeichnet das 95%ige *high density interval (HDI)*, dass die 95% der wahrscheinlichsten Werte umfasst. Die Grenzen werden so festgelegt, dass die Werte innerhalb des HDI eine größere Wahrscheinlichkeit haben als die Werte außerhalb (Kruschke, 2014).

Die Anzahl der Neuinfektionen pro Woche ist die Summe aus der Anzahl der primären und sekundären Neuinfektionen, wie in Kapitel 6.2.1 dargelegt wurde. Die

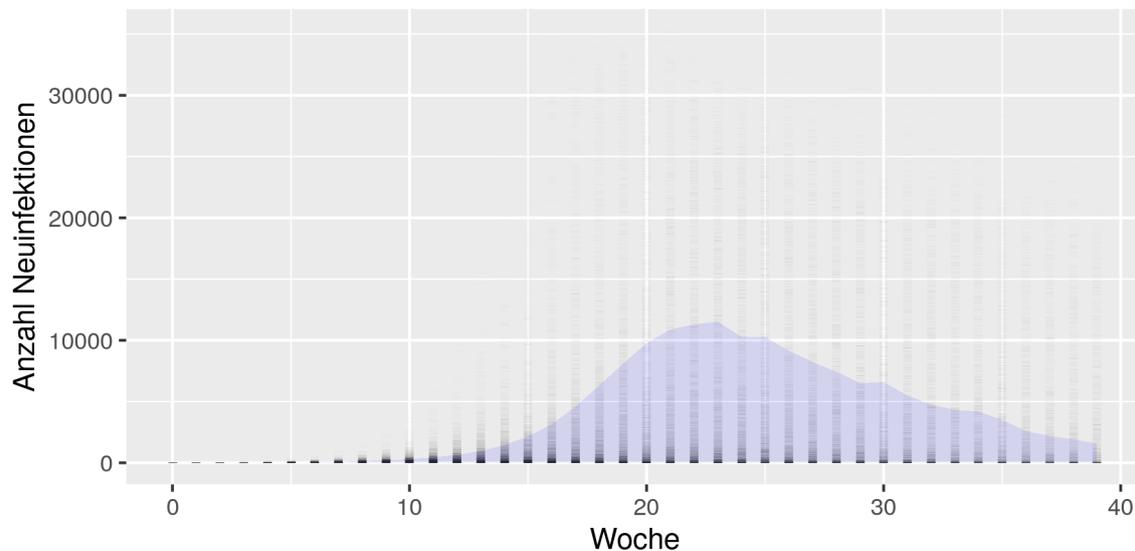


Abbildung 6.6: Ergebnisse von 10000 Simulationen einer Grippe-Saison über 40 Wochen mit einer Population von 100000 Individuen ohne Einbindung von Daten. Pro Woche sind die jeweils vorhergesagten Anzahlen an Neuerkrankungen abgetragen. Die blaue Fläche markiert das 95%ige *high density interval (HDI)*. Alle Werte außerhalb dieses Intervalls sind weniger wahrscheinlich als die Werte innerhalb des HDI.

Ergebnisse für diese Zufallsvariablen zeigt Abbildung 6.7.

Was in Abbildung 6.6 auffällt, ist die große Varianz der Ergebnisse, auf die in Kapitel 6.6 noch näher eingegangen wird. Ein besseres Bild über die Verteilungen der vorhergesagten Neuinfektionen der einzelnen Wochen bekommt man in Abbildung 6.8. In dieser Abbildung fällt noch ein zweiter Punkt auf. Es gibt eine große Anzahl an Simulationsläufen, in denen über die gesamte Dauer der Simulation überhaupt keine Infektionen auftreten. Darauf wird ebenfalls in Kapitel 6.6 näher eingegangen. Um die Wertebereiche besser sehen zu können, wurden diese Programmläufe in Abbildung 6.9 nicht berücksichtigt. Des Weiteren wurde die Skalierung der  $x$ -Achsen jeweils an die Daten angepasst.

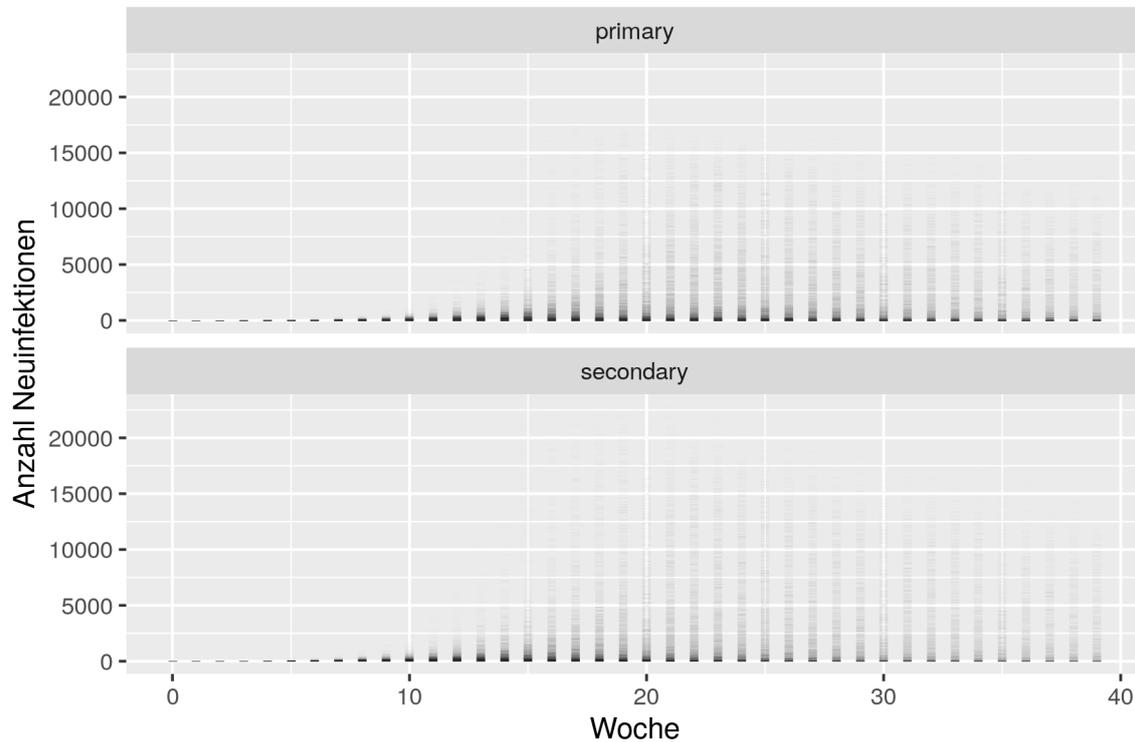


Abbildung 6.7: Ergebnisse für primäre und sekundäre Neuinfektionen pro Woche über 40 Wochen ohne Einbindung von Daten.

### 6.5.2 Simulation mit Einbindung von Daten

In einer zweiten Variante wurden Daten des RKI in die Simulation eingebunden (vgl. Kapitel 6.3), um die A-posteriori Verteilung des Modells gegeben die Daten zu erhalten. Aus dem Bericht zur Grippezeit 2015/16 wurden die Zahlen der nachgewiesenen Influenzafälle der ersten 23 Wochen entnommen und die Ausführung der Programmläufe wurde auf diese Daten bedingt. Wie schon in Kapitel 6.3 erläutert, mussten die Daten aus Abbildung 6.5 abgelesen werden und stellen deshalb eine grobe Schätzung zu betrachten. Von besonderem Interesse ist aber vor allem, zu sehen, wie sich die Ergebnisse durch das Bedingen auf Daten verändern.

Abbildung 6.10 zeigt die Ergebnisse von 10000 Programmläufen unter Einbezug der Daten. Die Datenpunkte sind als rote Striche eingezeichnet. Man sieht, dass das Bedingen auf Daten einen starken Einfluss auf die Simulation hat. Die Zahl

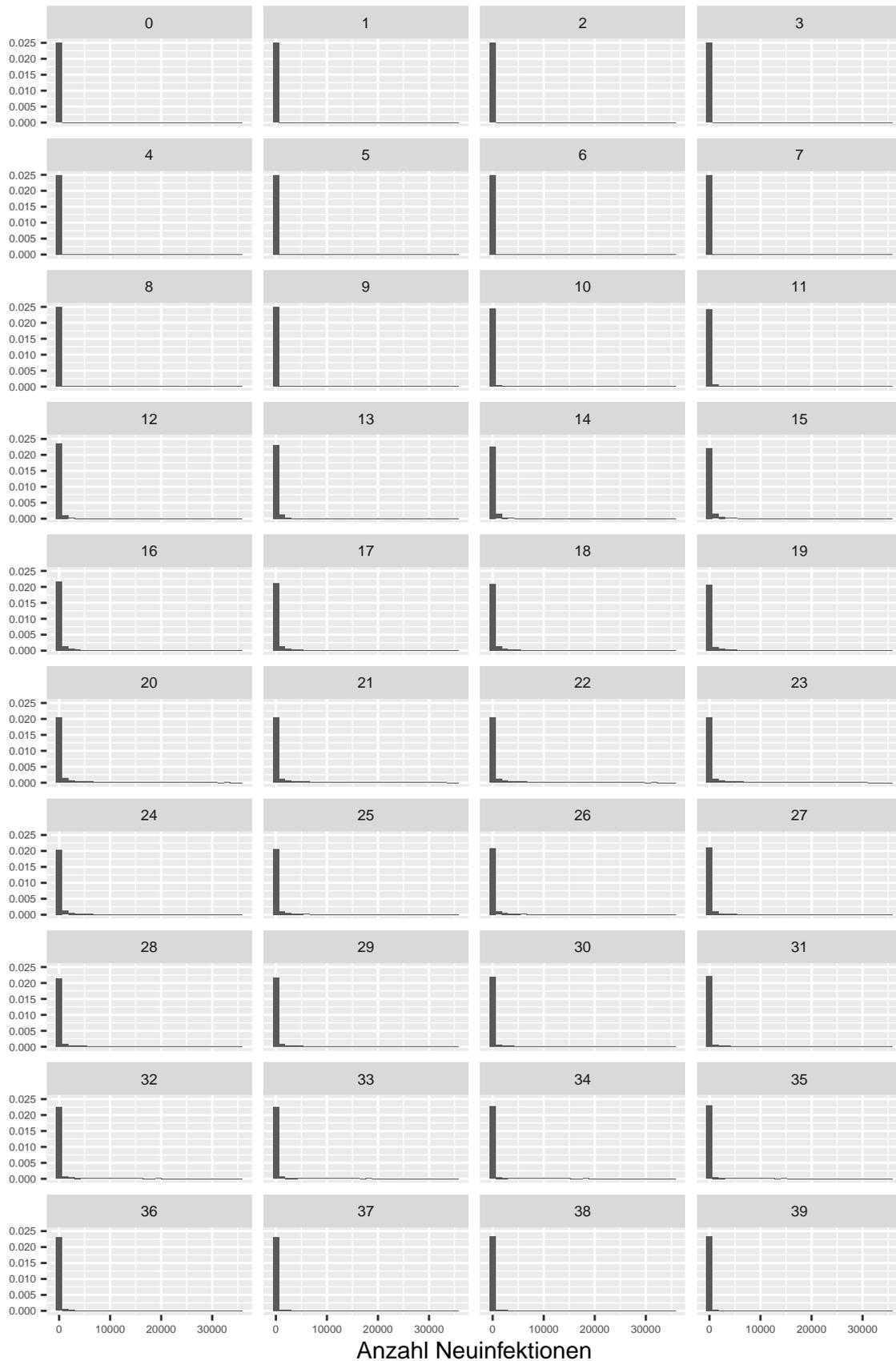


Abbildung 6.8: Simulation ohne Einbindung von Daten: Histogramme der Verteilungen der vorhergesagten Anzahl an Neuinfektionen nach Wochen. Identische Skalierung der  $x$ -Achsen.



Abbildung 6.9: Simulation ohne Einbindung von Daten: Histogramme der Verteilungen der vorhergesagten Anzahl an Neuinfektionen nach Wochen. Datenpunkte mit einer Anzahl von 0 Neuinfektionen wurden entfernt. Skalierung der  $x$ -Achsen wurde an die Daten angepasst.

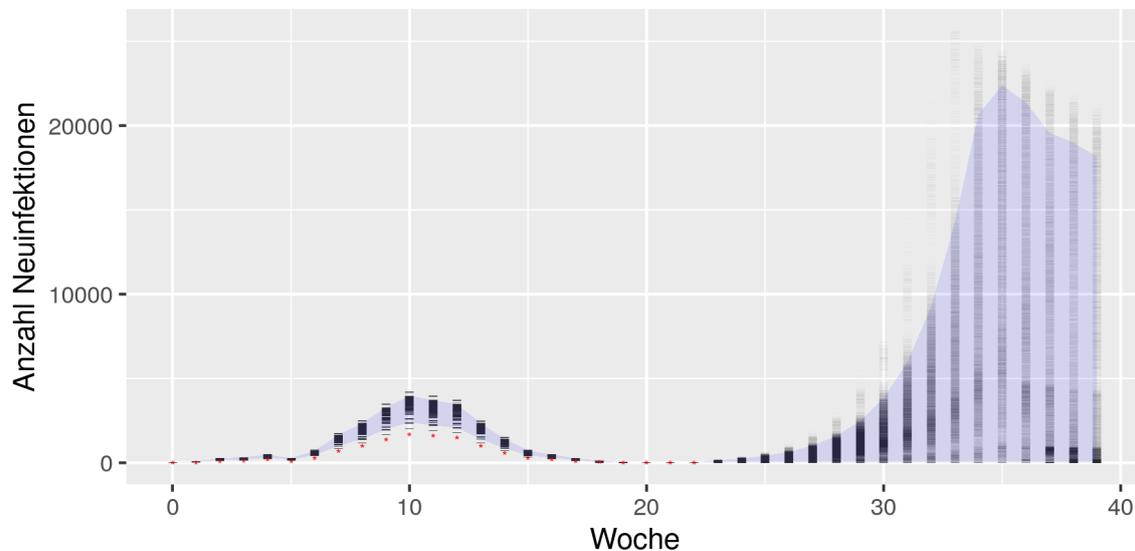


Abbildung 6.10: Ergebnisse von 10000 Programmläufen einer Grippezeit über 40 Wochen mit einer Population von 100000 Individuen unter Einbezug von Daten für die ersten 23 Wochen. Datenpunkte sind in rot eingezeichnet. Pro Woche sind die jeweils vorhergesagten Anzahlen an Neuerkrankungen abgetragen. Die blaue Fläche zeigt das HDI.

der Influenzafälle wird meistens etwas überschätzt. Die Varianz ist aber gegenüber der ersten Variante stark verringert. Man sieht auch, dass die Varianz sich erhöht, wenn keine Daten mehr vorhanden sind. Die ersten zwei bis drei Wochen, nachdem keine Daten mehr vorliegen, bewegt sich die Vorhersage noch in einem relativ engen Bereich. Je länger die Zeit in der Simulation fortschreitet, desto größer werden die Grenzen, in denen die vorhergesagten Werte liegen. Abbildung 6.11 zeigt die Anzahl der Neuinfektionen für primäre und sekundäre Fälle.

Die A-posteriori Verteilungen der einzelnen Wochen sind in Abbildung 6.12 zu sehen. Im Gegensatz zu Abbildung 6.8 variiert die Skalierung der  $x$ -Achse, um eine bessere Darstellung des Wertebereichs zu erreichen. Auch in dieser Abbildung zeigt sich, was passiert, wenn keine Daten mehr vorhanden sind. Etwas detaillierter als in Abbildung 6.10 ist hier zu sehen, wie sich das Simulationsmodell ab Woche 23 verhält, wenn also keine Daten mehr vorhanden sind. Die Verteilungen der Wochen 23 bis 27 haben einen deutlichen Gipfel und fallen zu den Seiten hin gleichmäßig

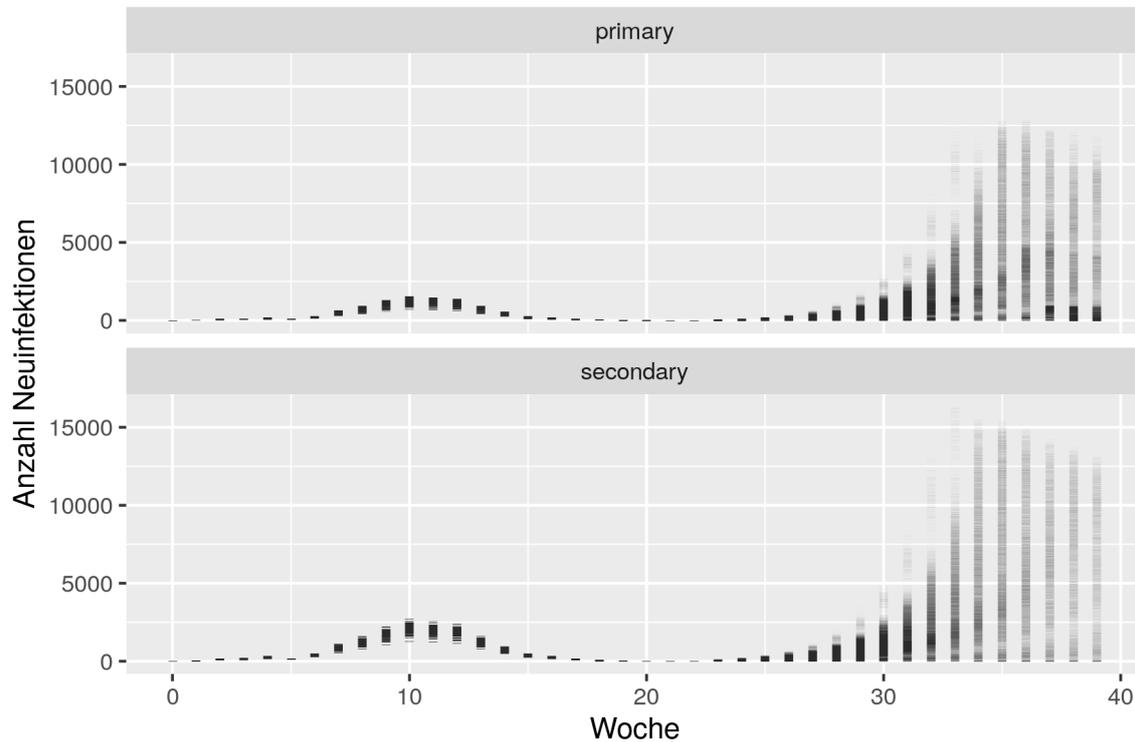


Abbildung 6.11: Ergebnisse für primäre und sekundäre Neuinfektionen der Simulation pro Woche über 40 Wochen unter Einbezug von Daten.

ab. Die Varianz wird dabei von Woche zu Woche größer. Ab Woche 28 betragen die Abweichungen vom Mittelwert dann nicht mehr mehrere hundert, sondern an die tausend Fälle. Über die letzten Wochen wird die vorhergesagte Verteilung immer flacher und umfasst dabei einen immer größer werdenden Wertebereich. In den letzten drei Wochen der Simulation zeigt sich dann aber wieder eine Verschiebung hin zu niedrigeren Werten.

Interessant sind auch die A-posteriori Verteilungen der Parameter  $\lambda_1$  und  $\lambda_2$ . Bedingt auf die Daten verändern sich die Verteilungen gegenüber der A-priori Vermutung so, wie in Abbildung 6.13 zu sehen ist. Ausgehend von einer Gleichverteilung zwischen 0.4 und 0.9 bzw. 1.0 und 1.8 (vgl. Abbildung 6.4) verschieben sich beide Verteilungen nach rechts hin.

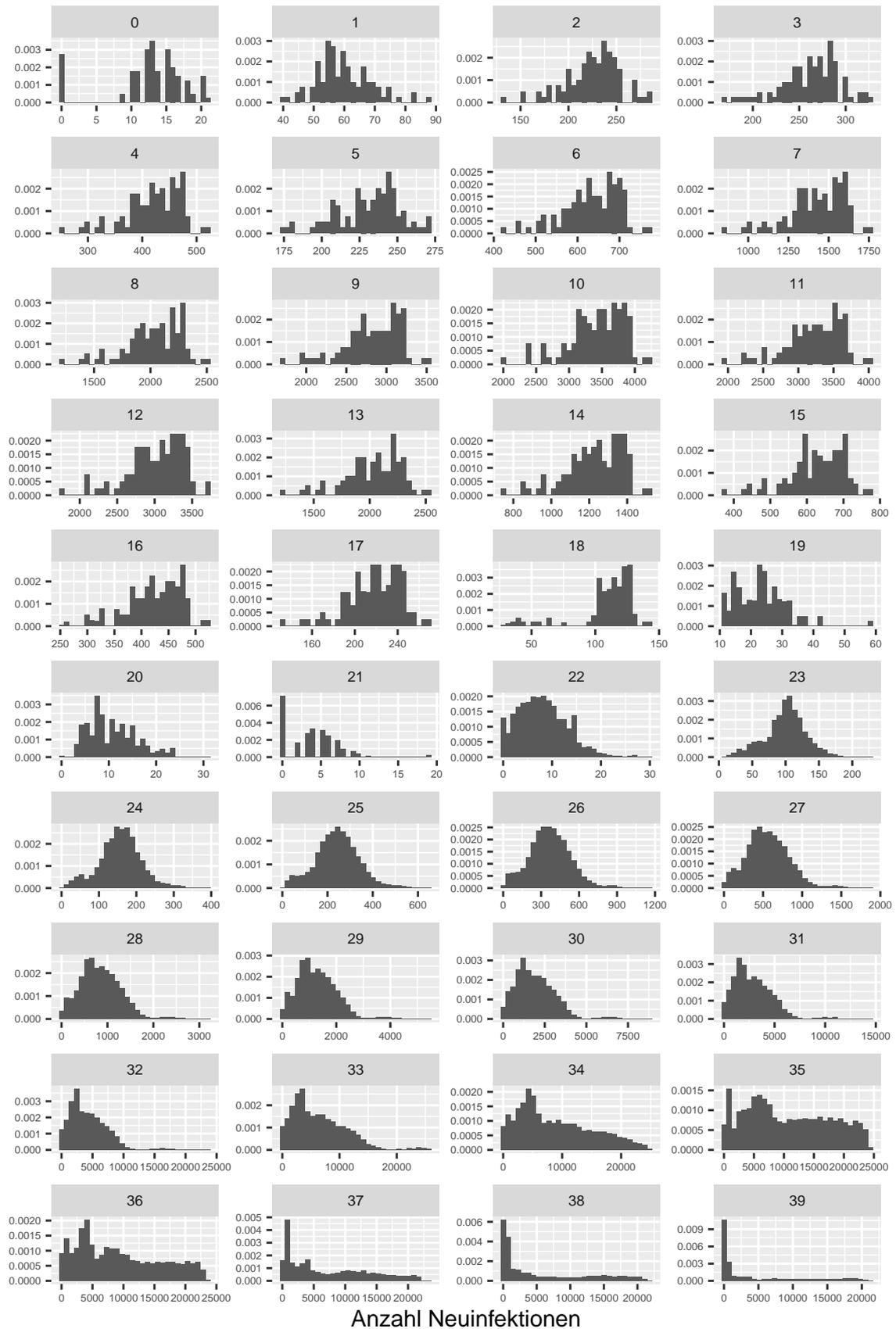


Abbildung 6.12: Simulation unter Einbezug von Daten: Histogramme der Verteilungen der vorhergesagten Anzahl an Neuinfektionen pro Woche. Skalierung der Achsen wurde angepasst, so dass die Wertebereiche besser zu erkennen sind.

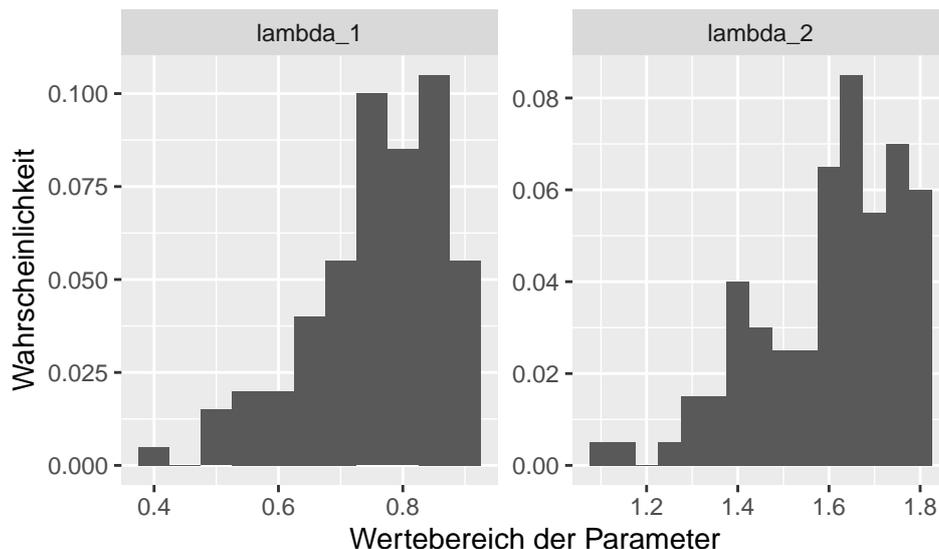


Abbildung 6.13: A-posteriori Verteilungen der primären und sekundären Ansteckungsparameter  $\lambda_1$  und  $\lambda_2$ .

## 6.6 Diskussion

Betrachtet man zunächst die A-priori Verteilung des Modells, die in Abbildung 6.6 zu sehen ist, dann sieht man, dass die Anzahl der Neuinfektionen zunächst ansteigt und zum Ende der Periode hin wieder abfällt. Damit wird die grundsätzliche Charakteristik einer Grippezeit wiederzugeben. Die Bandbreite, innerhalb derer sich die Vorhersage bewegt, steigt allerdings mit der Dauer der Simulation an. Während die Werte in den ersten Wochen noch eine vergleichsweise geringe Varianz haben, umfassen sie in der Mitte der Saison Werte zwischen Null und 30000. Bezogen auf eine Population von 100000 Individuen ist eine solche Bandbreite unrealistisch. Man kann deshalb sagen, dass das Modell mit einer großen Unsicherheit behaftet ist.

Der Grund für die hohe Varianz liegt auch in der Wahl der Poisson-Verteilung. Der Parameter dieser Verteilung gibt, wie in Kapitel 6.2.1 erläutert wurde, sowohl ihren Erwartungswert als auch ihre Varianz an. Da der Parameter im Modell proportional zur Anzahl der infizierten Individuen ist und diese Anzahl vier- und fünfstellige Werte annehmen kann, können sowohl Erwartungswert als auch Varianz ebenfalls groß werden. Genau das kann man hier beobachten. Zukünftige Forschung kann in diesem Punkt ansetzen und alternative Wahrscheinlichkeitsverteilungen identifizie-

ren, die ähnlich plausibel sind, aber eine geringere Varianz aufweisen. Die besondere Herausforderung dabei liegt darin, dass es für die Verwendung in einem Probabilistischen Programm möglich sein muss, die sogenannte *likelihood function* der Summe von vielen solcher Zufallsvariablen zu berechnen. Dabei handelt es sich um die Wahrscheinlichkeits- bzw. Dichtefunktion einer Wahrscheinlichkeitsverteilung, mit deren Hilfe man die Wahrscheinlichkeit eines Wertes berechnen kann. Dieser Punkt muss erfüllt sein, weil das Modell auf der Basis einzelner Individuen spezifiziert ist, die Daten aber nur aggregiert für eine gesamte Woche vorliegen (vgl. dazu auch die Ausführungen aus Kapitel 6.2.1). Das Bedingen auf Daten ist deshalb anders nicht möglich.

Es fällt außerdem auf, dass es für jede Woche der Simulation eine große Anzahl an Programmläufen gab, in denen keine Neuerkrankungen auftraten. Auch in diesem Punkt ist das Modell unrealistisch. Der Grund liegt darin, dass Neuinfektionen nur durch den Kontakt mit einem bereits infizierten Individuum zustande kommen können. Ist die Zahl der Neuinfektionen einmal auf Null gesunken, dann sind innerhalb dieses Modells keine neuen Infektionen zu einem späteren Zeitpunkt möglich. Auch an diesem Punkt kann zukünftige Forschung ansetzen und Lösungen finden, wie das Modell verbessert werden kann. Um die Idee der generativen Modellierung konsequent weiter zu verfolgen, wäre eine Theorie-geleitete Lösung wünschenswert. Der modulare Aufbau des Probabilistischen Programms macht das spätere Hinzufügen einer solchen Weiterentwicklung möglich.

Die A-posteriori Verteilung des Modells gegeben die Daten der Saison 2015/16 kommt dem tatsächlichen Verlauf der Saison einen großen Schritt näher. Durch das Bedingen auf die Daten wird die Varianz deutlich kleiner. Nimmt man die Daten wieder weg, dann wird diese Varianz nach und nach größer und ist nach ungefähr zehn Wochen wieder in einem ähnlichen Bereich wie auch in der ersten Variante. In den ersten zwei bis drei Wochen ist die Bandbreite aber noch relativ gering. Das Modell könnte deshalb für die kurzfristige Prognose interessant sein.

Der Aspekt, an dem die Herkunft der Probabilistischen Programmierung aus der künstlichen Intelligenzforschung Vielleicht am deutlichsten wird, betrifft die A-posteriori Verteilungen der Parameter. Diese zeigen nämlich deutliche Gipfel, wo die A-priori Vermutung eine flache Verteilung war. Man kann durchaus davon sprechen, dass diese Parameter durch die Daten gelernt wurden. Es wäre nun möglich, das

Modell mit diesen gelernten Parametern aufs Neue laufen zu lassen. Genauso könnte man auf einen anderen Datensatz bedingen und fragen, ob die Parameter auch unter anderen Bedingungen zu guten Ergebnissen führen. An dieser Stelle sei auch angemerkt, dass ein sehr kleiner Datensatz benutzt wurde. Lediglich 23 Datenpunkte, einer pro Woche für die ersten 23 Wochen der Simulation, wurden eingesetzt. Ein großes Potential für die Schätzung von Parametern besteht sicherlich in der Nutzung eines größeren Datensatzes. Grundsätzlich sind mehr Daten in Bezug auf die Influenza vorhanden. Das RKI, und dort speziell die *Arbeitsgemeinschaft Influenza (AGI)*, ist international ein Vorreiter auf dem Gebiet des Gesundheitsmonitoring. Durch die Zusammenarbeit mit der AGI kann die Datenbasis verbreitert und damit die Aussagefähigkeit des Modells vergrößert werden. Dies dürfte in beiderseitigem Interesse sein und sollte daher angestrebt werden.

Die Daten des RKI werden nach Bundesländern differenziert. Eine weitere Möglichkeit zur Verbesserung des Modells liegt darin, ebenfalls eine solche Differenzierung zu machen und ein hierarchisches Modell zu formulieren. Die Schätzung der übergeordneten Parameter für die Gesamtbevölkerung würde dabei von den Parametern der unteren Ebene, in dem Fall die Bundesländer, abhängen. In diesem Zusammenhang könnte auch die Modellierung Impfquote detaillierter angegangen werden. Das Modell berücksichtigt zwar, dass es in Deutschland die freiwillige Gripeschutzimpfung gibt und nimmt eine gewisse Anzahl von Individuen als geimpft an. Die Daten zum Impfstatus bei der Rotavirus- und Masernimpfung zeigen aber, dass es regional zum Teil deutliche Unterschiede geben kann (Rieck et al., 2017). Diese Erkenntnisse basieren auf der Auswertung von Abrechnungsdaten der Krankenversicherungen, die dem RKI zur Verfügung gestellt wurden. Wie dem Autor in einem persönlich Gespräch mitgeteilt wurde, ist eine solche Analyse auch für die Influenzaimpfung geplant. Das würde die Möglichkeit eröffnen, auch zur Schätzung der Impfquoten Daten heranzuziehen.

# Kapitel 7

## Fallbeispiel 2: Soziale Diffusion und Fake News

Das zweite Fallbeispiel, an dem die Probabilistische Programmierung für Systemische Risiken ausprobiert wird, steht im Zusammenhang mit der Idee der sozialen Diffusion. Soziale Diffusion ist ein Thema, das immer wieder in sozialwissenschaftlich orientierter Forschung aufgegriffen wird (Jäger, 2015; Karlova & Fisher, 2013; Byers, Mitzenmacher & Zervas, 2012; Bonabeau, 2002). Nach Bonabeau (2002) geht es im Kern darum, dass einzelne Akteure von ihrem sozialen Kontext, also etwa von Freunden und Bekannten, beeinflusst werden. Dabei ist es für das Konzept der sozialen Diffusion zweitrangig, auf was sich diese Beeinflussung bezieht. Wichtig ist vielmehr der Einflussmechanismus. Über die sozialen Kontakte wird etwas weitergegeben, was dann durch das soziale Gefüge diffundiert. In sozialwissenschaftlicher Forschung kann es sich um immaterielle Dinge wie Meinungen oder Informationen handeln (Karlova & Fisher, 2013). Aber auch die Ausbreitung von Elektroautos als Manifestation einer umweltfreundlichen Einstellung wurde auf diese Weise modelliert (Jäger, 2015).

Es erstaunt deshalb nicht, dass soziale Diffusion häufig in Zusammenhang mit sozialen Netzwerken betrachtet wird. Das in diesem Kapitel vorgestellte Modell wird ebenfalls von einem Netzwerk Gebrauch machen. Dabei wird manchmal von Netzwerken und manchmal von Graphen die Rede sein. Streng genommen handelt es sich dabei um ganz verschiedene Dinge. Ein Graph ist ein mathematisches Konstrukt,

dass mit Hilfe von verschiedenen Angaben als wohldefiniert gelten kann. Die Definition eines sozialen Netzwerks ist dagegen nicht so klar. Graphen werden aber häufig dazu benutzt, soziale Netzwerke zu repräsentieren und eignen sich dazu auch in besonderer Weise. Im Rahmen der vorliegenden Arbeit werden diese Bezeichnungen deshalb, sofern nichts anderes gesagt wird, synonym verwendet.

Um ein aktuelles Thema aufzunehmen, wurde ein Probabilistisches Programm zur Diffusion von Fake News bzw., frei übersetzt, von Falschmeldungen entwickelt. Dazu wurde ein Modell von Jäger (2015) adaptiert, in dem die Ausbreitung einer umweltfreundlichen Einstellung thematisiert wird, die sich im Kauf von Elektroautos zeigt. In dem im Folgenden vorgestellten Modell sind es nicht Elektroautos, sondern die Falschmeldungen, die durch das soziale Netzwerk diffundieren. Im Vergleich zum Probabilistischen Programm, das in Kapitel 6 vorgestellt wurde, sind die Prozesse, die dabei ablaufen, rudimentär. Interessant ist das Modell vor allem deshalb, weil die Prozesse auf einem Graphen ablaufen. Der Aspekt der sozialen Vernetzung, der für sozialwissenschaftliche Forschung grundsätzlich interessant ist, wurde in Kapitel 6 nicht berücksichtigt.

Auch dieses Beispiel kann mit Blick auf das Rahmenmodell im Zusammenhang mit Systemischen Risiken diskutiert werden. Schon die Idee der sozialen Diffusion an sich passt sehr gut zu dem Rahmenmodell, da die Rekursivität im Diffusionsmechanismus gewissermaßen enthalten ist. Tatsächlich gibt es auch Ähnlichkeiten mit der Ausbreitung einer Krankheit, wie sie in Kapitel 6 thematisiert wurde. Im Unterschied dazu wird aber in diesem Modell ein Graph verwendet, dessen Knoten die Fake News weiter tragen. Es wäre theoretisch möglich, ein Modell zur Ausbreitung von Influenza auch ausgehend von einem Graphen zu entwickeln. Da aber die Struktur des Graphen, der die Bevölkerung Deutschlands repräsentiert, nicht bekannt ist, wurde dort ein anderer Ansatz gewählt. Je mehr Knoten die Fake News übernehmen, desto mehr Knoten verbreiten diese dann auch im weiteren Verlauf. Die Anzahl ist demnach rekursiv von sich selbst abhängig. Das System  $S$  ist in diesem Fall der Graph bzw. das soziale Netzwerk, das als Ausgangspunkt des Modells gewählt wird. Der Aspekt der Globalität kommt darin zum Ausdruck, dass das gesamte Netzwerk betroffen sein kann. Dieses Netzwerk ist aber virtuell, es bildet nicht direkt ein tatsächliches soziales Netzwerk nach. In diesem Punkt wird deutlich, dass es sich hier, wie Gottschalk-Mazous sagen würde, um ein toy model handelt.

Der Quellcode für dieses Simulationsmodell ist in Anhang B zu sehen.

## 7.1 Beschreibung des Modells

Ausgangspunkt des Modells ist ein soziales Netzwerk, das in Form eines Graphen modelliert wird. Auf die genaue Topologie des Graphen wird weiter unten eingegangen. An dieser Stelle sei zunächst gesagt, dass die Knoten des Graphen die sozialen Akteure des Netzwerkes repräsentieren. Sind zwei Knoten durch eine Kante verbunden, so kennen sich die entsprechenden sozialen Akteure und können miteinander kommunizieren.

Initial wird nun zufällig eine gewisse Anzahl an Knoten ausgesucht, die einer Falschmeldung “aufgesessen” sind. Das Modell geht von einer einzelnen Falschmeldung aus, es wäre aber auch denkbar, sich hier eine Menge an Falschmeldungen vorzustellen. Falls die Annahme plausibel ist, dass die Ausbreitung der Falschmeldungen aus dieser Menge dem gleichen Prozess folgt, dann wäre ein solches Vorgehen vertretbar. Man könnte dann mit dem hier vorgestellten Modell allerdings nur die Verbreitung der Gesamtheit dieser Menge an Falschmeldungen untersuchen, eine Differenzierung wäre nicht ohne Modifikation möglich.

Auch dieses Modell beschreibt einen Prozess, der in diskreten Zeitschritten abläuft. Es wird nicht näher spezifiziert, wie groß ein Zeitschritt ist. Im Gegensatz zum Modell aus Kapitel 6, wo die Wahl der Zeitschritte mit Blick auf die empirischen Daten gewählt war, ist die Größe der Zeitschritte hier prinzipiell frei wählbar. Da jeder soziale Akteur des Netzwerkes aber in jedem Zeitschritt mit seinen Bekannten kommuniziert, wie gleich noch erklärt wird, erscheint eine Dauer von einem Tag oder auch einer Woche als sinnvoll.

In jedem Schritt der Simulation kommuniziert nun ein sozialer Akteur mit seinen Bekannten, die durch seine direkten Nachbarn im Graphen repräsentiert sind. Hat der Akteur selbst bereits die Falschmeldung als wahr angenommen, so wird er im Gespräch mit seinen Bekannten diese kommunizieren und seine Bekannten auf diese Weise beeinflussen. Auf der anderen Seite werden aber auch diejenigen unter seinen Bekannten, die ihrerseits die Fake News übernommen haben, ihm diese im Gespräch unterbreiten.

Ein sozialer Akteur wird die Falschmeldung deshalb nicht sofort übernehmen.

Diese Entscheidung wird stattdessen als Zufallsvariable modelliert, die von der Anzahl seiner Bekannten abhängt, die die Falschmeldung bereits übernommen haben. Bei der Wahl der Wahrscheinlichkeitsverteilung, der diese Zufallsvariable folgt, waren folgende Überlegungen wichtig. Zum ersten sollte die Wahrscheinlichkeit, die Falschmeldung zu übernehmen, proportional zur Anzahl der Bekannten sein, die sie bereits übernommen haben. Aber auch in dem Fall, dass alle Bekannten die Falschmeldung bereits übernommen haben, sollte der Akteur nicht sie nicht automatisch übernehmen. Schließlich hat, wie man umgangssprachlich sagt, "jeder seinen eigenen Kopf". Und auch wenn im Freundeskreis eine bestimmte Meinung vorherrscht, ein Einzelner kann davon abweichen. Und schließlich muss auch das andere Extrem bedacht werden. Selbst wenn also im Freundeskreis einer Person niemand die Falschmeldung glaubt, kann es passieren, dass ein Akteur sie übernimmt, weil er sie aus einer anderen Quelle wie etwa den Medien bezieht. Denn auch wenn der Freundes- bzw. Bekanntenkreis ein wichtiger Faktor bei der Meinungsbildung ist, es gibt andere Einflussgrößen. Da das Modell diese nicht explizit benennt, müssen sie zumindest implizit berücksichtigt werden.

Sei  $X_{A_i,n,m}$  die Zufallsvariable, die die Entscheidung des Akteurs  $A_i$  in Abhängigkeit seiner  $n$  Bekannten, von denen  $m$  bereits die Falschmeldung übernommen haben. Die Definition von  $X_{A_i,n,m}$  lautet dann:

$$X_{A_i,n,m} \sim \begin{cases} \text{Bernoulli}(p) & \text{falls } m = 0 \text{ mit } p \sim \text{Uniform}(0.01, 0.1) \\ \text{Bernoulli}(0.1 \frac{m}{n}) & \text{sonst.} \end{cases}$$

Mit dieser Definition wird  $X_{A_i,n,m}$  als Bernoulli-verteilte Zufallsvariable modelliert. Eine solche Zufallsvariable kann zwei Werte annehmen, die häufig mit 0 und 1 oder mit *Erfolg* und *Misserfolg* bezeichnet werden. Der Parameter der Bernoulli-Verteilung gibt an, mit welcher Wahrscheinlichkeit die 1 angenommen wird bzw. mit welcher Wahrscheinlichkeit ein Erfolg eintritt. Wenn keiner der Bekannten des Akteurs die Falschmeldung verbreitet, dann wird der Parameter selbst als gleichverteilte Zufallsvariable angenommen, deren Wert zwischen 0.01 und 0.1, also zwischen 1% und 10% liegt. Andernfalls berechnet sich der Parameter aus dem Anteil der Bekannten, die die Falschmeldung übernommen haben. Damit auch im Fall, dass alle Nachbarn die Falschmeldung verbreiten, diese nicht mit Sicherheit

übernommen wird, wird der Anteil mit 0.1 multipliziert. Auf diese Weise wird die Wahrscheinlichkeit maximal als 0.1 angenommen.

Die Verbreitung der Fake News  $Y_t$  innerhalb des Netzwerks zu einem gegebenen Zeitpunkt  $t$  ist dann die Anzahl derjenigen Akteure, die sie übernommen haben.  $Y_t$  ist dann die Summe der oben definierten Zufallsvariablen über alle  $N$  Akteure des Netzwerks:

$$Y_t = \sum_{i=1}^N X_{A_i,n,m}$$

## 7.2 Ergebnisse

Es liegt nahe, dass die Topologie des Graphen, also die Art der Vernetzung der Knoten untereinander, einen Einfluss auf die Verbreitung der Falschmeldung hat. Aus diesem Grund wird die Simulation des Modells auf zwei unterschiedlichen Graphen durchgeführt: auf einem Graphen nach Newman und Watts (1999) und auf einem Graphen nach Barabási und Albert (1999). Die Simulationen werden im Folgenden als *Sim-1* bzw. *Sim-2* bezeichnet. Die Unterschiede in Bezug auf die Eigenschaften der Graphen wurden bereits in Kapitel 3.3 diskutiert. Bei der nun folgenden Beschreibung der Ergebnisse wird zusätzlich auf die Unterschiede in der Konstruktion dieser Graphen eingegangen.

### 7.2.1 Simulation auf Basis eines Newman-Watts-Graphen

#### Konstruktion eines Newman-Watts-Graphen

Abbildung 7.1 zeigt einen Graphen nach Newman und Watts (1999) mit 500 Knoten. Auch wenn graphische Darstellungen von Graphen nicht immer gut zu interpretieren sind ist erkennbar, dass es einen insgesamt hohen und gleichmäßigen Vernetzungsgrad gibt. Um einen solchen Graphen zu erhalten, wählt man die Anzahl  $N$  der Knoten und verbindet dann zunächst jeden Knoten mit seinen  $d$  nächsten Nachbarn. Wie groß  $N$  und  $d$  gewählt werden, muss im Einzelfall entschieden werden. Für die hier beschriebenen Netzwerke wurden aus Gründen der Darstellbarkeit  $N = 500$  und  $d = 10$  gesetzt. Auf diese Weise entsteht eine Struktur, die in der Graphentheorie als *Ring* bezeichnet wird.

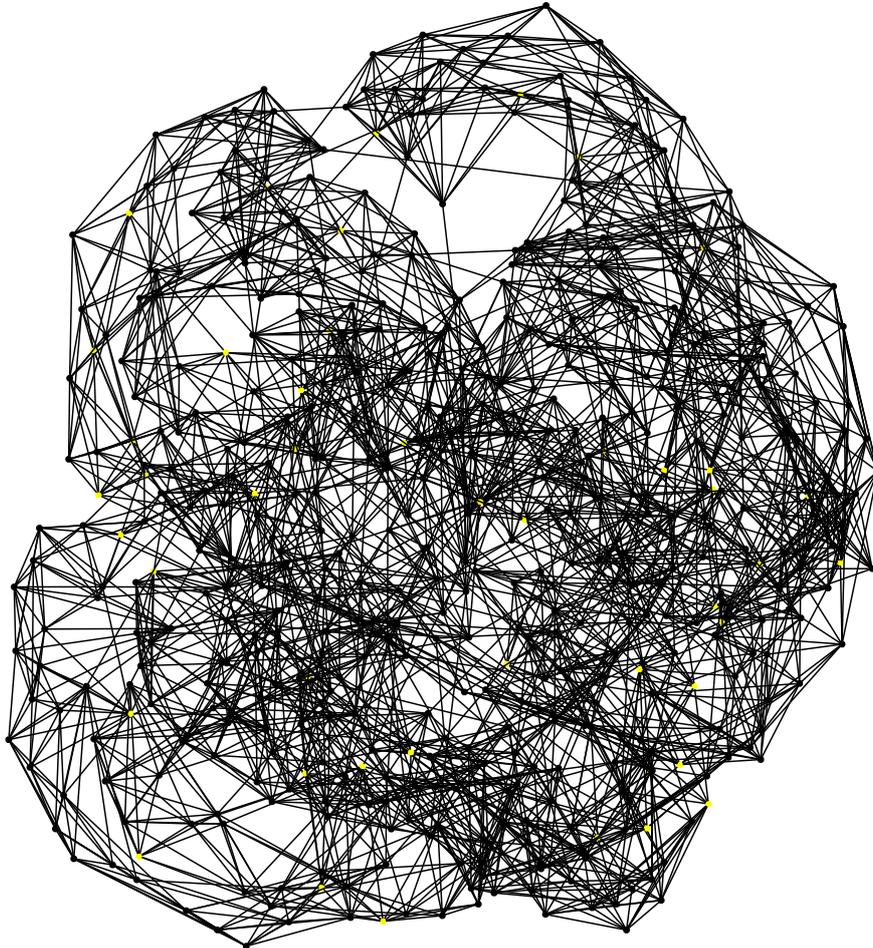


Abbildung 7.1: Graph nach Newman und Watts (1999), bestehend aus 500 Knoten. Bei der Konstruktion des Graphen erhielt jeder Knoten zunächst 5 Kanten zu seinen nächsten Nachbarn. In einem weiteren Schritt erhielten dann ca. 30% der Knoten je eine zusätzliche Kante zu einem zufällig ausgewählten Knoten.

Zusätzlich dazu erhält nun jeder Knoten mit einer gewissen Wahrscheinlichkeit  $p$  eine zusätzliche Kante zu einem zufällig ausgewählten Knoten des Graphen. Auch die Wahl von  $p$  kann variiert werden, mit unterschiedlichen Konsequenzen. Setzt man  $p = 0$ , so erhält kein Knoten eine zusätzliche Kante und es bleibt bei dem gerade be-

schriebenen Ring. Setzt man dagegen  $p = 1$ , so erhält jeder Knoten eine zusätzliche Kante. Die Folge davon wäre ein stark vernetzter Graph, in dem zusätzlich zu der Vernetzung innerhalb der Nachbarschaft für jeden Knoten eine zusätzliche Kante zu einem weiter entfernten Knoten dazu käme. Dadurch könnte man von jedem Knoten aus in relativ wenigen Schritten jeden anderen Knoten des Graphen erreichen, eine Eigenschaft, die auch als Small-World-Eigenschaft bekannt ist.

Wählt man nun ein  $0 < p < 1$ , so bleibt die Small-World-Eigenschaft auch bei insgesamt geringerer Vernetzung erhalten. Bei der Konstruktion von Graphen nach Newman und Watts (1999) wird genau so vorgegangen. Man wählt einen Parameter zwischen null und eins und fügt dem Graphen auf diese Weise weitere, zufällige Kanten hinzu. Bei den hier beschriebenen Graphen wurde der Parameter  $p = 0.3$  gewählt. Die Verteilung der Knotengrade der Graphen, die den hier beschriebenen Simulationen zugrunde liegt, ist in Abbildung 7.2 zu sehen. Man sieht deutlich, dass die meisten Knoten einen Grad von 10 haben, einige aber auch einen Grad von 11. Dass es zusätzlich Knoten mit einem Grad von 12 oder höher gibt, liegt daran, dass bei der oben beschriebenen Konstruktionsweise nicht ausgeschlossen ist, dass einige Knoten mehr als einmal als Ziel einer neuen Kante ausgewählt werden können. Außerdem ist es möglich, dass ein Knoten sowohl Start- als auch als Zielknoten einer solchen neuen Kante wird. Man sieht aber auch, dass die Anzahl dieser Knoten vergleichsweise gering ist.

In dem resultierenden Netzwerk verfügen die meisten Knoten über einen Grad von 10, ca. 30% der Knoten besitzen aber eine zusätzliche Kante zu einem zufällig ausgewählten Knoten. Durch die zusätzlichen Kanten sind die Knoten nicht nur mit ihrer direkten Nachbarschaft, sondern auch mit ursprünglich weiter entfernten Knoten verbunden. In Bezug auf ein soziales Netzwerk kann das so interpretiert werden, dass es zwischen verschiedenen Freundes- oder Bekanntenkreisen Querbezüge gibt. Soziale Akteure können sich in unterschiedlichen sozialen Umfeldern bewegen und Informationen zwischen ihnen transportieren.

### **Ausbreitung von Fake News**

Der Graph aus Abbildung 7.1 lag den insgesamt 1000 Simulationsläufen zugrunde. Bei der Initialisierung wurde die Falschmeldung 10% der Knoten bereits mitgege-

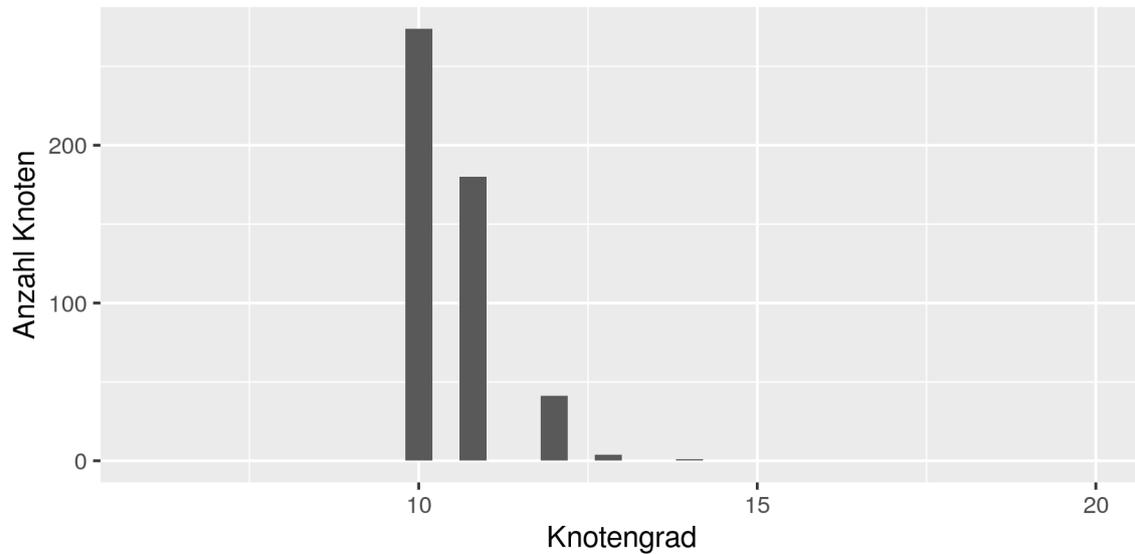


Abbildung 7.2: Verteilung der Knotengrade eines Newman-Watts Graphen bestehend aus 500 Knoten. Bei der Konstruktion erhielt jeder Knoten zunächst Kanten zu seinen 10 nächsten Nachbarn. Im so entstandenen Ring erhielt dann jeder Knoten mit einer Wahrscheinlichkeit von  $p = 0.3$  eine zusätzliche Kante zu einem zufällig ausgewählten Knoten.

ben. Diese Knoten wurden in jedem Simulationslauf aufs Neue zufällig ausgewählt. Abbildung 7.1 zeigt eine solche Zufallsauswahl, die initialisierten Knoten sind gelb markiert. Die Verbreitung der Fake News geht also von diesen Knoten aus.

Abbildung 7.3 zeigt die Verbreitung der Fake News über 50 Zeitschritte hinweg, bei Durchführung von 1000 Simulationsläufen. Auf der  $y$ -Achse ist die Anzahl der Knoten abgetragen, die die Falschmeldung übernommen haben. Ein Punkt repräsentiert die Anzahl von Knoten “mit Falschmeldung” in einem speziellen Programmlauf. Zu Beginn der Simulation glauben entsprechend der Initialisierung 10%, also 50 soziale Akteure an die Falschmeldung. Im weiteren Verlauf ändert sich diese Zahl. Da das Modell eine Wahrscheinlichkeitsverteilung beschreibt, ist diese Änderung von Simulationslauf zu Simulationslauf verschieden. Die Varianz ist zu Beginn der Simulation am geringsten, erreicht nach ca. 20 Zeitschritten ein Maximum, bei dem eine Schwankung von ca. 50 Knoten nach unten und nach oben vorkommt, und wird gegen Ende hin wieder etwas geringer. Nach 50 Zeitschritten hat nahezu jeder Knoten

die Falschmeldung übernommen.

Die S-förmige Kurve ähnelt dem, was Bonabeau (2002) in seinen Betrachtungen von sozialer Diffusion fand. Dieses Ergebnis ist dahingehend interessant, als dass Bonabeau (2002) keine probabilistische, sondern eine deterministische Simulation durchgeführt hat. Die Wahrscheinlichkeitsverteilung, die durch das Probabilistische Programm beschrieben wird, scheint also mit den Annahmen von Bonabeau einherzugehen.

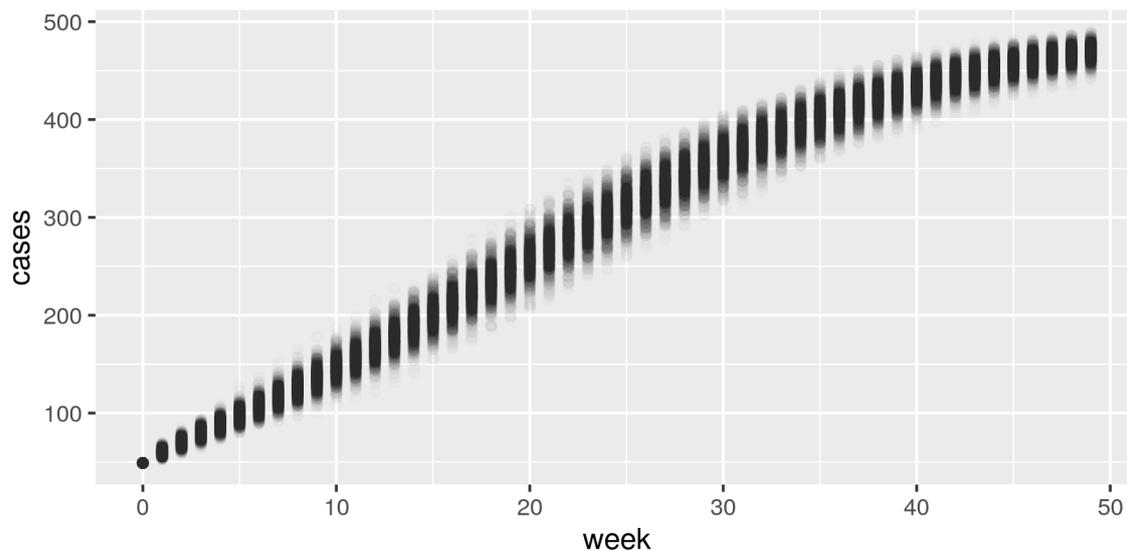


Abbildung 7.3: Entwicklung der Verbreitung der Fake News in dem in Abbildung 7.1 dargestellten sozialen Netzwerk über 50 Zeitschritte hinweg. Auf der  $y$ -Achse sind die Anzahl der Mitglieder des Netzwerks abgetragen, die die Fake News übernommen haben.

Abbildung 7.4 zeigt die Verteilungen für die unterschiedlichen Zeitschritte von Sim-1, zum besseren Vergleich der Ergebnisse für die einzelnen Zeitschritte. In dieser Abbildung ist jeder Wert durch einen senkrechten Strich repräsentiert. Die einzelnen Striche sind außerdem leicht transparent. Die Bereiche, an denen die Striche dunkler erscheinen, sind diejenigen, in denen viele Werte vorhergesagt wurden. Diese Bereiche sind auf Basis des Modells wahrscheinlicher als die anderen. Aus dieser Darstellung wird deutlich, dass auch da, wo die Streuung breiter wird, die höchste Konzentration der Werte in einem mittleren Bereich liegt.

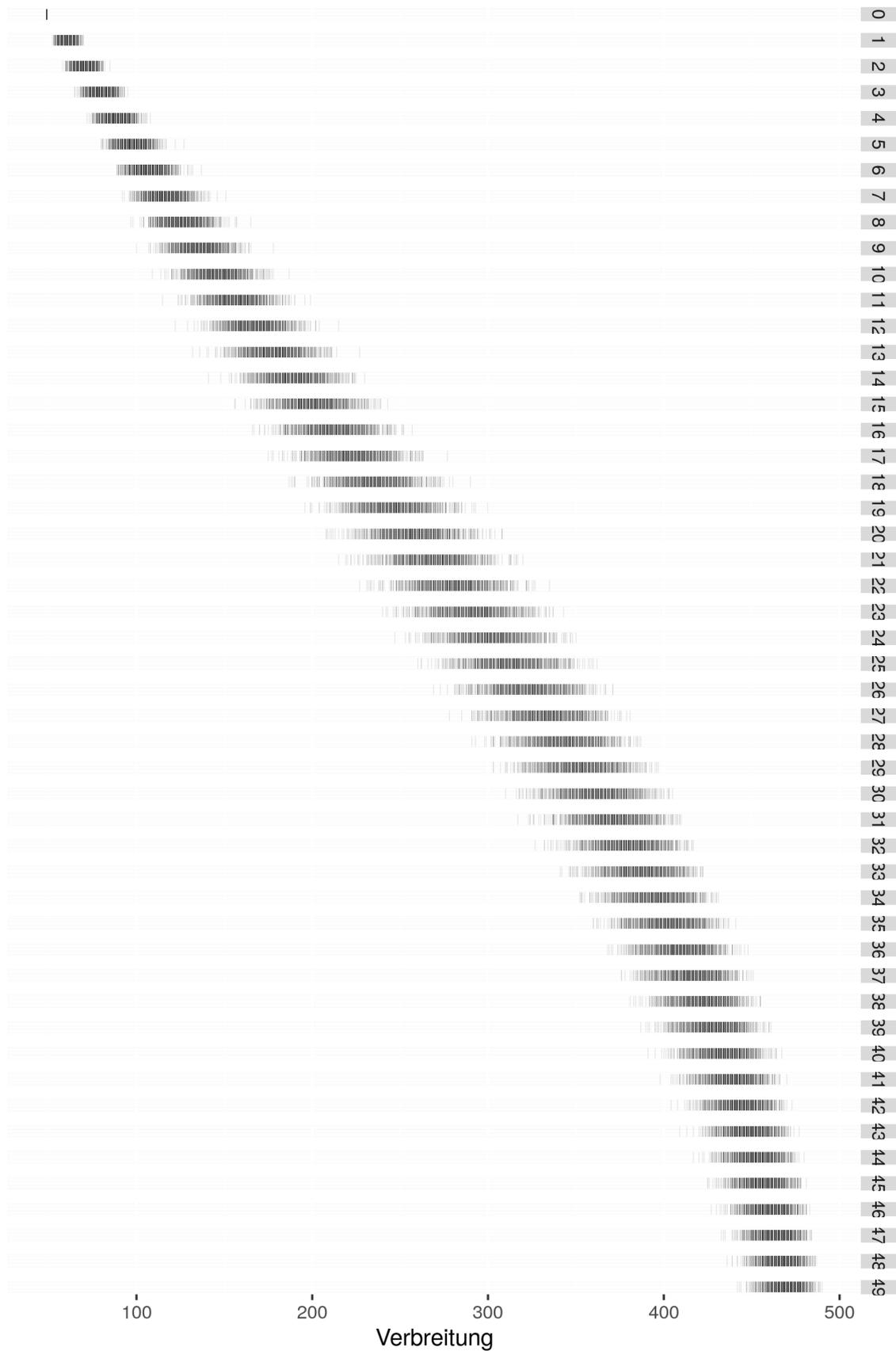


Abbildung 7.4: Verteilungen der Verbreitung von Fake News pro Zeitschritt im Newman-Watts Graphen aus Abbildung 7.1. Die Skalierung der  $x$ -Achse variiert, damit der tatsächliche Wertebereich besser dargestellt werden kann.

## 7.2.2 Simulation auf Basis eines Barabási-Albert-Graphen

### Konstruktion eines Barabási-Albert-Graphen

Bei der Konstruktion eines Graphen nach Barabási und Albert (1999) ist die Anzahl  $N$  der Knoten zunächst zweitrangig. Grund dafür ist der dynamische Konstruktionsprozess, in dem Schritt für Schritt neue Knoten zu einem Graphen hinzugefügt werden, und der theoretisch unendlich lang fortgesetzt werden kann. Für Sim-2 wurde  $N$ , wie schon für Sim-1, auf 500 begrenzt. Wichtig ist nun die Anzahl  $d$  der Kanten, die neu zum Graphen hinzukommende Knoten maximal in den Graphen einbringen. Dabei ist allerdings zu beachten, dass im resultierenden Graphen nicht mehr alle Knoten notwendigerweise einen Grad von  $d$  haben werden. Die grundlegende Idee bei der Entstehung eines Graphen nach Barabási und Albert (1999) besteht nämlich darin, dass neue Knoten sich bevorzugt mit den Knoten verbinden, die bereits stark vernetzt sind. Der Grad dieser Knoten wird deshalb größer als  $d$  sein. Je größer der Grad eines Knoten dabei zu Anfang ist, desto mehr Kanten wird er im Verlauf des Prozesses zusätzlich erhalten. Aus diesem Grund spricht man auch von *preferential attachment*. Für Sim-2 wurde  $d = 5$  gesetzt.

Um nun ein solches Netzwerk aufzubauen, betrachtet man in einem ersten Schritt eine kleine Anzahl von Knoten, die noch nicht miteinander vernetzt sind. Zu diesem sogenannten *Nullgraphen* wird dann ein neuer Knoten hinzugefügt, der Kanten zu maximal  $d$  Knoten erhalten soll. Wie oben schon gesagt, wird er sich bevorzugt mit den Knoten verbinden, die bereits einen relativ hohen Grad haben. Im speziellen Fall des Nullgraphen, wo noch keine Kanten im Graphen vorhanden sind, folgt diese Entscheidung einer Gleichverteilung. Ab dem zweiten hinzukommenden Knoten ist dann die Wahrscheinlichkeit, dass ein bereits im Graphen vorhandener Knoten mit diesem verbunden wird, proportional zum Grad des Knoten. Grundsätzlich handelt es sich hier also um eine probabilistische Entscheidung.

Ein solcher Barabási-Albert-Graph ist in Abbildung 7.5 zu sehen. Nach der Konstruktion des Graphen wurden auch hier 10% der Knoten die Fake News mitgegeben. Diese Knoten sind in der Abbildung gelb markiert.

Die Verteilung der Knotengrade ist in Abbildung 7.6 zu sehen. Die Verteilung ist typisch für Graphen dieses Typs und unterscheidet sich stark von der Verteilung der Knotengrade für Newman-Watts-Graphen (vgl. Abbildung 7.2). Der Großteil

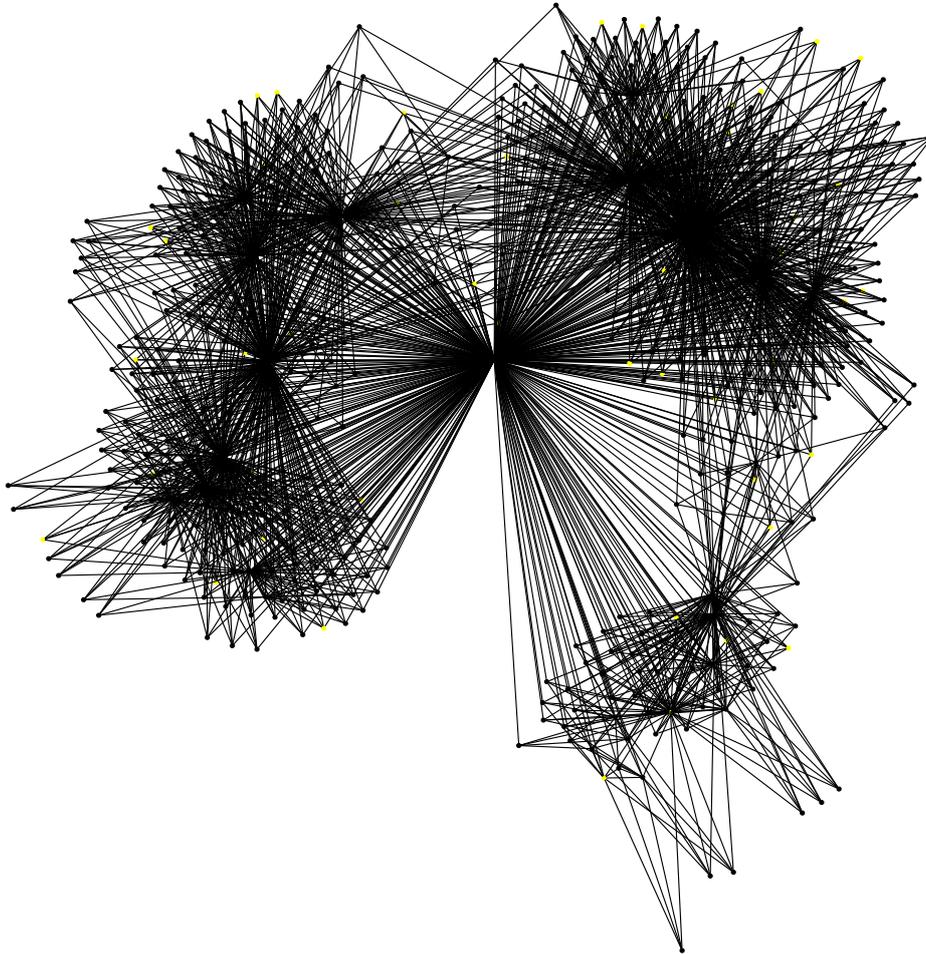


Abbildung 7.5: Graph nach Barabási und Albert (1999). Der Graph besteht aus 500 Knoten. Bei der Entstehung des Graphen brachte jeder neue Knoten maximal 5 neue Kanten in den Graphen ein, wobei die Zielknoten dieser neuen Kanten proportional zu deren Grad ausgewählt wurden. Nach Entstehung wurde 10% der Knoten die Falschmeldung mitgegeben. Diese Knoten sind gelb eingezeichnet.

der Knoten hat einen relativ geringen Grad, der bei  $d = 5$  liegt, entsprechend den Vorschriften, die bei der Konstruktion gemacht wurden. Es gibt aber auch Knoten, die bis zu 400 Kanten haben. Dies sind nur relativ wenige Knoten, wie aus der

Abbildung deutlich wird. Es gibt also in dem Graph aus Abbildung 7.5 einige wenige Knoten, die zu sehr vielen anderen Knoten des Graphs Verbindungen haben. Die meisten Knoten haben aber nur relativ wenige Verbindungen.

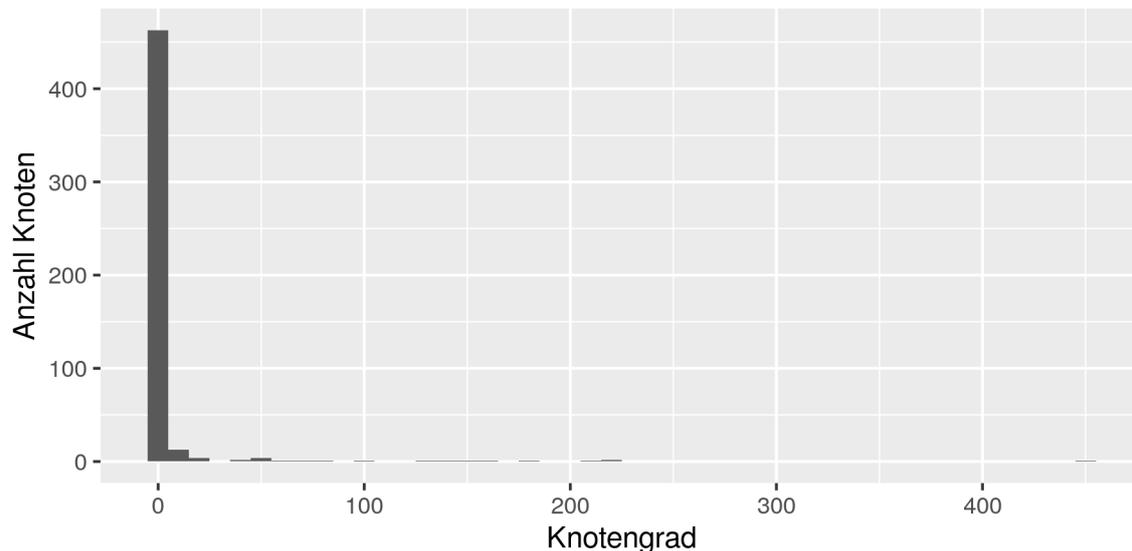


Abbildung 7.6: Verteilung der Knotengrade des Barabási-Albert-Graphen.

### Ausbreitung von Fake News

Die Entwicklung der Ausbreitung der Falschmeldung in Barabasi-Albert Graphen über 50 Zeitschritte ist in Abbildung 7.7 zu sehen. Der Verlauf unterscheidet sich in den Details von dem in Abbildung 7.3, folgt aber einer ähnlichen Form.

Auf der  $y$ -Achse ist die Anzahl der Knoten abgetragen, die die Falschmeldung übernommen haben. Wie in Sim-1 repräsentiert ein Punkt die Anzahl der Knoten “mit Falschmeldung” in einem Simulationslauf. Zu Beginn der Simulation gibt es keine Varianz, da immer 10% der Knoten die Falschmeldung mitgegeben wurde. Die Varianz steigt dann langsam an und erreicht nach ca. 25 Zeitschritten ein Maximum, also etwas später als das in Sim-1 der Fall war. Die Variationsbreite ist aber, zumindest dem Augenschein nach, in einer vergleichbaren Größenordnung wie in Sim-1. Auch in Sim-2 haben nach 50 Zeitschritten nahezu alle Knoten die Falschmeldung übernommen.

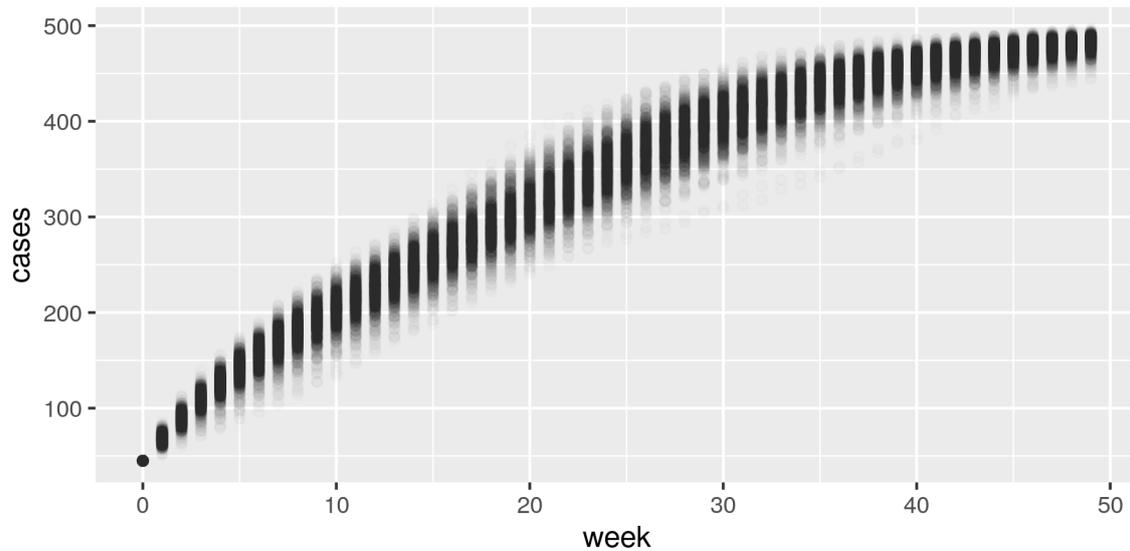


Abbildung 7.7: Entwicklung der Verbreitung von Fake News in dem in Abbildung 7.5 dargestellten sozialen Netzwerk über 50 Zeitschritte hinweg. Auf der  $y$ -Achse sind die Anzahl derjenigen Mitglieder des Netzwerks abgetragen, die die Fake News übernommen haben.

## 7.3 Diskussion

Vergleicht man die Ergebnisse von Sim-1 und Sim-2, dann fällt zunächst auf, dass in beiden Fällen eine ähnliche Dynamik der Ausbreitung von Fake News vorhergesagt wird. Das ist auf der einen Seite überraschend, denn die zwei Graphen, die den Simulationen zugrunde liegen, unterscheiden sich deutlich hinsichtlich ihrer Topologie, wie ein Blick auf die Verteilungen der Knotengrade zeigt. Andererseits werden sowohl Graphen nach Newman-Watts als auch Graphen nach Barabási-Albert als Small-World-Graphen bezeichnet, verfügen also beide über die Eigenschaft, dass man von jedem Knoten aus jeden anderen Knoten in relativ wenigen Schritten erreichen kann. Die große Ähnlichkeit der Ergebnisse liegt vermutlich darin begründet.

Schaut man sich die Unterschiede an, die dennoch bestehen, dann ist der Verlauf der Ausbreitung im Barabási-Albert-Graphen über die Zeit hinweg nicht ganz so regelmäßig wie im Newman-Watts-Graphen. Die Streuung der Anzahl der Knoten, die die Falschmeldung übernehmen, ist in Sim-2 etwas größer als in Sim-1. Ein

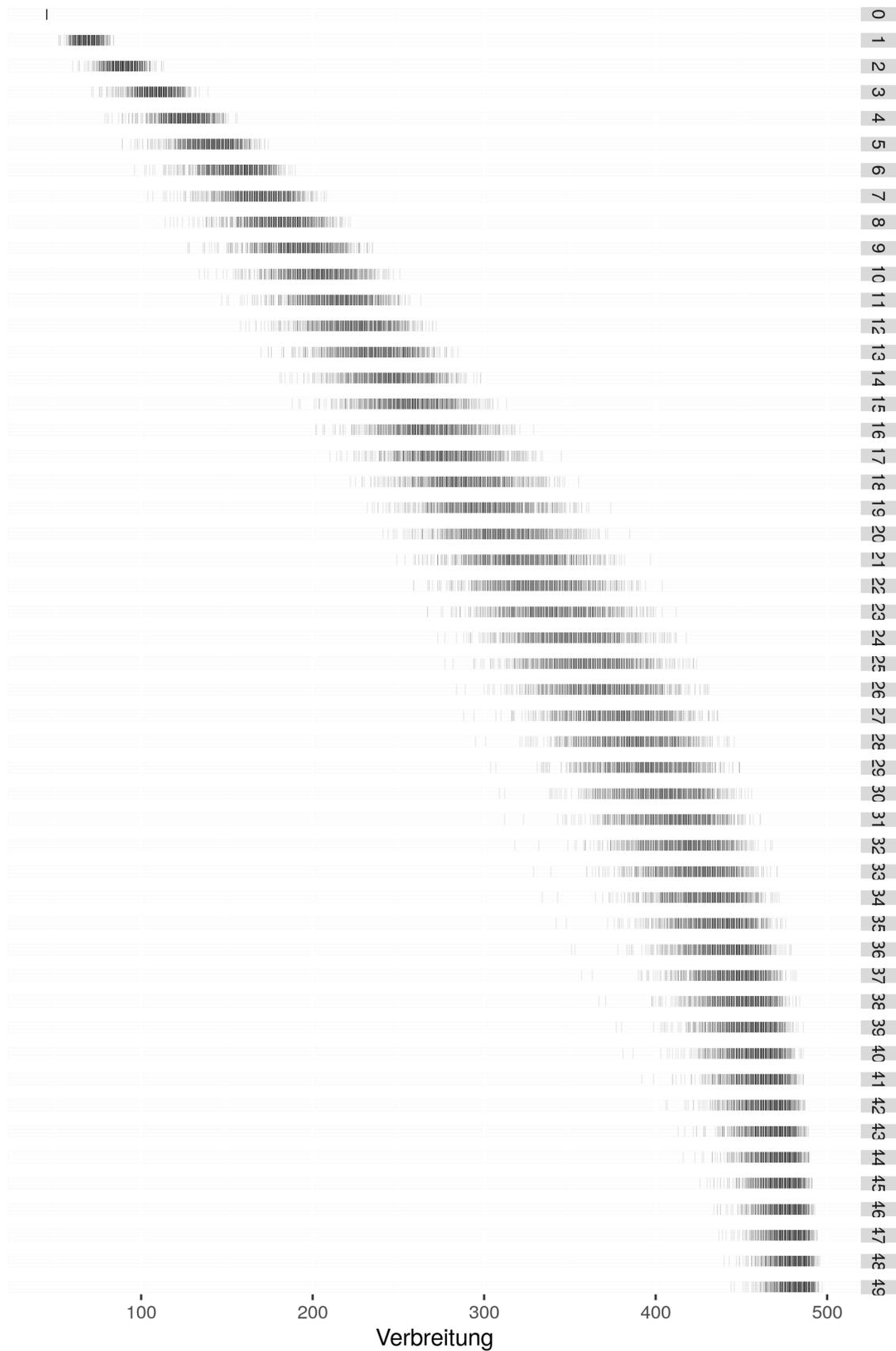


Abbildung 7.8: Verteilungen der Verbreitung von Fake News pro Zeitschritt im Barabasi-Albert-Watts Graphen aus Abbildung 7.5. Die Skalierung der  $x$ -Achse variiert, damit der tatsächliche Wertebereich besser dargestellt werden kann.

Grund hierfür könnte sein, dass es in manchen der 1000 Simulationsläufe etwas länger gedauert hat, bis die Falschmeldung die stark vernetzten Knoten erreicht hat. Im Newman-Watts Graphen gibt es diese großen Unterschiede in der Vernetzung der Knoten erst gar nicht, so dass man hier insgesamt eine gleichmäßigere Ausbreitung erwarten kann.

Mit Blick auf die Topologie der verwendeten Graphen ist zu sagen, dass es sich um idealisierte Typen von Netzwerken handelt. Watts (2004b) liefert einen ausführlichen Überblick (nicht nur) über unterschiedliche Typen von Small-World-Netzwerken und zeigt Beispiele auf, in denen solche Netzwerke verschiedene Eigenschaften sozialer Netzwerke gut abbilden. Vor allem für Newman-Watts Graphen macht er aber auch deutlich, dass sie aufgrund verschiedener Probleme für die Repräsentation von sozialen Netzwerken nur bedingt geeignet sind. Netzwerke nach Barabási-Albert scheinen hier besser empirisch abgesichert zu sein. Unter den frühen Beispielen, die von Barabási und Bonabeau (2003) genannt werden, sind ein Netzwerk aus Zitationen in wissenschaftlichen Publikationen und ein Netzwerk von Schauspielern aus Hollywood.

Während diese Beispiele sich auf eher kleine Netzwerke beziehen und inhaltlich für die sozialwissenschaftliche Forschung von geringem Interesse sein dürften, bringen spätere Publikationen Anhaltspunkte dafür, dass die Klasse von Netzwerken, die Barabási und Albert (1999) bekannt gemacht haben, sich auch für die Forschung zu größeren und in einem breiteren Kontext relevanten Netzwerken eignet. So gibt es empirisch fundierte Forschung zu Netzwerken sexueller Kontakte (Blasio, Svensson & Liljeros, 2007), *Followern* auf Twitter (Villazon-Terrazas, Aparicio & Alvarez, 2015; Tinati, Carr, Hall & Bentwood, 2012) oder Freundschaftsbeziehungen auf Facebook (Davoudi & Chatterjee, 2015), die sich durch skalenfreie Netzwerke beschreiben lassen.

Die Verwendung von skalenfreien Netzwerken zur Modellierung von Systemischen Risiken in sozialen Netzwerken scheint vor diesem Hintergrund ein gangbarer Weg zu sein. Es ist Aufgabe zukünftiger Forschung, empirische Daten für weitere soziale Netzwerke zu sammeln, um die Topologie realer Netzwerke mit den hier verwendeten (und anderen) Netzwerken zu vergleichen. Interessant ist aber auch, um auf den Anfang dieser Diskussion zurückzukommen, dass die Ergebnisse der durchgeführten Simulationen sich für beide Typen von Netzwerken nicht qualitativ unter-

scheiden. Vor diesem Hintergrund ist es eventuell zu rechtfertigen, auch einfachere Netzwerkstrukturen wie die von Newman und Watts (1999) bei der Modellierung zu verwenden, wenn dies zur besseren Durchführbarkeit eines Forschungsvorhabens führen kann.



# Kapitel 8

## Fallbeispiel 3: Soziale Ungleichheit und vernetzte Risiken

Im dritten Fallbeispiel soll ein Aspekt betrachtet werden, der von Renn et al. (2017) angesprochen wird, die Vernetzung von Risiken. Anhand eines Probabilistischen Programms wird dazu ein Prozess modelliert, in dessen Verlauf ein Zusammenhang zwischen sozialer Ungleichheit und der Struktur eines sozialen Graphen bzw. der Veränderung eben dieser Struktur zu Tage tritt.

Auch dieses Modell steht in Einklang mit dem Rahmenmodell. Das System, um das es geht, ist wieder ein soziales Netzwerk, das als Ausgangspunkt dient. Wie im vorigen Kapitel handelt es sich also um ein virtuelles System, das nicht als Repräsentation eines tatsächlich existierenden sozialen Netzwerks konzipiert wurde. Es zeigt sich also schon in diesem Punkt wieder der Charakter des toy models. Das rekursive Element in diesem Beispiel ist ein System aus zwei Elementen, die sich gegenseitig rekursiv beeinflussen. Dabei handelt es sich einerseits um die Menge eines Gutes, die jeder Teilnehmer zur Verfügung hat, und andererseits um die Verbindungen zu anderen Knoten. Der genaue Mechanismus wird weiter unten ausführlich erläutert.

Der Quellcode dieses Simulationsmodells ist in Anhang C abgedruckt.

## 8.1 Beschreibung des Modells

Zu Beginn des Prozesses wird im Graphen ein “Gut” verteilt. Dabei handelt es sich um eine Zahl zwischen 0 und 1. Das Gut selbst wird nicht näher spezifiziert, es ist aber zweckmäßig, sich hier eine monetäre Größe vorzustellen. Jeder Knoten erhält also eine bestimmte Menge dieses Gutes. Wie viel ein Knoten bekommt, bleibt dem Zufall überlassen. Genauer wird für jeden Knoten des Graphen ein Wert aus der kontinuierlichen Gleichverteilung zwischen 0 und 1 gezogen. Die Ungleichheit besteht demnach in Bezug auf die Menge des Gutes. Im weiteren Verlauf wird hierfür der Einfachheit halber auch die Bezeichnung “Reichtum” verwendet.

Nach dieser Initialisierung beginnt der eigentliche Prozess, der in diskrete Zeitschritte unterteilt ist und in dessen Verlauf zwei Dinge passieren. Zum einen wird in jedem Schritt erneut das Gut verteilt. Jetzt aber richtet sich die Menge nach dem Reichtum der Knoten. Sie wird zwar immer noch als Zufallsvariable modelliert, aber diesmal nicht als gleichverteilte, sondern als normalverteilte Zufallsvariable. Mittelwert der Normalverteilung ist dabei die Menge, über die ein Knoten bereits verfügt, die Standardabweichung wurde auf  $\sigma = 0.01$  festgesetzt. Auf diese Weise werden diejenigen, die zu Anfang vom Zufall begünstigt wurden, auch weiterhin eher mehr bekommen und diejenigen, die anfangs wenig bekamen, werden vermutlich nur einen geringen Zuwachs an Reichtum erlangen. Dieser Mechanismus ist unter verschiedenen Namen in die Literatur eingegangen. Meadows und Wright (2008) etwa bezeichnen es als *Success to the Successful*, Albert und Barabási (2002) benutzen den Begriff des *Preferential Attachment*. Es sei aber nochmals betont, dass die genaue Entwicklung des Reichtums nicht deterministisch ist. Insbesondere ist es möglich, wenn auch nicht sehr wahrscheinlich, dass ein “armer” Knoten viel bzw. ein “reicher” Knoten wenig erhält.

Zum anderen ändert sich die Struktur des Graphen im Laufe des Prozesses. Jeder Knoten hat das Bestreben, seine Verbindungen derart gegen neue einzutauschen, dass er mit möglichst reichen Knoten in Kontakt steht. Um das zu erreichen, werden in jedem Zeitschritt Anfragen an reiche Knoten gestellt, eine Verbindung einzugehen. Die reichen Knoten können aber eine Anfrage auch ablehnen und das tun sie dann, wenn die Differenz des Reichtums zwischen ihnen selbst und einem anfragendem Knoten zu groß ist. Der Schwellenwert wurde für das Modell auf ein

Viertel des Reichtums des angefragten Knoten festgesetzt. Wenn also die Differenz größer als dieser Schwellenwert ist, dann wird eine Anfrage auf jeden Fall abgelehnt. Andernfalls wird die Anfrage mit einer gewissen Wahrscheinlichkeit angenommen. Die Entscheidung ist ebenfalls abhängig von der Differenz und wurde als Bernoulli-verteilte Zufallsvariable mit Parameter  $p = 1 - D$  modelliert, wobei  $D$  die Differenz bezeichnet.

Nach einer erfolgreichen Verbindungsanfrage eines Knotens kann sich dieser dazu entscheiden, eine seiner alten Verbindungen zu lösen. Er tut dies allerdings nur dann, wenn es unter seinen Nachbarn einen Knoten gibt, der im Vergleich zu ihm selbst zu "arm" ist. Um das festzustellen, kommt der gleiche Schwellenwert zum Einsatz, der schon oben beschrieben wurde. Und auch bei Überschreitung des Schwellenwerts wird die Verbindung nur mit einer gewissen Wahrscheinlichkeit gelöst, die im Modell mit  $1/2$  festgesetzt wurde.

Die Simulation wurde in zwei Varianten durchgeführt, die sich durch die Topologie des Graphen unterscheiden, der zum Ausgangspunkt des oben beschriebenen Prozesses wird. Die Varianten werden im Folgenden als Sim-1 bzw. Sim-2 bezeichnet. In Sim-1 wird ein Small-World-Graph nach Newman und Watts (1999), der in diesem Fall aus 500 Knoten besteht, zu Grunde gelegt. Jeder Knoten besitzt 10 Verbindungen, entsprechend der Konstruktionsweise des Graphen erhält jeder Knoten zusätzlich mit Wahrscheinlichkeit von  $P = 0.3$  eine zusätzliche Kante zu einem ebenfalls zufällig gewählten Knoten. Ausgangspunkt von Sim-2 ist ein Graph nach Barabási und Albert (1999). Auch dieser Graph besteht aus 500 Knoten mit jeweils 10 Kanten. Einen zusätzlichen Parameter gibt es hier nicht. Stattdessen wird der Graph Knoten für Knoten aufgebaut, wobei die Wahrscheinlichkeit, mit der ein Knoten im Graphen eine Kante zu dem neu hinzukommenden Knoten bekommt, proportional von seinem Grad abhängt.

## 8.2 Ergebnisse

### Sim-1

In der ersten Variante dieser Simulation wurde das soziale Netzwerk als Small-World-Graph nach Newman und Watts (1999) modelliert. Eine graphische Darstellung des

Graphen ist in Abbildung 8.1 zu sehen. Auch wenn Einzelheiten in einer solchen Grafik nicht zu erkennen sind, die für solche Small-World-Netzwerke charakteristischen Eigenschaften sind zu erkennen: Der Graph ist zusammenhängend, es gibt keine Unterstruktur, die nicht mit dem Rest des Graphen durch mindestens eine Kante verbunden ist. Des Weiteren ist der Grad der Vernetzung sehr hoch, wodurch die Möglichkeit besteht, von jedem Knoten des Graphen aus jeden anderen in relativ wenigen Schritten zu erreichen.

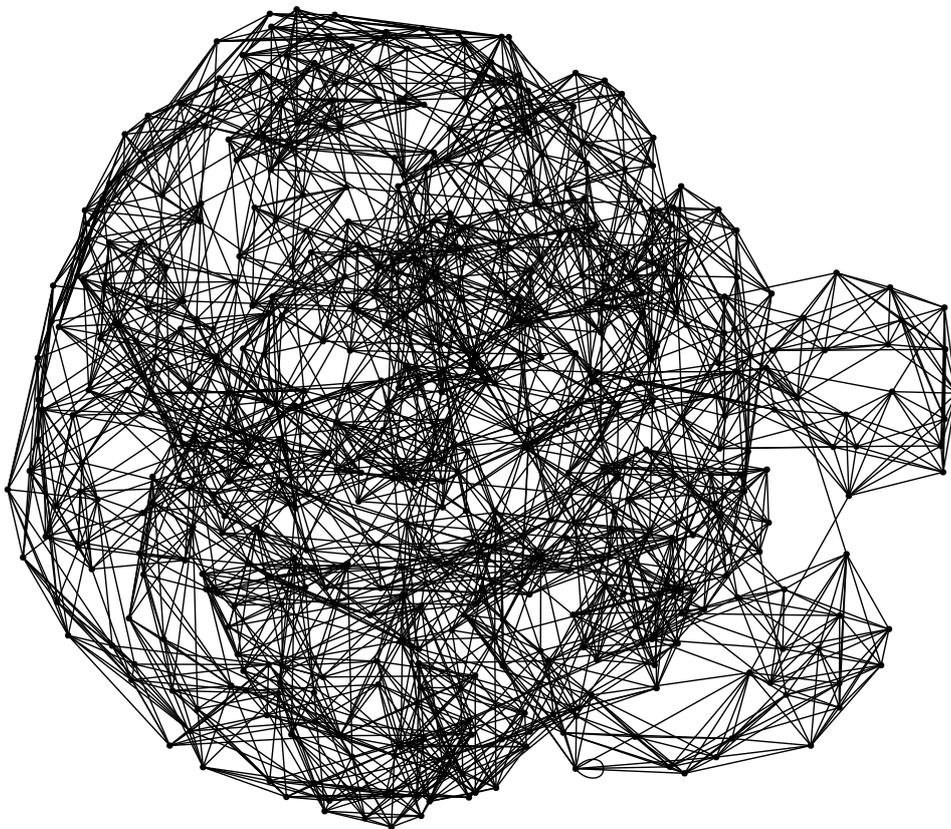


Abbildung 8.1: Initialisierter Newman-Watts-Graph.

Abbildung 8.4 zeigt die Verteilung des Gutes vor und nach der Simulation. Zu

beachten ist, dass die Skalierung der  $x$ -Achsen unterschiedlich ist, und darin zeigt sich auch der Effekt des spezifizierten Prozesses. Vor der Simulation ist das Gut gleichmäßig zwischen 0 und 1 verteilt. Das entspricht genau dem, was bei der Initialisierung des Graphen geschehen sollte. Der untere Teil der Abbildung zeigt nun die Verteilung aggregiert über alle Programmläufe. Es ergibt sich hier zwar ebenfalls eine relativ gleichmäßige Verteilung, diesmal aber über einem deutlich größeren Bereich. Tatsächlich scheint sich die Verteilung nur wenig hinsichtlich ihrer Form, dafür aber umso mehr hinsichtlich der Skalierung verändert zu haben. Während der Maximalwert vor der Simulation bei 1 lag, liegt er danach aggregiert über alle Simulationen bei 1 Mio.

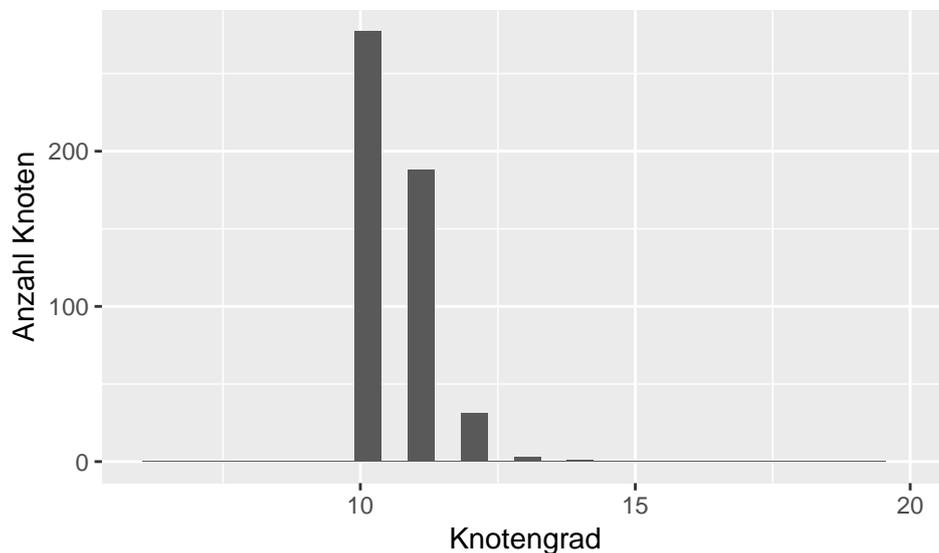


Abbildung 8.2: Verteilung der Knotengrade vor Sim-1. Entsprechend der Vorschriften zur Konstruktion eines Newman-Watts-Graphen haben die meisten Knoten einen Grad von 10, ca. 30% haben durch das Hinzufügen einer zusätzlichen Kante einen Grad von 11. Weil beim Hinzufügen das mehrfache Auswählen von Knoten erlaubt ist, gibt es außerdem einige Knoten mit einem Grad höher als 11.

Vor diesem Hintergrund kann nun die Topologie des Graphen vor und nach der Simulation betrachtet werden. Ein genauerer Blick auf die Vernetzung des Graphen vor der Simulation ist in Abbildung 8.2 zu sehen. Abgebildet ist die Verteilung der Knotengrade vor der Simulation. Die Mehrheit der Knoten besitzt 12 Verbindungen, was genau der Konstruktionsvorschrift entspricht, die bei der Erstellung des Graphen

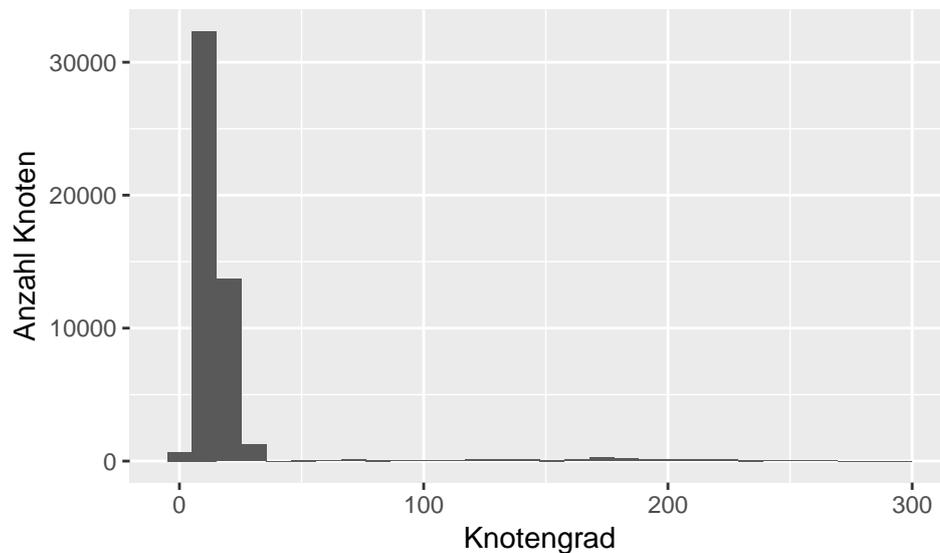


Abbildung 8.3: Verteilung der Knotengrade nach Sim-1, aggregiert über alle Programmläufe. Auf der  $x$ -Achse ist die Anzahl der Knoten, auf der  $y$ -Achse der Knotengrad abgetragen. Der Knotengrad der Mehrzahl der Knoten liegt nach der Simulation zwischen 2 und ca. 30. Es gibt aber auch einige wenige Knoten mit einem deutlich höheren Grad.

angewandt wurde. Die nächste Gruppe bilden die Knoten mit einem Knotengrad von 13. Das sind diejenigen Knoten, die bei der Erstellung des Graphen eine zusätzliche Kante zu einem zufällig ausgewählten Knoten des Graphen erhalten haben. Dass es außerdem Knoten mit Grad 14 bzw. 15 gibt, liegt daran, dass bei der zufälligen Auswahl der Zielknoten für die zusätzlichen Verbindungen eines Newman-Watts-Graphen die mehrfache Auswahl nicht ausgeschlossen ist. Es passt aber auch zur Art der Konstruktion, dass die Anzahl dieser Knoten deutlich geringer ist.

Ein ganz anderes Bild ergibt sich nach der Simulation. Abbildung 8.3 zeigt die Verteilung der Knotengrade nach der Simulation, aggregiert über alle Programmläufe. Die größte Anzahl bilden nun Knoten mit nur wenigen oder gar keinen Verbindungen. Auf der anderen Seite gibt es eine kleine Anzahl an Knoten mit sehr vielen Verbindungen. Zwischen diesen Extremen verteilen sich die Knoten in etwa gleich. Die Abbildung zeigt, dass sich die Struktur des Graphen durch den Prozess stark verändert hat.

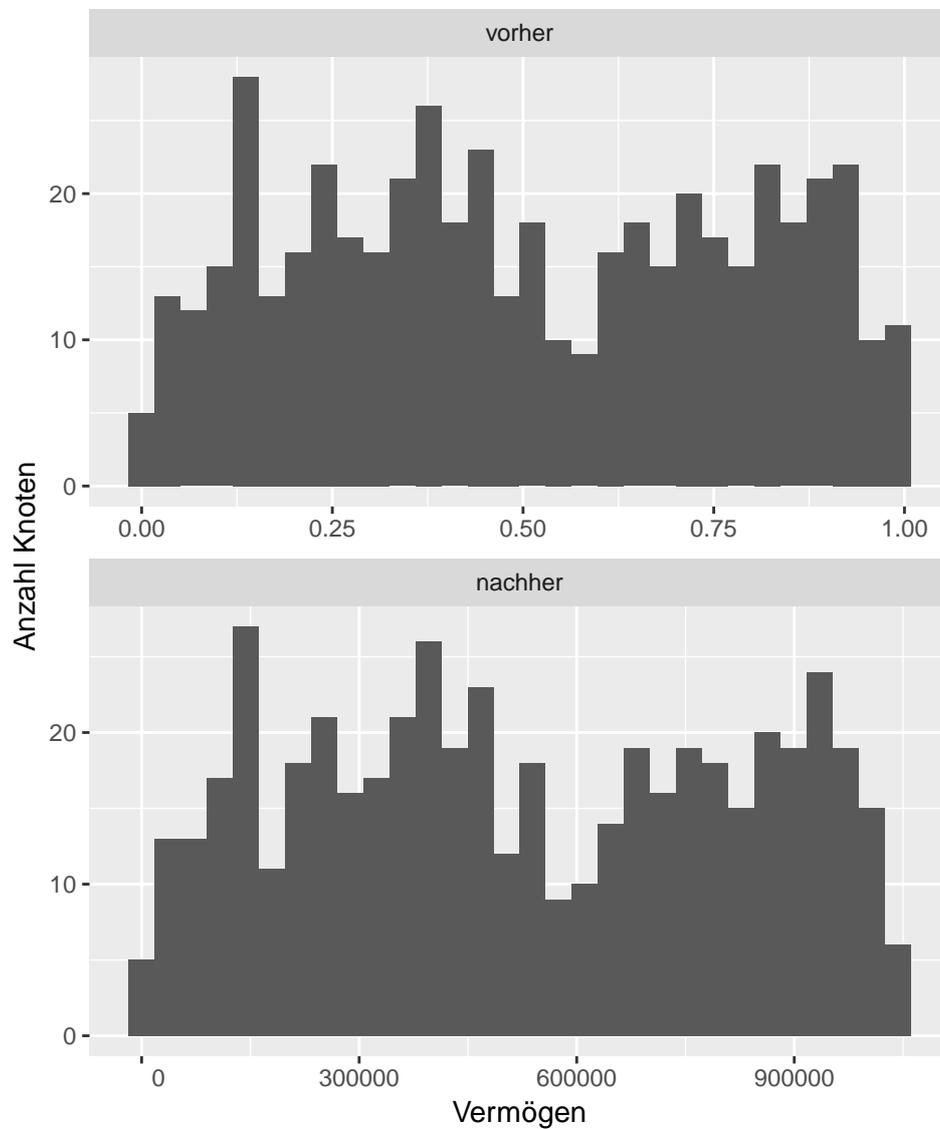


Abbildung 8.4: Verteilung des Guts vor und nach der Sim-1. Der obere Teil der Abbildung zeigt die Verteilung des Guts vor der Simulation, es ist gleichmäßig verteilt zwischen 0 und 1. Der untere Teil zeigt die Verteilung nach der Simulation, aggregiert über alle Simulationen. Die Verteilung ist ebenfalls gleichmäßig, allerdings über einem Intervall von 0 bis 1000000.

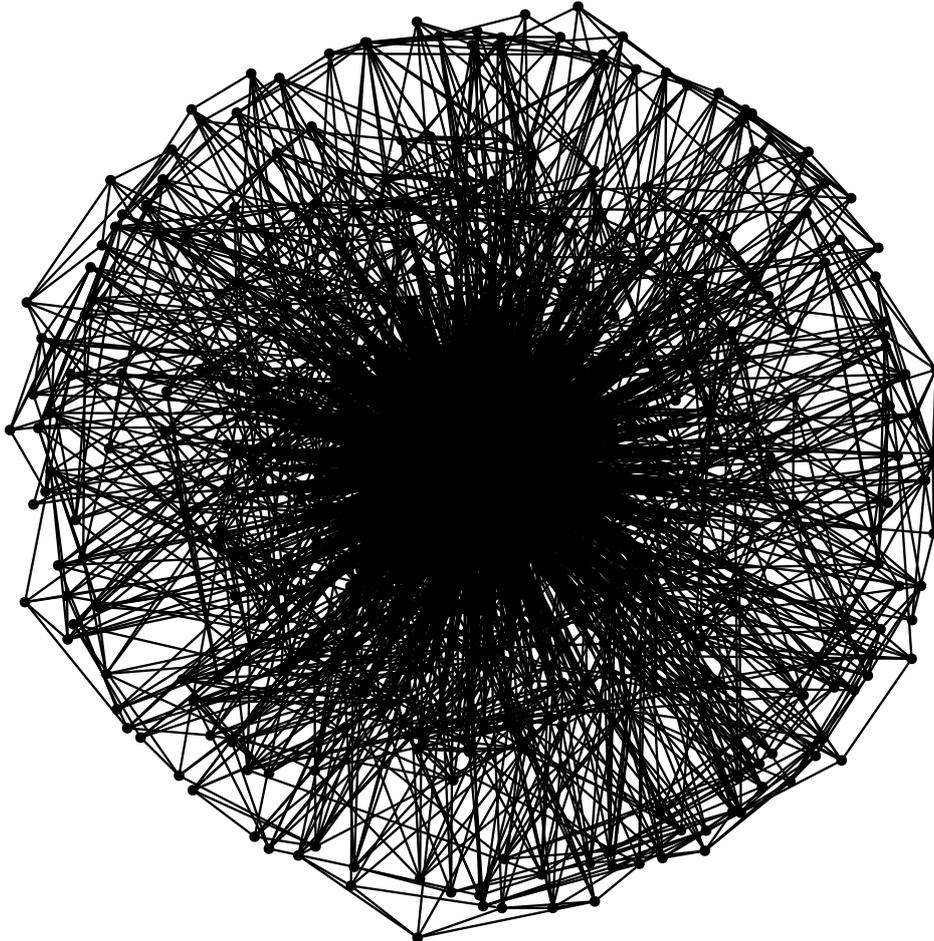


Abbildung 8.5: Graph nach Sim-1. Im Laufe der Simulation wird in jedem Programmmlauf ein Graph erzeugt. Die Abbildung zeigt einen dieser Graphen.

## Sim-2

In der zweiten Variante dieser Simulation wurde das soziale Netzwerk als Graph nach Barabási und Albert (1999) modelliert. Eine graphische Darstellung des Graphen ist in Abbildung 8.6 zu sehen.

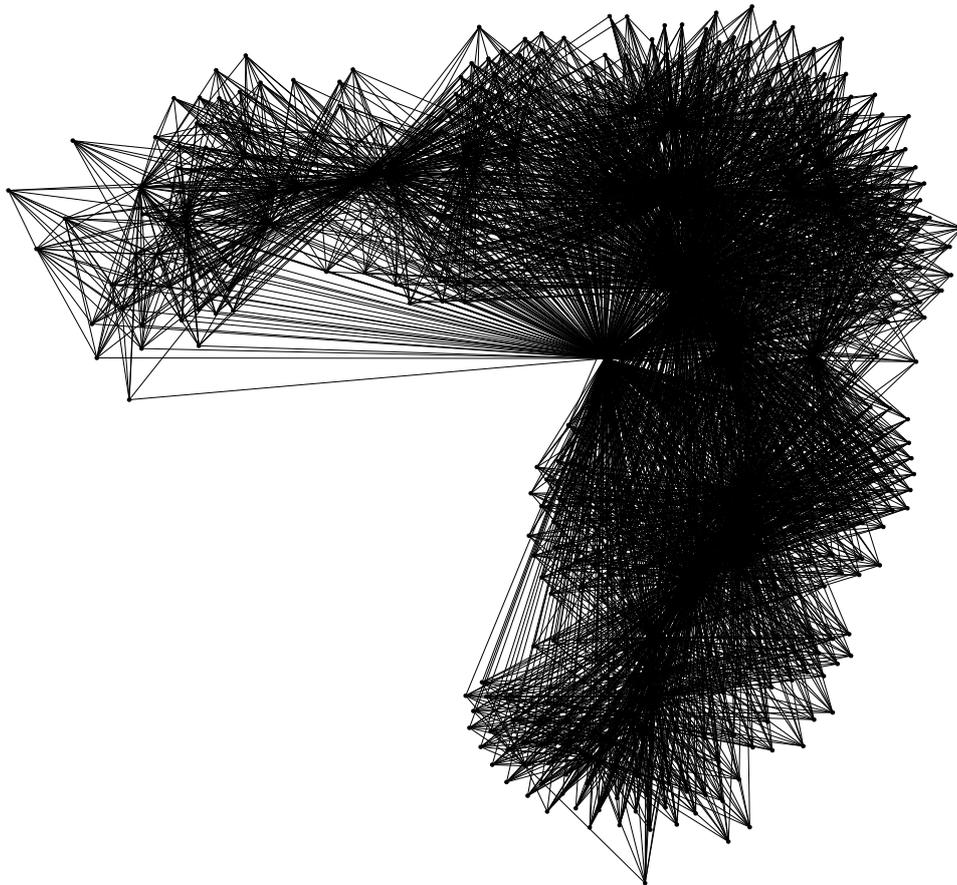


Abbildung 8.6: Initialisierter Barabási-Albert-Graph.

Auch in diesem Fall gilt, dass eine solche bildliche Darstellung eines Graphen schwer zu interpretieren ist. Der qualitative Unterschied zu einem Graphen nach

Newman-Watts ist aber zu erkennen. Der Barabási-Albert-Graph ist zwar ebenfalls zusammenhängend, aber nicht so stark vernetzt wie der Graph aus Abbildung 8.1. Die Verteilung der Knotengrade folgt einem Potenzgesetz, nach dem die Wahrscheinlichkeit, mit der ein Knoten den Grad  $k$  hat, proportional zum Kehrwert von  $k$  ist. Dementsprechend haben nur wenige Knoten einen hohen, aber viele einen niedrigen Grad. Abbildung 8.7 zeigt die Verteilung der Knotengrade vor der Simulation. Das Histogramm zeigt die typische Form der *Power-Law*-Verteilung.

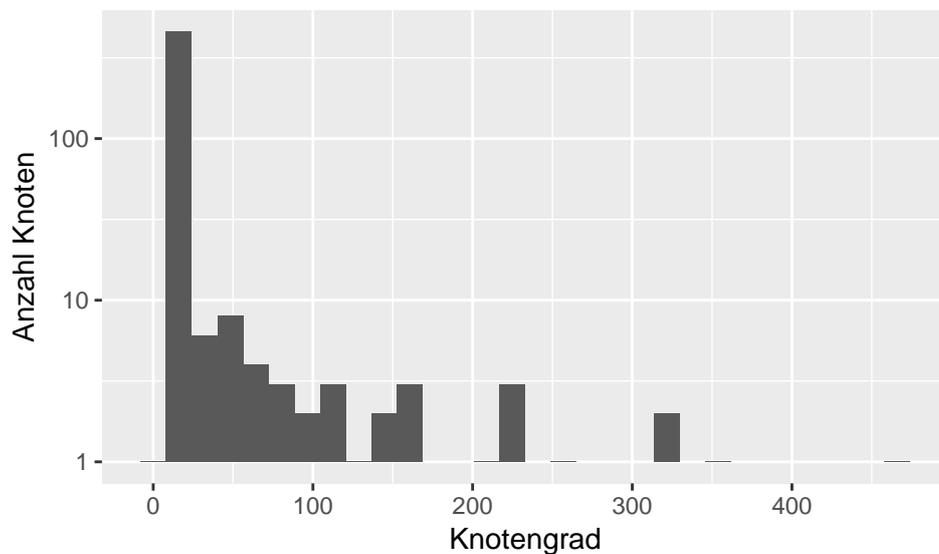


Abbildung 8.7: Verteilung der Knotengrade vor Sim-2. Die Häufigkeit, mit der ein Knotengrad  $k$  angenommen wird, ist proportional zum Kehrwert von  $k$ .

Abbildung 8.8 zeigt die Verteilung des Gutes vor und nach der Simulation. Wie in Sim-1 ist auch hier die unterschiedliche Skalierung der  $x$ -Achsen zu beachten. Vor der Simulation ist das Gut gleichmäßig zwischen 0 und 1 verteilt. Nach der Simulation ergibt sich ebenfalls eine relativ gleichmäßige Verteilung, diesmal aber über einem deutlich größeren Intervall. Auch in Sim-2 hat sich die Verteilung nur wenig hinsichtlich ihrer Form, dafür aber hinsichtlich der Skalierung verändert. Während der Maximalwert vor der Simulation bei 1 lag, liegt er danach aggregiert über alle Simulationen bei 1 Mio.

Die Verteilung der Knotengrade nach der Simulation, aggregiert über alle Simulationen, ist in Abbildung 8.9 zu sehen. Auch in der zweiten Simulation zeigt

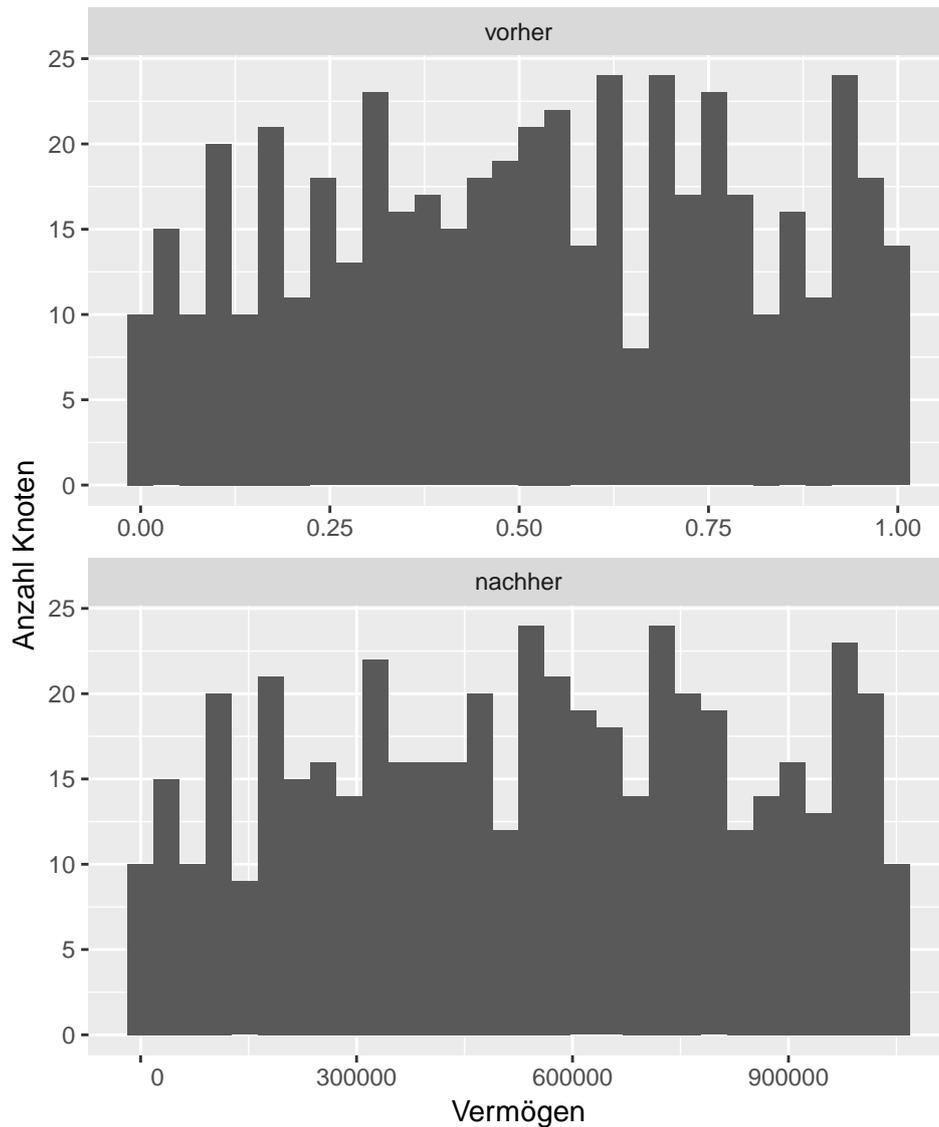


Abbildung 8.8: Verteilung des Guts vor und nach Sim-2. Der obere Teil der Abbildung zeigt die Verteilung des Guts vor der Simulation, es ist gleichmäßig verteilt zwischen 0 und 1. Der untere Teil zeigt die Verteilung nach der Simulation, aggregiert über alle Simulationen. Die Verteilung ist ebenfalls gleichmäßig, allerdings über einem Intervall von 0 bis 1000000.

sich ein deutlicher Unterschied gegenüber der Verteilung vor der Simulation. Auch hier hat die Verteilung eine Spitze am unteren Ende der Skala, insgesamt sind die Knotengrade aber relativ gleichmäßig über den gesamten Bereich verteilt.

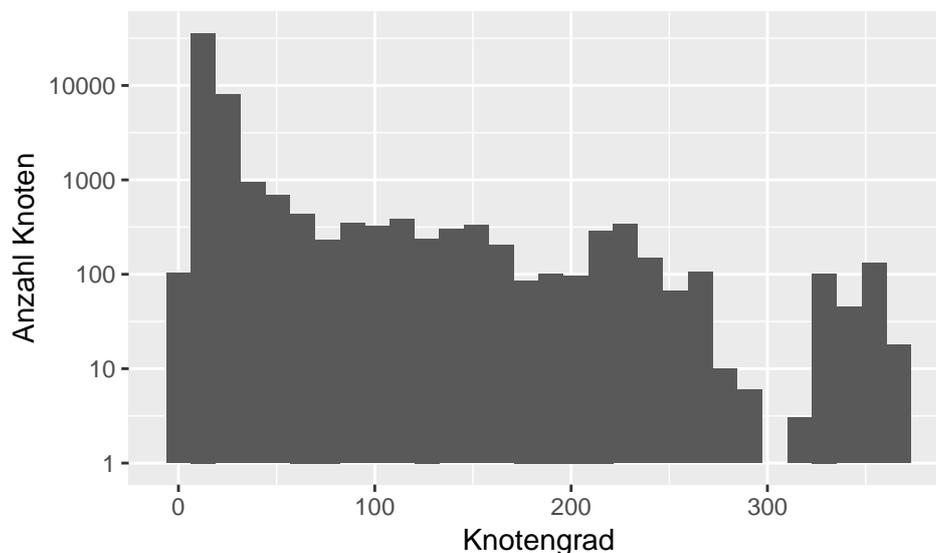


Abbildung 8.9: Verteilung der Knotengrade nach Sim-2, aggregiert über alle Programmläufe.

### 8.3 Diskussion

Das Modell, das in diesem Kapitel vorgestellt und in Sim-1 bzw. Sim-2 umgesetzt wurde, unterscheidet sich von den anderen Modellen in dieser Arbeit vor allem darin, dass ein dynamischer Zusammenhang zwischen zwei sozialen Phänomenen hergestellt wird, die beide für sich schon Risiken bergen: soziale Ungleichheit, die in diesem Fall durch eine Art von Verteilungsungerechtigkeit zustande kommt, und die Veränderung der Struktur eines sozialen Netzwerks, die von den sozialen Akteuren selbst auf Basis eben dieser Ungleichheit vorangetrieben wird. Das Modell ist, das muss klar gesagt werden, rudimentär. Aus diesem Grund sollten auch die Ergebnisse nicht überbewertet werden. Es war allerdings auch nicht das primäre Ziel des Autors, ein realistisches Modell zu entwickeln. Es ging vielmehr darum, die Möglichkeiten der Probabilistischen Programmierung zur Modellierung von Systemischen Risiken auch im Zusammenhang mit vernetzten Systemen bzw. Risiken zu erproben. Dass es möglich sein würde, in einer vertretbaren Zeit ein verhältnismäßig komplexes Modell eines sozialen Netzwerks zu entwickeln, dessen Teilnehmer miteinander interagieren und dessen Struktur sich aufgrund dieser Interaktion dynamisch verändert, war nicht von vornherein klar und kann als ein erster Erfolg gewertet

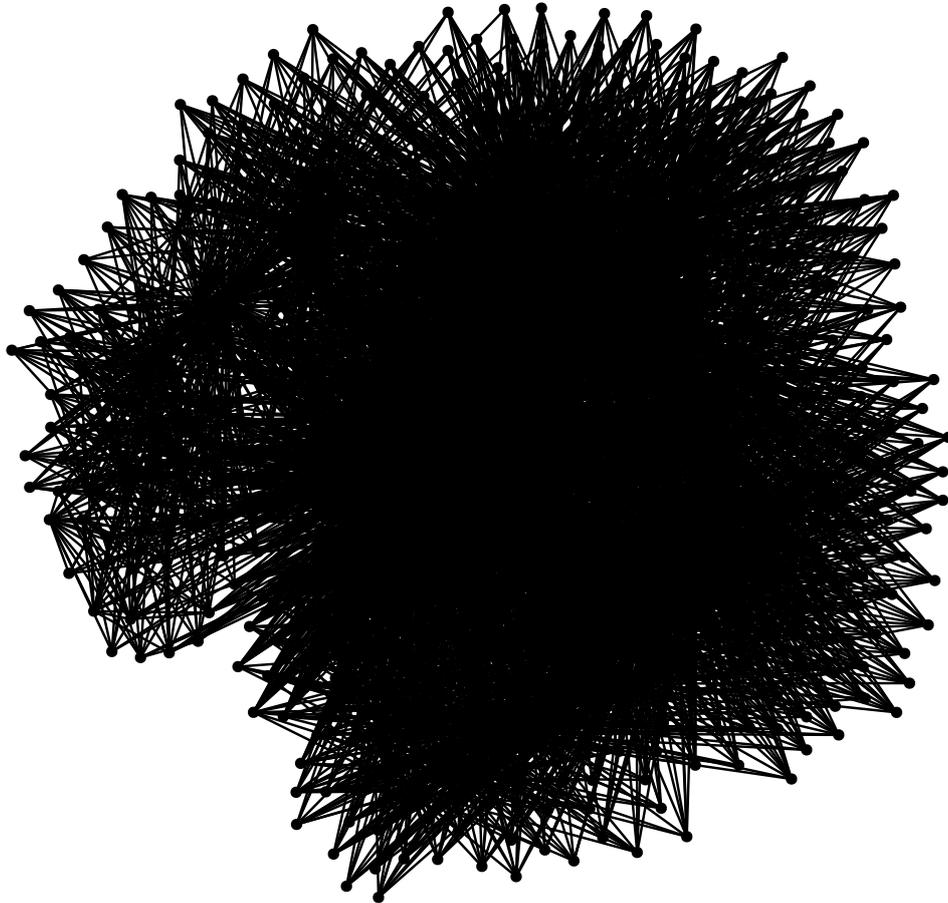


Abbildung 8.10: Graph nach Sim-2. Im Laufe der Simulation wird in jedem Programmmlauf ein Graph erzeugt. Die Abbildung zeigt einen dieser Graphen.

werden. Dieser explorative Charakter zeigt sich in verschiedenen Aspekten des Modells, in denen pragmatische Lösungen realistischeren Varianten vorgezogen wurden. Zukünftige Anstrengungen könnten und sollten sich daher unter anderem mit den nun folgenden Aspekten auseinandersetzen.

## Verteilung des Guts

Zunächst ist anzusprechen, dass die Verteilung des Guts “von oben” gesteuert wird. Weder die sozialen Akteure selbst noch die Topologie des Netzwerks haben einen Einfluss auf die Menge, die verteilt wird. Es überrascht daher auch nicht, dass die Verteilungen des Guts vorher und nachher sich in Sim-1 und Sim-2 nicht wesentlich unterscheiden. Dennoch ist der Unterschied im Spektrum dieser Verteilungen hervorzuheben. Der Reichtum eines Knotens zu Beginn der Simulation liegt maximal bei 1, während er nach der Simulation auf bis zu 1 Mio. gestiegen sein kann. Dieses exponentielle Wachstum ist durch gerade diese Dynamik des Modells begründet, nach welcher der Zuwachs ausschließlich vom Besitz abhängt. Auch die Fixierung der Varianz des Zuwachs ist Streitbar. Ein realistischeres Modell würde hier auch andere, evtl. antagonistische Einflussgrößen für den Zuwachs des Reichtums berücksichtigen. Ein weiterer Grund für das enorme Wachstum ist auch die Tatsache, dass das Modell nur einen Zuwachs und niemals auch einen Verlust des Guts vorsieht. Wenn man das Gut, wie vorgeschlagen, als monetäre Größe auffasst, dann erscheint es notwendig, auch die Ausgabenseite zu betrachten.

Es sei aber darauf hingewiesen, dass die Größenordnung nicht so unrealistisch ist, wie sie auf den ersten Blick erscheinen mag. Stellt man sich das Gut tatsächlich als Geld vor, das in Euro verteilt wird, so postuliert das Modell ein Wachstum von - im günstigsten Fall - ca. 1 Euro auf ca. 1 Mio. Euro in 20 Zeitschritten dar. Stellt man sich die Zeitschritte nun weiter als Jahre vor, dann gibt es also einige Akteure des Netzwerks, die innerhalb dieser Zeit gewissermaßen vom Tellerwäscher zum Millionär werden. Nun ist es fraglich, ob es tatsächlich möglich ist, mit einem so geringen Startkapital so weit zu kommen. Genauso fraglich ist es aber, zumindest in einer Industrienation wie Deutschland, ob es Personen gibt, die mit nur einem Euro an den Start gehen. Wenn das Ausgangskapital größer ist, dann kann es auch bei geringerem Wachstum zu einem starken Zuwachs kommen.

Nicht realistisch ist aber sicherlich das Ergebnis, dass die Verteilung des Reichtums zwischen den extremen Polen ungefähr gleich verteilt ist. Zwar ist bekannt, dass es - wieder in Deutschland - einige Personen gibt, deren Reichtum noch deutlich über einer Million liegt. Diese Gruppe ist aber zahlenmäßig relativ klein. Dass das Modell diese Zahl so hoch schätzt, liegt daran, dass der Reichtum sich ungehin-

dert und nur durch sich selbst vermehren kann. Hier ist es notwendig, wie oben schon diskutiert, weitere Einflussgrößen auszumachen und in das Modell mit aufzunehmen.

Die gute Nachricht ist, dass sich diese und andere Punkte mit einem vertretbaren Aufwand in das Modell “einbauen” ließen. Das Modell wurde modular aufgebaut, die einzelnen Mechanismen wurden separat in Form von Funktionen implementiert und zusammengefügt. Das Ändern dieser Funktionen, genau wie das Hinzufügen von neuen Bausteinen, sind typische Tätigkeiten in der Entwicklung von Software und stellen per se kein besonderes Hindernis dar.

### Verteilung der Knotengrade

Ein wesentlicher Aspekt des Modells betrifft die Veränderung der Struktur eines vorgegebenen Graphen. Die Vermutung, dass die ursprünglich vorhandene Struktur einen Einfluss auf die Art der Veränderung hat, liegt nahe. Aus diesem Grund wurden in Sim-1 und Sim-2 Graphen unterschiedlicher Topologien zum Ausgangspunkt des Modells gemacht. Sowohl der Newman-Watts-Graph aus Sim-1, als auch der Barabási-Albert-Graph aus Sim-2 sind sogenannte Small-World-Netzwerke. Für beide Typen von Netzwerken gibt es eine ganze Reihe von Forschungsarbeiten, die die Angemessenheit dieser Graphen für verschiedene real existierende Netzwerke untersuchen (für einen Überblick siehe Watts, 2004b). Dabei wurden im Laufe der Zeit vor allem Beispiele gefunden, die auf eine Ähnlichkeit mit der Struktur des Barabási-Albert-Graphen hinweisen. Die Verwendung dieser Graphen erschien also berechtigt, zumal das Modell, wie schon erwähnt, einen eher explorativen Charakter hat. Letztlich ist man aber an der Modellierung realer Netzwerke interessiert, für die es bisher noch kein befriedigendes Modell gibt. Wie beispielsweise der soziale Graph für die Bevölkerung in Deutschland aussieht, ist nicht bekannt. Erschwerend kommt hinzu, dass Prozesse wie die aus dem Modell ständig ablaufen, so dass sich dieser Graph stetig verändert. Es wird daher vielleicht noch eine ganze Weile notwendig sein, auf idealisierte Netzwerke zurückzugreifen.

Dass die Topologie einen großen Einfluss hat, zeigt sich an den Ergebnissen bezüglich der Verteilung der Knotengrade. Während in Sim-1 die Ungleichheit in Bezug auf die Anzahl der Verbindungen zu anderen Knoten durch den simulierten Prozess größer wird, zeigt sich in Sim-2 ein anderer Effekt. Hier gibt es deutlich mehr Kno-

ten, die im Laufe der Simulation die Anzahl ihrer Verbindungen steigern konnten. Das ist auch insofern interessant, als die Verteilung der Knotengrade in Sim-2 vor der Simulation ungleicher ist als die in Sim-1. Worin dieses Ergebnis begründet ist, ist nicht klar. Es zeigt aber sehr schön eine Eigenschaft von Prozessen in komplexen und dynamischen Systemen, nämlich dass sie Situationen hervorbringen können, die intuitiv nicht unbedingt verständlich sind.

# Kapitel 9

## Diskussion

Die verschiedenen Kapitel der vorliegenden Arbeit beschäftigen sich mit Systemischen Risiken aus zwei verschiedenen Perspektiven. Die große Frage, die im ersten Teil der Arbeit gestellt wird, ist die nach dem Wesen Systemischer Risiken. Was genau ist ein Systemisches Risiko? Die Literaturrecherche, deren Ergebnisse in Kapitel 2 präsentiert wurden, hat offenbart, dass diese Frage nicht ohne Weiteres zu beantworten ist. Weder inter- noch intradisziplinär gibt es einen Konsens dahingehend, was genau Systemische Risiken ausmacht. Ein Ziel dieser Arbeit ist es, in diesem Punkt Abhilfe zu schaffen und einen Vorschlag für ein Rahmenmodell zu machen, das abstrakt genug ist, um in den unterschiedlichen Disziplinen angewendet werden zu können und dabei trotzdem die wesentlichen Aspekte berücksichtigt. Nach Meinung des Autors wurde dieses zugegebenermaßen ambitionierte Ziel erreicht. Der Autor ist andererseits nicht so naiv, zu glauben, dass zukünftige Leser hier nicht anderer Meinung sein können. Denn auch wenn die Identifizierung der wesentlichen Merkmale auf Basis einer sorgfältigen Literaturrecherche erfolgt ist, handelt es sich letztlich um eine subjektive Bewertung. Kritik ist deshalb nicht nur gut vorstellbar, sondern auch willkommen.

Ein wichtiger Schritt auf dem Weg zu einem Rahmenmodell war die Unterscheidung verschiedener Betrachtungsebenen, die in Kapitel 4.1.1 eingeführt wurde. Vor allem die Unterscheidung der Ebene theoretischer Abstraktion und der Ebene formaler Repräsentationen hat bei der Integration der Ansätze geholfen. Auf der Ebene theoretischer Abstraktion werden die zu erforschenden Phänomene zunächst ab-

strakt und losgelöst vom Kontext beschrieben. Diese Formulierung stellt eine qualitative Auseinandersetzung mit dem Forschungsgegenstand dar, der die wesentlichen Merkmale hervorbringen soll. Erst danach, in einem weiteren Schritt, kann man beginnen, nach einer geeigneten Formalisierung zu suchen. Eine Kopplung von abstrakten Konzepten an konkrete Realisierungen sollte grundsätzlich vermieden werden.

Es ist daher nur folgerichtig, dass das Rahmenmodell auf der Ebene theoretischer Abstraktion angesiedelt ist. Der Schlüssel zur fächerübergreifenden Anwendung liegt gerade in der abstrakten Sichtweise. Das Rahmenmodell verbindet drei Merkmale: Globalität, Rekursivität und Dynamik. Global ist ein Systemisches Risiko in dem Sinn, dass das ganze System betroffen ist, über das gesprochen wird. Innerhalb dieses Systems, dessen Bestandteile natürlich zunächst festgelegt werden müssen, gibt es mindestens ein rekursives Element, das sich über die Zeit hinweg selbst beeinflusst. Auch das Vorliegen mehrerer Elemente, die sich gegenseitig rekursiv beeinflussen, ist denkbar. In diesem Fall ist das rekursive Element selbst ein System. Die Dynamik ergibt sich dann aus diesem rekursiven Mechanismus. Wenn der rekursive Mechanismus zu einer Entwicklung des Systems führen kann, die man als Zusammenbruch des Systems bezeichnen kann, dann spricht man von einem Systemischen Risiko.

Dieses Rahmenmodell ist sparsam gehalten und steht in Einklang mit den unterschiedlichen Konzeptionen von Systemischen Risiken, die bei der Literaturrecherche zu Tage traten. In diesem Sinne handelt es sich um ein integratives Rahmenmodell, was sicherlich positiv zu bewerten ist. Trotzdem ist auch Kritik angebracht. So bleibt etwa zu überlegen, ob die Redundanz, die durch die Betonung der Dynamik Einzug erhält, entfernt werden sollte, weil sie sich logisch aus der Rekursivität ergibt. Man könnte dann einfach von zwei wesentlichen Aspekten sprechen. Das ist auf der einen Seite wünschenswert, denn man hätte eine noch sparsamere Beschreibung, ohne das Modell beschneiden zu müssen. Auf der anderen Seite wird hier aber der Unterschied zu den konventionellen Risiken herausgestellt. Dies hervorzuheben erschien wichtig, denn ohne einen solchen Unterschied bräuchte man auch keine neue Klasse von Risiken. Aus diesem Grund wurde der Aspekt der Dynamik als eins der wesentlichen Merkmale Systemischer Risiken beibehalten.

Deutlich tiefer gehen andere Überlegungen. Ein paar Sätze weiter oben wurde vom Zusammenbruch des Systems gesprochen, aber was genau unter einem Zu-

sammenbruch des Systems zu verstehen ist, wird nicht weiter erläutert. Tatsächlich wurde diese Formulierung im Rahmenmodell selber auch vermieden. Dort wird stattdessen ausgeführt, dass man von einem Systemischen Risiko sprechen kann, wenn die Entwicklung, und damit auch die Veränderung des Systems, die an sich nicht schlimm, sondern ganz normal und zu erwarten ist, so stark wird, dass man nicht mehr vom gleichen System sprechen kann. Diese Formulierung wurde gewählt, um die Subjektivität des Begriffs des *Zusammenbruchs* zu vermeiden. Damit handelt man sich aber die Subjektivität an anderer Stelle ein. Jetzt muss nämlich entschieden werden, was es bedeutet, zu sagen, dass es sich zweimal um das gleiche System handelt. Da eine gewisse Veränderung als normal eingestuft wurde, kann hier nicht das vollständige Fehlen von Unterschieden gemeint sein.

Vielleicht liegt es an der abstrakten Sichtweise, dass das Rahmenmodell hier keine klare Antwort geben kann. Bei der Formulierung wurde schließlich die Benennung konkreter Sachverhalte sorgfältig vermieden. Vielleicht kommt hier aber auch ein normativer Aspekt Systemischer Risiken zum Ausdruck, der auch in der Definition konventioneller Risiken eine wichtige Rolle einnimmt (SRA, 2015). Dort heißt es, dass Risiken in Bezug auf etwas bestehen, das von Menschen als wertvoll erachtet wird. Wann etwas als wertvoll erachtet wird, wird nicht weiter erläutert. Und tatsächlich dürfte es schwierig sein, eine vollständige Liste zu erarbeiten. Hier eine Leerstelle zu lassen, erscheint deshalb als gangbarer Weg. Das bedeutet aber auch, dass unterschiedliche Akteure ganz unterschiedliche, vielleicht sogar konträre Ansichten haben können. Was dem einen als Risiko erscheint, ist für den anderen vielleicht eher eine Chance. Das gilt sowohl für konventionelle als auch für Systemische Risiken. Wann also ein System sich zu einem anderen System entwickelt hat, oder welche Entwicklung innerhalb eines Systems als akzeptabel oder inakzeptabel erachtet wird, ist nicht grundsätzlich zu entscheiden und muss im Diskurs ausgehandelt werden.

Zu beachten ist außerdem, dass im Rahmenmodell nicht festgelegt wird, was ein Systemisches Risiko ist. Stattdessen werden Merkmale benannt, bei deren Vorliegen von einem Systemischen Risiko gesprochen werden kann. Vielleicht sind es vor allem sprachliche Aspekte, die die Beantwortung dieser Frage so schwierig erscheinen lassen. So findet man etwa immer wieder die Formulierung, dass "der Klimawandel ein Systemisches Risiko ist". Bei näherer Betrachtung stellt sich aber die Frage, ob der

Klimawandel an sich nicht einfach ein Phänomen ist, das Risiken *birgt*, weil nämlich unter den möglichen Folgen des Klimawandels auch solche mit katastrophalen Auswirkungen sind. Diese Unterscheidung mag dem einen oder anderen haarspalterisch erscheinen. Für die seriöse wissenschaftliche Auseinandersetzung ist aber die Definition der verwendeten Begriffe unumgänglich. Der Autor folgt deshalb der Vorgehensweise der SRA (2015) und gibt die Bedingungen an, die im Fall von Systemische Risiken erfüllt sein müssen. Denn genau wie im Fall Systemischer Risiken fehlt schon für konventionelle Risiken eine einheitliche Definition, und zu einer allgemein akzeptierten Definition zu kommen, wird nicht als realistisch betrachtet. Eine übergeordnete qualitative Formulierung, die auf verschiedene Perspektiven anwendbar ist, wird aber sehr wohl als möglich und notwendig erachtet.

Die Notwendigkeit für ein Rahmenmodell wurde während der Beschäftigung mit dem deutlich, was den zweiten Teil der Arbeit ausmacht. Ursprünglich sollte die Arbeit vor allem methodischen Charakter haben und sich ganz auf die probabilistische Modellierung Systemischer Risiken konzentrieren. Um überhaupt zu entscheiden, was als Beispiel für ein Systemisches Risiko gehandelt werden kann, hat sich das Rahmenmodell als sehr wertvoll erwiesen. Die Entwicklung probabilistischer Modelle ist aus Sicht des Autors vor allem in den Sozialwissenschaften angezeigt, um dem epistemischen Status gerecht zu werden. Denn auch dort, wo Theorien vorliegen, ist die Unsicherheit in Bezug auf Aussagen sozialwissenschaftlicher Theorien stets vorhanden. Manche Autoren gehen deshalb so weit, die Phänomene selber als stochastisch zu betrachten (Renn et al., 2017), anstatt lediglich die Unwissenheit mit stochastischen Mittel zu beschreiben. Auch wenn dieser Standpunkt im hier vorgestellten Rahmenmodell nicht eingenommen wird, die Notwendigkeit probabilistischer Methoden wird vom Autor der vorliegenden Arbeit geteilt. Es erscheint deshalb als sinnvoll, diese Unsicherheit zu einem integralen Bestandteil bei der Modellbildung zu machen. Die Probabilistische Programmierung als Mittel zur Entwicklung solcher Modelle erschien als ausgezeichneter Kandidat und sollte an verschiedenen Beispielen ausprobiert werden.

Der Ansatz, Modelle nicht mathematisch, sondern mittels einer Programmiersprache zu spezifizieren, erscheint auch nach Fertigstellung der Arbeit vielversprechend. Die Ausdrucksmöglichkeiten, die man durch die Verwendung einer Probabilistischen Programmiersprache erhält, sind enorm. Alles, was man in Form eines

Computerprogramms “hinschreiben” kann, kann man ebenso als Probabilistisches Programm formulieren. Das erleichtert die Modellbildung gegenüber der mathematischen Modellierung deutlich. Zusätzlich gewinnt man Möglichkeiten aus der Software-Entwicklung dazu. So wird es möglich, ein Modell modular aufzubauen. Man kann so ein eventuell komplexes Modell auch langsam und Stück für Stück aufbauen und muss nicht die ganze Komplexität auf einmal bewältigen. Es ist außerdem möglich, für verschiedene Aspekte alternative Implementierungen zu entwickeln, die dann verglichen und evaluiert werden könnten.

Für letzteres können dann auch Daten hinzugezogen werden, sofern sie vorhanden sind. Dieser Punkt muss hervorgehoben werden, denn in diesem Fall stellt die Probabilistische Programmierung eine Verbindung von Simulation und Datenanalyse dar. Die Parameter des Modells können dann anhand der Daten “gelernt” werden. Ein Beispiel dafür wurde in Kapitel 6 gegeben. Obwohl dabei relativ wenige Datenpunkte zur Verfügung standen, hat der Einbezug von Daten die Ergebnisse in beeindruckender Art verbessert. Simulation und Lernen von Parametern sind mit der Probabilistischen Programmierung auch für non-parametrische Modelle möglich, bei denen die Anzahl der Parameter entgegen dem, was der Name vermuten lässt, theoretisch unendlich groß ist. Eine Demonstration bleibt die Arbeit schuldig. Aber gerade im Bezug auf die Ausbreitung von Influenza ist ein solcher Ansatz denkbar. Hier könnte zukünftige Forschung sich die Modularität zu Nutze machen und eine alternative Implementierung des Ansteckungsmechanismus entwickeln, die auf einem non-parametrischen Ansatz beruht.

Es sei aber auch betont, dass die Modellbildung via eines Probabilistischen Programms zwar vermutlich leichter von der Hand geht als die mathematische Modellierung, aber natürlich auch Schwierigkeiten mit sich bringt. Es ist eine Form der Programmierung. Auch wenn zumindest rudimentäre Programmierkenntnisse unter Sozialwissenschaftlern mittlerweile häufig anzutreffen sind: wer hier noch kein Vorwissen hat, dem stellt sich erstmal eine große Hürde. Es sollen also keine falschen Hoffnungen geweckt werden, wenn die Verwendung Probabilistischer Programmiersprachen für Systemische Risiken und in den Sozialwissenschaften allgemein hier propagiert werden. Die Entwicklung von formalen Modellen ist und bleibt auch mit dieser Methode eine anspruchsvolle Tätigkeit.

Was die Fallbeispiele angeht, die in dieser Arbeit präsentiert wurden, ist zunächst

zu sagen, dass sie, wie vermutlich alle sozialwissenschaftlichen Simulationen, als toy models zu bezeichnen sind. Es sind aber deutliche Unterschiede auszumachen, dahingehend, wieviel Realismus in diesen Modellen steckt. Am weitesten fortgeschritten ist das SIR-Modell aus Kapitel 6. Das liegt sicherlich daran, dass es eine lange Tradition der Forschung im Zusammenhang mit der Influnza gibt. Auf dieses Vorwissen konnte bei der Entwicklung des Modells zurückgegriffen werden. Trotzdem bleibt auch hier noch Raum zur Verbesserung. So wurde das Modell in dieser Arbeit global für eine Population erstellt, die als die deutsche Bevölkerung interpretiert werden kann. Es wäre wünschenswert, hier eine größere räumliche Auflösung einzuführen. Dabei würde es sich anbieten, zunächst auf die Ebene der Bundesländer zu gehen, da auf dieser Ebene auch Daten vom Robert-Koch-Institut gesammelt werden, die in das Modell einbezogen werden können. Aktuell wird im Rahmen eines Projekts im Robert-Koch-Institut daran gearbeitet, Abrechnungsdaten der Krankenkassen aufzubereiten und öffentlich zugänglich zu machen. Wenn solche Daten vorliegen, dann wäre eine noch feinere räumliche Auflösung möglich.

Die anderen zwei Beispiele sind als deutlich weniger realistisch einzustufen. Im Gegensatz zum SIR-Modell werden hier aber Netzwerke als Ausgangspunkt der Simulation benutzt. Gerade im Bezug auf soziale Phänomene sind solche Strukturen von großem Interesse. Es ist also positiv zu sehen, dass es mit der Probabilistischen Programmierung möglich ist, auch solche Modelle zu entwickeln. Es ist außerdem möglich, dass sich die Netzwerke im Laufe der Simulation verändern. Gerade dieser Punkt ist interessant, trifft er doch auf echte soziale Netzwerke zu. Die Annahmen, die in die Modelle eingehen, sind stark vereinfacht und als nicht besonders realistisch einzuschätzen. Trotzdem sind die Ergebnisse interessant. So hat sich gezeigt, dass der Ausgangsgraph für die Art und Weise, in der sich das Netzwerk in 8 verändert, keine große Rolle spielt. Dieser Befund ist nicht intuitiv verständlich, eine genauere Untersuchung könnte wertvolle Impulse für weitere Forschung liefern. Und es gibt viele Möglichkeiten, diese Modelle etwas realistischer zu gestalten. Durch die Möglichkeit des modularen Aufbaus eines Probabilistischen Programms ist eine nachträgliche Veränderung diesbezüglich gut möglich. In Kapitel 7 wurde bei der Ausbreitung von Fake News für jeden Teilnehmer des Netzwerks der gleiche Mechanismus für die Übernahme der Falschmeldung angegeben. Dies ist offensichtlich unrealistisch. Eine mögliche Weiterentwicklung des Modells besteht darin, mehrere unterschied-

liche Mechanismen zu spezifizieren. Welche Mechanismen hier in Frage kommen, könnte evtl. in einem qualitativen Forschungsprojekt, z.B. unter Verwendung von Interviews, erforscht werden.

Abschließend kann man sagen, dass die theoretischen Überlegungen zum Wesen Systemischer Risiken, die in dieser Arbeit angestellt wurden, bei der Identifikation von Beispielen für Systemische Risiken hilfreich sein können. Die Notwendigkeit probabilistischer Modelle wurde betont und mit der Probabilistischen Programmierung wurde ein Mittel zur Entwicklung solcher Modelle vorgestellt. Der Autor hofft, dass zukünftige Forschung zumindest einige Aspekte dieser Arbeit aufnimmt und weiter entwickelt.



# Literatur

- Albert, R. [Reka] & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 47–97. arXiv: cond-mat/0106096. doi:[10.1103/RevModPhys.74.47](https://doi.org/10.1103/RevModPhys.74.47)
- Atkinson, A. C. (1979). The Computer Generation of Poisson Random Variables. *Applied Statistics*, 28(1), 29. doi:[10.2307/2346807](https://doi.org/10.2307/2346807)
- Aven, T. (2016). Risk assessment and risk management: Review of recent advances on their foundation. *European Journal of Operational Research*, 253(1), 1–13. doi:[10.1016/j.ejor.2015.12.023](https://doi.org/10.1016/j.ejor.2015.12.023)
- Bach, C., Bouchon, S., Fekete, A., Birkmann, J. & Serre, D. (2013). Adding value to critical infrastructure research and disaster risk management: The resilience concept. *S.A.P.I.EN.S. Surveys and Perspectives Integrating Environment and Society*, (6.1). Zugriff 2. Juni 2016 unter <http://sapiens.revues.org/1626>
- Balcan, D., Gonçalves, B., Hu, H., Ramasco, J. J., Colizza, V. & Vespignani, A. (2010). Modeling the spatial spread of infectious diseases: The GLObal Epidemic and Mobility computational model. *Journal of Computational Science*, 1(3), 132–145. doi:[10.1016/j.jocs.2010.07.002](https://doi.org/10.1016/j.jocs.2010.07.002)
- Barabási, A.-L. & Albert, R. [Réka]. (1999). Emergence of scaling in random networks. *science*, 286(5439), 509–512.
- Barabási, A.-L. & Bonabeau, E. (2003). Scale-free networks. *Scientific American*, 288(5), 50–59. Zugriff 15. Februar 2016 unter <http://cat.inist.fr/?aModele=afficheN&cpsidt=14745269>
- Beisbart, C. (2017). Advancing Knowledge Through Computer Simulations? a Socratic Exercise. In M. M. Resch, A. Kaminski & P. Gehring (Hrsg.), *The Science and Art of Simulation i* (S. 153–174). Springer.

- Bianchi, D., Billio, M., Casarin, R. & Guidolin, M. (2015). Modeling contagion and systemic risk. *Available at SSRN*.
- Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., ... Goodman, N. D. (2018). Pyro: Deep Universal Probabilistic Programming. *arXiv preprint arXiv:1810.09538*.
- Blasio, B. F. d., Svensson, Å. & Liljeros, F. (2007). Preferential attachment in sexual networks. *Proceedings of the National Academy of Sciences*, 104(26), 10762–10767. doi:[10.1073/pnas.0611337104](https://doi.org/10.1073/pnas.0611337104)
- Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(Supplement 3), 7280–7287. doi:[10.1073/pnas.082080899](https://doi.org/10.1073/pnas.082080899)
- Borsboom, D. (2008). Latent Variable Theory. *Measurement: Interdisciplinary Research & Perspective*, 6(1-2), 25–53. doi:[10.1080/15366360802035497](https://doi.org/10.1080/15366360802035497)
- Braun, N. & Saam, N. J. (Hrsg.). (2015). *Handbuch Modellbildung und Simulation in den Sozialwissenschaften*. Wiesbaden: Springer Fachmedien Wiesbaden. Zugriff 6. Juni 2016 unter <http://link.springer.com/10.1007/978-3-658-01164-2>
- Buzna, L., Peters, K. & Helbing, D. (2006). Modelling the dynamics of disaster spreading in networks. *Physica A: Statistical Mechanics and its Applications*. Information and Material Flows in Complex Networks Information and Material Flows in Complex Networks, 363(1), 132–140. doi:[10.1016/j.physa.2006.01.059](https://doi.org/10.1016/j.physa.2006.01.059)
- Byers, J. W., Mitzenmacher, M. & Zervas, G. (2012). Daily deals: Prediction, social diffusion, and reputational ramifications. In *Proceedings of the fifth ACM international conference on Web search and data mining* (S. 543–552). ACM.
- Cartwright, N. (1994). *Abstract and Concrete*. Oxford University Press. Zugriff 16. Juli 2018 unter <http://www.oxfordscholarship.com/view/10.1093/0198235070.001.0001/acprof-9780198235071-chapter-6>
- Chowell, G., Miller, M. A. & Viboud, C. (2008). Seasonal influenza in the United States, France, and Australia: Transmission and prospects for control. *Epidemiology and Infection*, 136(6), 852–864. doi:[10.1017/S0950268807009144](https://doi.org/10.1017/S0950268807009144)
- Coburn, B. J., Wagner, B. G. & Blower, S. (2009). Modeling influenza epidemics and pandemics: Insights into the future of swine flu (h1n1). *BMC Medicine*, 7, 30. doi:[10.1186/1741-7015-7-30](https://doi.org/10.1186/1741-7015-7-30)

- Colizza, V., Barrat, A., Barthélemy, M. & Vespignani, A. (2006). The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences of the United States of America*, 103(7), 2015–2020. doi:10.1073/pnas.0510525103
- Davis, E. P. (2003). Towards a typology for systemic financial instability. Zugriff 24. Oktober 2016 unter <http://bura.brunel.ac.uk/handle/2438/916>
- Davoudi, A. & Chatterjee, M. (2015). Probabilistic spreading of recommendations in social networks. In *Military Communications Conference, MILCOM 2015-2015 IEEE* (S. 1373–1378). IEEE.
- De Bandt, O. & Hartmann, P. (2000). *Systemic risk: A survey* (European Central Bank Working Paper Series Nr. 35). Zugriff 24. Oktober 2016 unter [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=258430](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=258430)
- Dezsó, Z. & Barabási, A.-L. (2002). Halting viruses in scale-free networks. *Physical Review E*, 65(5). doi:10.1103/PhysRevE.65.055103
- Diebold, F. X. & Yilmaz, K. (2014). On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics. Causality, Prediction, and Specification Analysis: Recent Advances and Future Directions*, 182(1), 119–134. doi:10.1016/j.jeconom.2014.04.012
- Dörner, D. (1989). *Die Logik des Misslingens: Strategisches Denken in komplexen Situationen*.
- Dwyer, G. (2009). What is systemic risk, anyway? *Federal Reserve Bank of Atlanta Macroblog*, 6. Zugriff unter <https://macroblog.typepad.com/macroblog/2009/11/what-is-systemic-risk-anyway.html>
- Erdos, P. & Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5(1), 17–60.
- Ferguson, R. W., finance ministry and central bank staff from Australia, Belgium, Canada, France, Germany, Italy, Japan, the Netherlands, Spain, Sweden, Switzerland, the United Kingdom and the United States, finance ministry and central bank staff from Australia, representatives from the Bank for International Monetary Fund and the Organisation for Economic Co-operation and Development. (2001). *Report on consolidation in the financial sector*. Bank for International Settlements, Monetary and Economic Department. OCLC: 782989982. Basle.

- Forbes, K. J. & Rigobon, R. (2001). Measuring Contagion: Conceptual and Empirical Issues. In *International Financial Contagion* (S. 43–66). Boston, MA: Springer US.
- Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A. & Rubin, D. B. (2013). *Bayesian data analysis*. Chapman und Hall/CRC.
- Georg, C.-P. (2013). The effect of the interbank network structure on contagion and common shocks. *Journal of Banking & Finance*, 37(7), 2216–2228. doi:[10.1016/j.jbankfin.2013.02.032](https://doi.org/10.1016/j.jbankfin.2013.02.032)
- Gladwell, M. (2006). *The tipping point: How little things can make a big difference*. Little, Brown.
- Global Risk Network. (2006). World economic forum global risk report 2006. In *World Economic Forum, Geneva*.
- Goldin, I. & Vogel, T. (2010). Global Governance and Systemic Risk in the 21st Century: Lessons from the Financial Crisis: Governance and Systemic Risk. *Global Policy*, 1(1), 4–15. doi:[10.1111/j.1758-5899.2009.00011.x](https://doi.org/10.1111/j.1758-5899.2009.00011.x)
- Goodman, N. D., Mansinghka, V., Roy, D. M., Bonawitz, K. & Tenenbaum, J. B. (2012). Church: A language for generative models. *arXiv preprint arXiv:1206.3255*. Zugriff 24. Oktober 2016 unter <http://arxiv.org/abs/1206.3255>
- Goodman, N. D. & Stuhlmüller, A. (2014). *The design and implementation of probabilistic programming languages*. Retrieved 2015/1/16, from <http://dippl.org>.
- Gordon, A. D., Henzinger, T. A., Nori, A. V. & Rajamani, S. K. (2014). Probabilistic programming. In *Proceedings of the on Future of Software Engineering* (S. 167–181). ACM. Zugriff 25. September 2017 unter <http://dl.acm.org/citation.cfm?id=2593900>
- Gottschalk-Mazouz, N. (2012). Toy Modeling: Warum gibt es (immer noch) sehr einfache Modelle in den empirischen Wissenschaften. In P. Fischer & A. Luckner (Hrsg.), *Reflexion des Möglichen* (S. 17–30). Berlin: LIT. Zugriff 22. Mai 2017 unter [https://mail.hlrs.de/service/home/~/?auth=co&loc=en\\_US&id=5046&part=4](https://mail.hlrs.de/service/home/~/?auth=co&loc=en_US&id=5046&part=4)
- Haldane, A. G. & May, R. M. (2011). Systemic risk in banking ecosystems. *Nature*, 469(7330), 351–355. doi:[10.1038/nature09659](https://doi.org/10.1038/nature09659)
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications.

- Helbing, D. (2012). Systemic risks in society and economics. In *Social Self-Organization* (S. 261–284). Springer. Zugriff 3. Februar 2016 unter [http://link.springer.com/chapter/10.1007/978-3-642-24004-1\\_14](http://link.springer.com/chapter/10.1007/978-3-642-24004-1_14)
- Helbing, D. (2013). Globally networked risks and how to respond. *Nature*, 497(7447), 51–59. doi:10.1038/nature12047
- Hendricks, D., Kambhu, J. & Mosser, P. (2007). Systemic Risk and the Financial System. *Economic Policy Review*, 13(2). Zugriff 14. Januar 2016 unter [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1027734](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1027734)
- Jäger, C. (2015). *CoEGSS - Center of Excellence for Global System Science*.
- Jones, M. R. (2005). Idealization and abstraction: A framework. *Idealization XII: Correcting the model. Idealization and abstraction in the sciences*, 86, 173–217.
- Karlova, N. A. & Fisher, K. E. (2013). Plz RT<sup>m</sup>: A social diffusion model of misinformation and disinformation for understanding human information behaviour. *Information Research*, 18(1), 1–17.
- Kaufman, G. G. & Scott, K. E. (2003). What is systemic risk, and do bank regulators retard or contribute to it? *The Independent Review*, 371–391. Zugriff 26. Oktober 2016 unter <http://www.jstor.org/stable/24562449>
- Kermack, W. O. & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Bulletin of mathematical biology*, 53(1/2), 33–55. Zugriff 22. Februar 2017 unter <http://rspa.royalsocietypublishing.org/content/royprsa/115/772/700.full.pdf>
- Kruschke, J. (2014). *Doing Bayesian data analysis: A tutorial with r, JAGS, and Stan*. Academic Press.
- Lucas, K., Jaeger, C. C. & Renn, O. (2017). *Systemische Risiken und Dynamische Strukturen - Eine konzeptionelle Ortsbestimmung*. IASS. Potsdam. Zugriff 24. April 2017 unter [http://www.iass-potsdam.de/sites/default/files/files/discussion\\_paper\\_17-03\\_systemische\\_risiken\\_dynamische\\_strukturen.pdf](http://www.iass-potsdam.de/sites/default/files/files/discussion_paper_17-03_systemische_risiken_dynamische_strukturen.pdf)
- Macal, C. M. & North, M. J. (2009). Agent-based modeling and simulation. In *Winter simulation conference* (S. 86–98). Winter simulation conference. Zugriff 10. Mai 2017 unter <http://dl.acm.org/citation.cfm?id=1995474>
- Mansinghka, V. (2015). Overview of Probabilistic Programming. Zugriff 27. August 2018 unter <https://www.youtube.com/watch?v=-8QMqSWU76Q>

- Mansinghka, V., Selsam, D. & Perov, Y. (2014). Venture: A higher-order probabilistic programming platform with programmable inference. *arXiv:1404.0099 [cs, stat]*. arXiv: 1404.0099. Zugriff 31. August 2018 unter <http://arxiv.org/abs/1404.0099>
- May, R. M., Levin, S. A. & Sugihara, G. (2008). Complex systems: Ecology for bankers. *Nature*, 451(7181), 893–895. Zugriff 12. August 2015 unter <http://www.nature.com/nature/journal/v451/n7181/full/451893a.html>
- McGrayne, S. B. (2014). *Die Theorie, die nicht sterben wollte: Wie der englische Pastor Thomas Bayes eine Regel entdeckte, die nach 150 Jahren voller Kontroversen heute aus Wissenschaft, Technik und Gesellschaft nicht mehr wegzudenken ist*. Springer Spektrum. Zugriff 25. September 2018 unter [//www.springer.com/de/book/9783642377693](http://www.springer.com/de/book/9783642377693)
- Meadows, D. H. & Wright, D. (2008). *Thinking in systems: A primer*. Chelsea Green Publishing. Zugriff 7. September 2015 unter [https://books.google.de/books?hl=de&lr=&id=CpbLAgAAQBAJ&oi=fnd&pg=PR9&dq=thinking+in+systems&ots=Lyn9q7xHU\\_&sig=f8O15KNqJ6MQmTLHwbC0Or69YZw](https://books.google.de/books?hl=de&lr=&id=CpbLAgAAQBAJ&oi=fnd&pg=PR9&dq=thinking+in+systems&ots=Lyn9q7xHU_&sig=f8O15KNqJ6MQmTLHwbC0Or69YZw)
- Neitzke, H.-P. (2007). *Systemische Risiken*. AACCrisk. Zugriff 7. September 2015 unter [http://www.ecolog-institut.de/fileadmin/user\\_upload/Publikationen/T\\_U\\_Publ/AACCrisk\\_Report\\_2007-2\\_SystemRisk.pdf](http://www.ecolog-institut.de/fileadmin/user_upload/Publikationen/T_U_Publ/AACCrisk_Report_2007-2_SystemRisk.pdf)
- Newman, M. E. J. & Watts, D. J. (1999). Scaling and percolation in the small-world network model. *Physical Review E*, 60(6), 7332–7342. arXiv: cond-mat/9904419. doi:10.1103/PhysRevE.60.7332
- OECD. (2003). *Emerging systemic risks in the 21st century: An agenda for action*. OECD. Paris.
- Paige, B. & Wood, F. (2014). A Compilation Target for Probabilistic Programming Languages. In *ICML (S. 1935–1943)*. Zugriff 26. September 2016 unter <http://www.jmlr.org/proceedings/papers/v32/paige14.pdf>
- Rainforth, T., Naesseth, C., Lindsten, F., Paige, B., Vandemeent, J.-W., Doucet, A. & Wood, F. (2016). Interacting Particle Markov Chain Monte Carlo. In *International Conference on Machine Learning (S. 2616–2625)*.
- Renn, O. (2005). *White Paper on risk governance: Towards and integrative approach*. International Risk Governance Council (IRGC). Zugriff 24. Oktober 2016 unter <http://lib.riskreductionafrica.org/handle/123456789/638>

- Renn, O. (2015). Risikowahrnehmung in der Bevölkerung – Implikationen für das Sicherheitsempfinden. *Zeitschrift für Außen- und Sicherheitspolitik*, 8(1), 49–67. doi:10.1007/s12399-014-0436-6
- Renn, O. (2016). Systemic Risks: The New Kid on the Block. *Environment: Science and Policy for Sustainable Development*, 58(2), 26–36. doi:10.1080/00139157.2016.1134019
- Renn, O., Jovanovic, A. & Schröter, R. (2011). *Social unrest*. OECD/IFP Project on “Future Global shocks. Zugriff 26. Oktober 2016 unter <http://www.oecd.org/futures/globalprospects/46890018.pdf>
- Renn, O. & Klinke, A. (2004). Systemic risks: A new challenge for risk management. *EMBO reports*, 5(S1), S41–S46. Zugriff 7. September 2015 unter <http://embor.embopress.org/content/5/1S/S41.abstract>
- Renn, O., Lucas, K., Haas, A. & Jaeger, C. (2017). Things are different today: The challenge of global systemic risks. *Journal of Risk Research*, 0(0), 1–15. doi:10.1080/13669877.2017.1409252
- Resch, M. M. (2017). On the Missing Coherent Theory of Simulation. In *The Science and Art of Simulation i* (S. 23–32). Springer.
- Rieck, T., Feig, M., Wichmann, O. & Siedler, A. (2017). Impfquoten der Rotavirus-, Masern-, HPV- und Influenza-Impfung in Deutschland. *Epidemiologisches Bulletin*, (1), 1–16. doi:DOI10.17886/EpidBull-2017-001
- RKI. (2015). *Gesundheit in Deutschland 2015*. Robert Koch-Institut. Berlin. Zugriff 18. Januar 2017 unter [http://www.rki.de/DE/Content/Gesundheitsmonitoring/Gesundheitsberichterstattung/GesInDtlD/gesundheit\\_in\\_deutschland\\_2015.pdf?\\_\\_blob=publicationFile](http://www.rki.de/DE/Content/Gesundheitsmonitoring/Gesundheitsberichterstattung/GesInDtlD/gesundheit_in_deutschland_2015.pdf?__blob=publicationFile)
- Rochet, J.-C. & Tirole, J. (1996). Interbank lending and systemic risk. *Journal of Money, credit and Banking*, 28(4), 733–762. Zugriff 24. Oktober 2016 unter <http://www.jstor.org/stable/2077918>
- Saam, N. J. (2015). Simulation in den Sozialwissenschaften. In N. J. Saam & N. Braun (Hrsg.), *Handbuch Modellbildung und Simulation in den Sozialwissenschaften* (S. 61–95). Springer.
- Schwarcz, S. L. (2008). Systemic risk. *Geo. LJ*, 97, 193.

- SRA. (2015). *Society for Risk Analysis Glossary*. Society for Risk Analysis. Zugriff 23. Juli 2018 unter [http://www.sra.org/sites/default/files/pdf/SRA\\_glossary\\_20150622.pdf](http://www.sra.org/sites/default/files/pdf/SRA_glossary_20150622.pdf)
- Strogatz, S. H. (2018). *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering*. CRC Press.
- Systemrisiko. (2018). Page Version ID: 180605529. Zugriff 21. November 2018 unter <https://de.wikipedia.org/w/index.php?title=Systemrisiko&oldid=180605529>
- Tinati, R., Carr, L., Hall, W. & Bentwood, J. (2012). Scale Free: Twitter's Retweet Network Structure.
- Tolpin, D., van de Meent, J. W., Yang, H. & Wood, F. (2016). Design and Implementation of Probabilistic Programming Language Anglican. *arXiv:1608.05263 [cs]*. arXiv: 1608.05263. Zugriff 29. Mai 2017 unter <http://arxiv.org/abs/1608.05263>
- Tolpin, D., van de Meent, J.-W., Paige, B. & Wood, F. (2015). Output-Sensitive Adaptive Metropolis-Hastings for Probabilistic Programs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (S. 311–326). Springer. Zugriff 14. September 2017 unter [http://link.springer.com/chapter/10.1007/978-3-319-23525-7\\_19](http://link.springer.com/chapter/10.1007/978-3-319-23525-7_19)
- Tolpin, D. & Wood, F. (2015). Maximum a Posteriori Estimation by Search in Probabilistic Programs. In *Proceedings of the Eighth Annual Symposium on Combinatorial Search* (S. 201–205).
- Travers, J. & Milgram, S. (1967). The small world problem. *Psychology Today*, 1(1), 61–67.
- Tversky, A. & Kahneman, D. (1981). The framing of decisions and the psychology of choice. *science*, 211(4481), 453–458.
- van de Meent, J.-W., Paige, B., Yang, H. & Wood, F. (2018). An Introduction to Probabilistic Programming. *arXiv:1809.10756 [cs, stat]*. arXiv: 1809.10756. Zugriff 9. Oktober 2018 unter <http://arxiv.org/abs/1809.10756>
- van de Meent, J.-W., Yang, H., Mansinghka, V. & Wood, F. (2015). Particle Gibbs with Ancestor Sampling for Probabilistic Programs. In *Proceedings of the 18th International conference on Artificial Intelligence and Statistics* (S. 986–994).

- Villazon-Terrazas, J., Aparicio, S. & Alvarez, G. (2015). Study on Twitter as a Complex Network. In *The Third International Conference on Digital Enterprise and Information Systems (DEIS2015)* (Bd. 54).
- Watts, D. J. (1999). Networks, dynamics, and the small-world phenomenon. *American Journal of sociology*, *105*(2), 493–527.
- Watts, D. J. (2004a). *Six degrees: The science of a connected age*. WW Norton & Company.
- Watts, D. J. (2004b). The “New” Science of Networks. *Annual Review of Sociology*, *30*, 243–270. Zugriff 12. August 2015 unter <http://www.jstor.org/stable/29737693>
- Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, *393*(6684), 440–442. doi:[10.1038/30918](https://doi.org/10.1038/30918)
- Weidemann, F., Renschmidt, C., Buda, S., Buchholz, U., Ultsch, B. & Wichmann, O. (2017). Is the impact of childhood influenza vaccination less than expected: A transmission modelling study. *BMC Infectious Diseases*, *17*(1). doi:[10.1186/s12879-017-2344-6](https://doi.org/10.1186/s12879-017-2344-6)
- Weisberg, M. (2007). Three kinds of idealization. *The journal of Philosophy*, *104*(12), 639–659.



# Anhang A

## Quellcode zu Kapitel 6

Der Quellcode aller Modelle ist auf mehrere sogenannte *namespaces* aufgeteilt. Namespaces stellen Clojures Möglichkeit dar, den Quellcode eines größeren Programms zu modularisieren. Anglican ist in Clojure geschrieben und folgt der Konvention dieser Sprache darin, pro namespace eine Datei zu verlangen. Die Dateien enden wie Clojure-Dateien auf `.clj`. Ein namespace enthält die Implementierung von Komponenten des Programms, die inhaltlich sinnvoll zusammengefasst werden können.

Das SIR-Modell auf Basis der Progressionen von Kohorten ist im namespace `two_stage_poisson` implementiert. Funktionen, die übergeordneter Natur sind und auch für alternative Ansteckungsmechanismen verwendet werden können, sind in `framework.clj` implementiert. Für das Modell wurde die Poisson-Verteilung neu implementiert, um die Laufzeit der Simulation zu verkürzen. Diese ist in `distributions` zu finden. Funktionen, die mit dem eigentlichen Modell inhaltlich nichts zu tun haben, aber notwendig sind, um den Fluss zwischen den Kompartimenten zu berechnen, wurden im namespace `dataflow` implementiert. Schließlich gibt es noch eine Datei `util.clj`, in der verschiedenen Hilfsfunktionen implementiert wurden, mit Hilfe derer die Ergebnisse in eine Datei geschrieben oder auch visualisiert werden können. Und der namespace `core` enthält Funktionen, die das Simulationsprogramm starten, die Ergebnisse sammeln und in eine Datei schreiben.

## Der Namespace two\_stage\_poisson

```
1 (ns sir-model.two-stage-poisson
2   (:require [sir-model.distributions :as d]
3             [sir-model.dataflow :as flow]
4             [sir-model.framework :as fw])
5   (:use [anglican [core :exclude [-main]] runtime emit]))
6
7
8
9 (defdist new-cases-dist
10  "Custom distribution to sample new infections from N infectious individuals
11  with parameter lambda."
12  [N lambda]
13  [*lambda (* N lambda)]
14  (sample* [this]
15    (cond
16      (zero? *lambda) 0
17      (> *lambda 30) (sample* (d/fast-poisson *lambda))
18      :else (sample* (poisson *lambda))))
19  (observe* [this value]
20    (if (pos? N)
21        (observe* (poisson *lambda) value)
22        0)))
23
24 (defdist two-stage-poisson-dist
25  "Custom distribution representing new cases in the two stage poisson model.
26  Expects number of primary and secondary cases and respective parameters as
27  inputs. sample returns a sample from this distribution. When given a number
28  of observed new infections, observe computes the log-likelihood of the given
29  number of infections. Wraps around fast-poisson and new-cases-dist."
30  [N-1 l-1 N-2 l-2]
31  [*l-1 (* N-1 l-1)]
```

```
32     *l-2 (* N-2 l-2)
33     *lambda (+ *l-1 *l-2)]
34 (sample* [this] (sample* (new-cases-dist 1 *lambda)))
35 (observe* [this value]
36         (if (pos? *lambda)
37             (observe* (poisson *lambda) value)
38             0)))
39
40
41 (with-primitive-procedures
42   [new-cases-dist]
43   (defm generate-poisson
44     [N lambda]
45     "Draws a sample from the distribution of new infections given parameter
46     lambda and number of already infected individuals."
47     (sample (new-cases-dist N lambda))))
48
49
50 (defm primary-poisson
51   "Samples number of primary cases at time t from a poisson distribution.
52   Primary cases are those that are generated by the already infected
53   people, given as the value for key :I for the current timestep. Returns
54   the map of compartments with updated number of primary cases for time t."
55   [t lambda coll]
56   (fw/infect t :I :primary #(generate-poisson % lambda) coll))
57
58
59 (defm secondary-poisson
60   "Samples the number of secondary cases at time t based on the number of
61   primary cases. See primary-poisson."
62   [t lambda coll]
63   (fw/infect t :primary :secondary #(generate-poisson % lambda) coll))
64
```

```
65
66 (with-primitive-procedures
67   [flow/a->b]
68   (defm remove-vaccinated
69     "Because some people are vaccinated, they will not get sick after
70     encountering a sick person. To account for this, some of those that
71     have been infected are now removed from the primary and secondary
72     compartemnts and are put back into the S compartment."
73     [t vac-rate coll]
74     (let
75       [rem-pri (sample (binomial (get-in coll [t :primary]) vac-rate))
76        rem-sec (sample (binomial (get-in coll [t :secondary]) vac-rate))]
77
78       (a->b [t :primary] [t :S] rem-pri
79            (a->b [t :secondary] [t :S] rem-sec coll))))
80
81
82
83 (defm start-poisson-poisson
84   "Start a cohort in two phases. First phase uses primary-poisson, second
85   phase uses secondary-poisson. Is really just a convenience wrapper around
86   those functions."
87   [t l-1 l-2 coll]
88   ((comp
89     #(remove-vaccinated t 0.2 %)
90     #(secondary-poisson t l-2 %)
91     #(primary-poisson t l-1 %))
92    coll))
93
94
95 (defm update-in-ang
96   "'Updates' a value in a nested associative structure, where ks is a sequence
97   of keys and f is a function that will take the old value and any supplied
```

---

```

98     args and return the new value, and returns a new nested structure.  If any
99     levels do not exist, hash-maps will be created."
100 [m ks f & args]
101 (let [up (fn up [m ks f args]
102         (let [[k & ks] ks]
103             (if ks
104                 (assoc m k (up (get m k) ks f args))
105                 (assoc m k (apply f (get m k) args))))))]
106     (up m ks f args)))
107
108
109 (defm progress
110 [t cases coll]
111
112 (if (= t (count coll))
113     ;; if time's up, remaining cases
114     ;; are discarded
115     coll
116
117     (let
118         [removed (sample (binomial cases 0.45))
119          remaining (- cases removed)
120          susceptible (get-in coll [(dec t) :S])
121
122          ;; during progression a number of things need to happen at each timestep
123          ;; some individuals recover, their number has to be added to [t :R]
124          ;; the rest remains infected, this number has to be added to [t :I]
125          ;; the number of susceptibles has to be copied from (dec t) to t
126          updated-coll (assoc-in
127                        (update-in-ang
128                          (update-in-ang coll [t :R] + removed)
129                          [t :I] + remaining)
130                        [t :S] susceptible)]

```

```
131
132     (progress (inc t) remaining updated-coll))))
133
134
135 (defm init-compartments
136     "Before the actual simulation, progression for already infected
137     individuals at time 0 must be run once. Necessary for correctness of the
138     number of individuals over the course of the simulation."
139     [coll]
140     (let
141         [initially-infected (get-in coll [0 :I])]
142         (progress 1 initially-infected coll)))
143
144
145 (with-primitive-procedures
146     [flow/cohort-size]
147     (defm start-and-progress
148         "Simulate formation and progression of a cohort."
149         [t l-1 l-2 coll]
150         ((comp
151             #(progress (inc t) (cohort-size t %) %)
152             #(start-poisson-poisson t l-1 l-2 %))
153          coll)))
154
155
156 (with-primitive-procedures [two-stage-poisson-dist flow/cohort-size flow/S->]
157     (defm start-observe-progress
158         "In case data is available, observe on it after start and before
159         progression of a cohort."
160         [t l-1 l-2 coll datapoint]
161         (let
162             [primary (generate-poisson datapoint l-1)
163              secondary (generate-poisson primary l-2)]
```

```

164
165     ;; updating coll
166     ;; primary and secondary cases have to be subtracted from :S
167     ;; and added to :primary and :secondary
168     temp (S-> t :primary primary coll)
169     updated-coll (S-> t :secondary secondary temp)]
170
171     (if (zero? t)
172         ;; start of simulation, no observing
173         (progress (inc t) (cohort-size t updated-coll) updated-coll)
174         ;; else observe
175         ;; how likely is the datapoint to have been generated by the primary
176         ;; and secondary cases from the previous timestep?
177         (let [prev-primary (get-in updated-coll [(dec t) :primary])
178               prev-secondary (get-in updated-coll [(dec t) :secondary])]
179             (observe
180              (two-stage-poisson-dist prev-primary l-1 prev-secondary l-2)
181              datapoint)
182             (progress (inc t) (cohort-size t updated-coll) updated-coll))))))
183
184
185 (with-primitive-procedures
186   [new-cases-dist]
187   (defm cohort-lifetime
188     "Represent a cohort through its entire lifetime. Take care of start and
189     progress and, if data is available, of observing the parameters. This
190     function needs to be passed to the season-fn to be recursively called for
191     as many timesteps as needed."
192     [t l-1 l-2 coll args]
193     (if (and
194         (:data args) (< t (count (:data args))))
195         (start-observe-progress t l-1 l-2 coll (get (:data args) t))
196         (start-and-progress t l-1 l-2 coll))))

```

```
197
198
199 (with-primitive-procedures
200   [flow/create-args-coll flow/cohort-size]
201   (defquery
202     two-stage-poisson-query
203     [args]
204
205     (let
206       [compartments (create-args-coll
207                       (:t-max args)
208                       (:compartments args)
209                       (:inits args))
210        initialized-comps (init-compartments compartments)
211
212        lambda-1 (sample (:prior-1 args))
213        lambda-2 (sample (:prior-2 args))
214
215        ;f #(start-and-progress %1 lambda-1 lambda-2 %2)
216        f #(cohort-lifetime %1 lambda-1 lambda-2 %2 args)
217
218        season (fw/season-fn 0 compartments f)]
219
220
221     {:lambda-1 lambda-1 :lambda-2 lambda-2 :season season}))))
```

## Der Namespace framework

```
1 (ns sir-model.framework
2   (:require [sir-model.dataflow :as flow])
3   (:use [anglican emit runtime [core :exclude [-main]]]))
4
5
```

```
6
7 (with-primitive-procedures
8   [flow/S->]
9   (defm infect
10     "Infections for time t. Infecting individuals from who infect new
11     individuals as specified by how (must be a function). Newly infected
12     individuals are subtracted from :S department. Returns updated-coll."
13     [t who whom how coll]
14     (let
15       [max-cases (get-in coll [t :S])
16        old-cases (get-in coll [t who])
17
18        ;; there cannot be more new cases than susceptibles
19        new-cases (if (pos? old-cases)
20                      (min (how old-cases) max-cases)
21                      0)]
22       (S-> t whom new-cases coll))))
23
24
25
26 (with-primitive-procedures
27   [flow/cohort-size]
28   (defm season-fn
29     "Generic function for simulating an influenza season. Takes a starting-
30     timestep, a collection of compartments and a lifetime-fn. timestep and
31     collection are arguments for lifetime-fn. This of course means that
32     lifetime-fn has to be a function that expects two arguments, the
33     timestep and and collection."
34     [t coll lifetime-fn]
35
36     (loop [t-cur t
37           c coll]
38
```

```
39         (if (= t-cur (count c))
40             c
41
42             (recur (inc t-cur)
43                  (lifetime-fn t-cur c))))))
```

## Der Namespace distributions

```
1 (ns sir-model.distributions
2   (:use [anglican core emit runtime]))
3
4 (defn geometric-recursion [p acc]
5   (if (sample* (flip p))
6       acc
7       (geometric-recursion p (inc acc))))
8
9 (defdist geometric
10  "Geometric distribution. Returns the number of misses up to and not including
11  the first success."
12  [p] []
13  (sample* [this] (geometric-recursion p 0))
14  (observe* [this value]
15    (let [unpacked (repeat value 0)]
16      (first (map #(observe* (bernoulli p) %) unpacked)))))
17
18
19
20 (defdist new-infections
21  "Distribution of new-infections. The number of new infections depends on the
22  number of already infected people. An infected individual transmits the
23  disease to some number of individuals. Parameters to this distribution are R0,
24  the basic reproduction number and the number of already infected individuals
25  (prevalence). R0 is conceptualized as a geometric random variable."
```

```

26 [RO already-infected] []
27 (sample* [this] (reduce + (repeatedly already-infected #(sample* (geometric (/ 1
28 (observe* [this value] (observe* (reduce + (repeatedly already-infected #(geometr
29
30
31 ; In Anglican werden die Implementierungen zum Ziehen von Zufallszahlen aus einer
32 ; gegebenen Verteilung des Apache ; Commons Math Projekts verwendet. Im Fall der
33 ; Poisson-Verteilung nimmt die dafür beanspruchte Zeit mit der Größe des
34 ; Parameter der Verteilung linear zu. Da dieser Parameter für das SIR-Modell
35 ; proportional zur Größe der betrachteten Population ist, führt sie zu sehr
36 ; langen Programmlaufzeiten. Eine alternative Implementierung ist daher
37 ; wünschenswert. Im folgenden wird dazu ein Algorithmus implementiert, der von
38 ; Atkinson (1979) beschrieben wurde.
39
40 (defn propose [lambda b a]
41   (let
42     [helper (fn [b a]
43               (let [u (sample* (uniform-continuous 0 1))]
44                 (/
45                   (-
46                     a
47                     (/
48                       (log (- 1 u))
49                       u))
50                   b)))]
51
52     (loop [x (helper b a)]
53       (if (>= (floor (+ x 0.5)) 0)
54           x
55           (recur (helper b a))))))
56
57
58 (defn sample-fast-poisson [lambda]

```

```
59 (let
60   ;; helper functions
61   beta-f (fn [lambda]
62            (/
63             Math/PI
64             (sqrt (* 3 lambda))))
65   alpha-f (fn [lambda b]
66            (* lambda b))
67
68   ;; constants
69   b (beta-f lambda)
70   a (alpha-f lambda b)
71   c (- 0.767 (/ 3.36 lambda))
72   k (- (- (log c) lambda) (log b))
73
74   ;; functions for rejection sampling
75   lhs (fn [x]
76         (let
77           [v (sample* (uniform-continuous 0 1))
78            y (- a (* b x))]
79            (+
80             y
81             (log
82              (/
83               v
84               (pow (+ 1 (exp y)) 2))))))
85
86   rhs (fn [n]
87         (-
88          (+ k (* n (log lambda)))
89          (log-gamma-fn (inc n))))]
90
91   ;; rejection sampling
```

```

92     (loop [x (propose lambda b a)]
93
94         (if (<= (lhs x) (rhs (floor (+ x 0.5))))
95             (int (floor (+ x 0.5)))
96
97             (recur (propose lambda b a))))))
98
99
100 (defdist fast-poisson
101     "This implementation uses an algorithm described by Atkinson (1979)
102     to sample from the poisson distribution. Runtime is independent of the
103     size of lambda, which is needed for the SIR-model."
104     [lambda]
105     []
106     (sample* [this]
107             (sample-fast-poisson lambda))
108     (observe* [this value] (observe* (poisson lambda) value)))

```

## Der Namespace dataflow

```

1 (ns sir-model.dataflow)
2
3
4 (defn create-args-coll
5     "Returns a datastructure holding the compartments for n timesteps. Takes an
6     int, a vector of keywords and optionally a map of key value pairs. Returns a
7     vector of n maps, where each map has all the the keywords in keys with initial
8     value of zero. If inits is supplied, the first entry in the returned vector
9     will associate the given values to the corresponding keys.
10
11     (create-args-coll 10 [:S :I :R] {:S 10000 :I 100})"
12     ([n keys]
13      (let [m (zipmap keys (repeat (count keys) 0))]]

```

```
14     (into [] (repeat n m))))
15   ([n keys inits]
16     (let [tail (create-args-coll (dec n) keys)
17           head (reduce #(assoc-in %1 [(key %2)] (val %2)) (first tail) inits)]
18       (into [head] tail))))
19
20
21 (defn cohort-size
22   "Number of individuals for cohort starting from time t."
23   [t coll]
24   (+
25     (get-in coll [t :primary])
26     (get-in coll [t :secondary])))
27
28
29 (defn a->b
30   "Takes a map m and two vectors a and b with keys, subtracts value from a and
31   adds it to b."
32   [a b value m]
33   (-> m
34     (update-in ,,, a #(- % value))
35     (update-in ,,, b #(+ % value))))
36
37
38 (defn S->
39   "Takes a map of compartments, subtracts value from :S at time t and adds it
40   to to at time t."
41   [t to value coll]
42   (a->b [t :S] [t to] value coll))
43
44
45 (defn S->primary [t value coll]
46   (a->b [t :S] [t :primary] value coll))
```

```
47
48
49 (defn S->secondary [t value coll]
50   (a->b [t :S] [t :secondary] value coll))
51
52
53 (defn ->coll
54   "Adds value to compartments coll. target is a keyseq for determining where to
55   add value."
56   [target value coll]
57   (update-in coll target + value))
58
59
60 (defn ->compartments
61   "Adds multiple values to corresponding compartments. args needs to be a
62   sequence of vector nested keys and corresponding values. Returns updated
63   coll with each value added to the corresponding compartment."
64   [targets-and-vals coll]
65   (let [pairs (partition 2 targets-and-vals)]
66     (reduce #(->coll (first %2) (second %2) %1) coll pairs)))
67
68
69 (defn update-rules
70   "Given the number of cases and the number of removed individuals, determines
71   how the compartments need to be updated. Returns a vector of (nested) keyseqs
72   and corresponding values, which is fed into ->compartments."
73   [t cases removed]
74   [[t :R] removed [t :I] (- cases removed)])
75
76
77 (defn ->I [t value coll]
78   (->coll [t :I] value coll))
79
```

```
80
81 (defn ->R [t value coll]
82   (->coll [t :R] value coll))
```

## Der Namespace util

```
1 (ns sir-model.util
2   (:require [clojure.java.io :as io]
3             [clojure.data.csv :as csv]))
4
5 (defn from-maps
6   "Extracts all vals for key(s) ks from a coll of maps. ks must be given as
7   a vector. Returns a vector."
8   [coll ks]
9   (reduce #(conj %1 (get-in %2 ks)) [] coll))
10
11
12 (defn from-season
13   "Extracts value for key from a single sample. key must be in a result from
14   an anglican query."
15   [sample key]
16   (reduce
17     #(conj %1 (key %2))
18     []
19     (get-in sample [:result :season])))
20
21
22 (defn from-seasons
23   "Extracts value for key from seasons. See from-season."
24   [samples key]
25   (reduce
26     #(conj %1 (from-season %2 key))
27     []
```

```
28     samples))
29
30
31 (defn from-result
32   "Extracts value for key from a single sample. key must be in a result from
33   an anglican query. keys must be given in vector:
34     (from-result a-sample [:a :b])"
35   [sample keys]
36   (get-in sample (flatten [:result keys])))
37
38
39 (defn from-results
40   "Extracts values for key from samples. See from-result."
41   [samples keys]
42   (reduce
43     #(conj %1 (from-result %2 keys))
44     []
45     samples))
46
47
48 (defn total-infected
49   "Extracts number of infected individuals for a single sample from the Model."
50   [sample]
51   (from-season sample :I))
52
53
54 (defn new-infections-in-season
55   [sample]
56   (let [primary (from-season sample :primary)
57         secondary (from-season sample :secondary)]
58     (map #(+ %1 %2) primary secondary)))
59
60
```

```
61 (defn new-infections-in-seasons
62   [samples]
63   (reduce #(conj %1 (new-infections-in-season %2)) [] samples))
64
65
66 (defn sum-compartments
67   "Sum specified compartments of an sir-record. Compartments have to be given
68   as coll. Used to check, if e.g. the [:I :R] compartments sum to the same
69   number over the course of a progression, as they should."
70   ([sir-record comps]
71    (sum-compartments sir-record 0 comps))
72   ([sir-record acc comps]
73    (loop [[head & tail] comps
74          sum acc]
75          (if (not (seq tail))
76              (+ sum (get sir-record head))
77              (recur tail (+ sum (get sir-record head)))))))
78
79
80 (defmulti #^{:private true} data-for-single-season
81          (fn [query-result f sim-id] (sequential? f)))
82
83 (defmethod data-for-single-season false [query-result f sim-id]
84   (let
85     [weeks (range (count (from-result query-result :season)))
86      cases (f query-result)
87      sim-ids (repeat (count weeks) sim-id)]
88
89     (partition 3
90               (interleave weeks cases sim-ids))))
91
92
93 (defmethod data-for-single-season true [query-result f sim-id]
```

```
94 (let
95   [weeks (range (count (from-result query-result :season)))
96     sim-ids (repeat (count weeks) sim-id)
97     cases (map #(%1 query-result) f)]
98
99   (as-> cases v
100     (apply interleave v)
101     (partition (count cases) v)
102     (interleave weeks v sim-ids)
103     (flatten v)
104     (partition (+ (count cases) 2) v))))
105
106
107 (defn write-seasons!
108   "Takes output generated from anglican and writes data for seasons generated
109   by getter-fn to outfile. An additional header can be specified. getter-fn can
110   be a single function or a vector of functions, which can be used to simply
111   collect data from the output or to compute values based on them. Note that
112   these functions have to operate on a single sample, since they are called
113   recursively inside a loop."
114   [samples getter-fn outfile & header]
115   (letfn
116     [(csv-data [samples]
117       (loop [coll []
118             from-query samples
119             n 0]
120
121             (if (not (seq from-query))
122                 coll
123
124                 (let
125                   [single-sample (first from-query)
126                    csv-dat (data-for-single-season single-sample getter-fn n)]
```

```
127
128         (recur (apply conj coll csv-dat)
129                (rest from-query)
130                (inc n))))))]]
131
132 (with-open [writer (io/writer outfile)]
133   (when header
134     (csv/write-csv writer header))
135     (csv/write-csv writer (csv-data samples))))))
136
137
138 (defn write-seasons-as-df!
139   "Write simulation results to be read in as dataframe and plotted by R.
140   Wraps around write-seasons!. outfile is given without file extension
141   and automatically stored as csv-file."
142   [samples outfile]
143   (let
144     [getter-fns [new-infections-in-season
145                 #(from-season % :S)
146                 #(from-season % :I)
147                 #(from-season % :R)
148                 #(from-season % :primary)
149                 #(from-season % :secondary)]
150     filedir "data"
151     path (str filedir "/" outfile ".csv")
152     header ["week" "new" "S" "I" "R" "primary" "secondary" "sim_id"]]
153
154     (write-seasons! samples getter-fns path header)))
155
156
157 (defn write-lambdas!
158   "Write l-1 and l-2 to csv."
159   [samples outfile]
```

```
160 (let [l-1 (from-results samples [:lambda-1])
161       l-2 (from-results samples [:lambda-2])
162       dat-out (->>
163               (interleave l-1 l-2)
164               (partition 2))]
165     (with-open [writer (io/writer outfile)]
166       (csv/write-csv writer [[:lambda-1 "lambda-2"]])
167       (csv/write-csv writer dat-out))))
168
169
170 (defn vec->time-series
171   "Converts a vector of values. Returns a seq of maps with two key-value-pairs
172   each, one for :week and one for :data. This format is useful for plotting with
173   vega-lite."
174   [v]
175   (let [steps (range (count v))
176         steps-and-vals (zipmap steps v)]
177     (for [[t v] steps-and-vals]
178       {:week t :data v})))
179
180
181 (defn vecs->time-series
182   "Converts a nested vector or seq of values. Wrapper around vec->time-series."
183   [vec]
184   (flatten
185     (map
186       #(vec->time-series %)
187       vec)))
188
189
190 (defn extract-for-vega
191   "Extracts data for kw from seasons in an anglican sample and converts to a
192   format for plotting with vega-lite. Works only for single values, not e.g."
```

```
193   for seasons."
194   [samples kw]
195   (->
196     (from-seasons samples kw)
197     (vecs->time-series)))
198
199
200 (defn weekly-plot-spec
201   "Return a spec for vega-lite to plot data for kw in samples by week."
202   [samples kw]
203   {:data      {:values (extract-for-vega samples kw)}
204    :mark      "tick"
205    :encoding  {:x {:field "week"
206                   :type  "nominal"
207                   }
208               :y {:field "data"
209                   :type  "quantitative"}}})
210
211
212 (defn filter-by-week
213   "Takes anglican samples and returns only those for specified week."
214   [samples week]
215   (let
216     [seasons-by-week (vecs->time-series (from-results samples [:season]))
217      week-only (filter #(= week (get % :week)) seasons-by-week)
218      week-only])
219
220
221 (defn week-histo-spec
222   "Returns a spec for vega-lite to plot a histogram of new infections for a
223   specified week."
224   [samples week]
225   (let [new-all (new-infections-in-seasons samples)
```

```
226     week-only (map #(nth % week) new-all)]
227   {:data      {:values week-only}
228    :mark      "bar"
229    :encoding  {:x {:field "data" :type "quantitative"
230                  ;:bin {:maxbins 50}
231                  }
232               :y {:aggregate "count" :type "quantitative"}}})
233
234
235 (defn weekly-dists-spec
236   "Returns a vega-lite spec for vertically plotting the weekly dists for kw."
237   [samples kw]
238   {:data      {:values (extract-for-vega samples kw)}
239    :encoding  {:x {:field :data :type "quantitative"}
240               :y {:field :week :type "ordinal"}}
241    :mark      "tick"
242   })
243
244
245 (defn histo-spec
246   "Returns a vega-lite spec for plotting a histogram for seq of vals."
247   [vals]
248   {:data {:values vals}
249    :mark "bar"
250    :encoding {:x {:field "data"
251                  :type "quantitative"}
252               :y {:aggregate "count"
253                     :type "quantitative"}}})
254
255 (declare hdi-plot-spec)
256
257 (defn new-infections-plot-spec
258   "Create spec to plot weekly new infections using vega-lite. If cred-mass is
```

```

259     supplied, a laayer with highest density intervals (one for each week) is
260     added to the plot."
261     ([samples]
262      {:data      {:values (vecs->time-series (new-infections-in-seasons samples))}
263       :mark      {:type "tick" :opacity 0.3}
264       :encoding  {:x {:field :week :type "ordinal"}
265                  :y {:field :data :type "quantitative"}
266                  :axis  {:title "new infections"}}}))
267     ([samples cred-mass]
268      {:layer [(new-infections-plot-spec samples)
269              (hdi-plot-spec samples cred-mass)]}))
270
271
272 (defn hdi [samples-from-dist cred-mass]
273   "Compute highest density interval for probability distribution represented by
274   samples-from-dist. Algorithm is adopted from Kruschke, J.K. (2015)."
275   (let [sorted (sort samples-from-dist)
276         ;;number of samples-from-dist needed with given cred-mass
277         n-keep (int (Math/floor (* cred-mass (count sorted))))
278         ;;how many CIs to compare
279         n-CIs (- (count sorted) n-keep)
280         ;;computing widths for the different CIs
281         ci-width (reduce #(conj %1 (- (nth sorted (+ %2 n-keep)) (nth sorted %2)))
282                          [])
283         ;;keep the narrowest width
284         min-width (apply min ci-width)
285         ;;locate index of min-width
286         ind-of-min (.indexOf ci-width min-width)
287         ;;get hdi borders
288         borders [(nth sorted ind-of-min) (nth sorted (+ ind-of-min n-keep))]
289                 ]
290                 borders))
291

```

```
292 (defn samples->weekly-new
293   "Takes samples and returns a seq of vectors holding weekly numbers of
294   newly infected individuals"
295   [samples]
296   (->> samples
297     (new-infections-in-seasons)
298     (apply map vector)))
299
300
301 (defn samples->hdi-borders
302   "Compute highest density interval borders for samples as returned by
303   anglican. Wrapper around hdi."
304   [samples cred-mass]
305   (let [weekly (samples->weekly-new samples)]
306     (map #(hdi % cred-mass) weekly)))
307
308
309 (defn borders->vega-lite
310   "Takes a nested seq of vectors of length two, holding the low and high border
311   of the hdi. Returns a seq of maps with keys representing week numbers and
312   values representing the borders."
313   [borders]
314   (let [lo (map first borders)
315         hi (map second borders)]
316     (into (vec->time-series lo) (vec->time-series hi))))
317
318
319 (defn hdi-plot-spec
320   [samples cred-mass]
321   (let [borders (samples->hdi-borders samples cred-mass)]
322     {:data      {:values (borders->vega-lite borders)}
323      :mark      {:type "bar" :opacity 0.3 :color "green"}
324      :encoding  {:x {:field :week :type "ordinal"}}
```

```
325         :y {:field :data :type "quantitative"}}}))
326
327
328 (defn data-plot-spec
329   [data]
330   {:data      {:values (vec->time-series data)}
331    :mark      {:type "tick" :color "red"}
332    :encoding  {:x {:field :week :type "ordinal"}
333               :y {:field :data :type "quantitative"}}})
334
335
336 (defn dashboard-spec
337   "Plot a dashboard-like view of different aspects of the simulation data."
338   ([samples]
339    (let
340      [primary (weekly-plot-spec samples :primary)
341       secondary (weekly-plot-spec samples :secondary)
342       new (new-infections-plot-spec samples 0.95)
343       lambda-1 (histo-spec (from-results samples :lambda-1))
344       lambda-2 (histo-spec (from-results samples :lambda-2))
345       weekly-dists {:data      {:values
346                                (vecs->time-series
347                                 (new-infections-in-seasons samples))}
348                    :mark      "tick"
349                    :encoding  {:x {:field :data :type "quantitative"}
350                               :y {:field :week :type "ordinal"}}}]
351
352      board {:hconcat
353             [[:vconcat [primary secondary new {:hconcat [lambda-1 lambda-2]}]]
354              weekly-dists]}
355
356      ]
357     board)
358   )
```

```

358 ([samples data]
359   (let
360     [primary (weekly-plot-spec samples :primary)
361      secondary (weekly-plot-spec samples :secondary)
362      new-and-data {:layer [(new-infections-plot-spec samples 0.95)
363                          (data-plot-spec data)]}
364      lambda-1 (histo-spec (from-results samples :lambda-1))
365      lambda-2 (histo-spec (from-results samples :lambda-2))
366      weekly-dists {:data      {:values
367                               (vecs->time-series
368                                (new-infections-in-seasons samples))}
369                  :mark      "tick"
370                  :encoding  {:x {:field :data :type "quantitative"}
371                             :y {:field :week :type "ordinal"}}}]
372
373     board {:hconcat
374            [{:vconcat
375              [primary secondary new-and-data {:hconcat [lambda-1 lambda-2]}]}
376            weekly-dists]}
377   ]
378   board)))

```

## Der Namespace core

```

1 (ns sir-model.core
2   (:gen-class)
3   (:require [clojure.java.io :as io]
4             [clojure.data.csv :as csv]
5             [sir-model.util :as util]
6             [sir-model.two-stage-poisson :as model]
7             [com.climate.claypoole :as cp]
8             [oz.core :as oz]
9             [clojure.tools.cli :refer [parse-opts]])

```

```
10  (:use [anglican [core :exclude [-main]] runtime emit stat])
11  )
12
13
14  (def arg-map
15    {:t-max      40
16     :compartments [:S :I :R :primary :secondary]
17     :inits      {:S 100000 :I 2}
18     :prior-1    (uniform-continuous 0.4 0.9)
19     :prior-2    (uniform-continuous 1.0 1.8)
20     :n-samples  10
21     :n-thin     1
22    })
23
24  (def data
25    [5 20 100 120 200 100 300 700
26     1000 1400 1700 1600 1500 1000
27     600 300 200 100 50 20 10 5 4])
28
29  (def counter (agent 0))
30  (add-watch counter :watcher
31    (fn [key agent old-state new-state]
32      (println "sir-model.core: just started simulation" new-state)))
33
34
35  (defn force-sample
36    "Evaluates a lazy-sample and increments counter."
37    [query-results n]
38    (send counter inc)
39    (nth query-results n))
40
41
42  (defn lazy-samples
```

```
43 "Takes an anglican query and returns samples as a lazy-seq."  
44 ([anglican-query args n-particles]  
45 (doquery :smc anglican-query [args] :number-of-particles n-particles))  
46 ([anglican-query args]  
47 (lazy-samples anglican-query args 100)))  
48  
49  
50 (defn pmap-samples  
51 "Takes the result from an anglican doquery and an optional thinning parameter  
52 and extracts n elements out of that lazy-seq."  
53 [query-results n]  
54 (let [n-cpus (.. Runtime getRuntime availableProcessors)  
55       n-threads (+ n-cpus 2)]  
56   ;; if no logging to screen is needed, use the commented version  
57   ;(pmap #(nth query-result %) (range 0 n thin))  
58   (cp/upmap n-threads #(force-sample query-results %) (range 0 n))))  
59  
60  
61 (defn sampler  
62 "Runs an anglican query and forces to results. This is just wrapper around  
63 lazy-samples and pmap-samples."  
64 ([anglican-query args n n-particles]  
65 (-> anglican-query  
66 (lazy-samples args n-particles)  
67 (pmap-samples n)))  
68 ([anglican-query args n]  
69 (sampler anglican-query args n 100)))  
70  
71 (defn write-lambda-priors!  
72 [args n outfile]  
73 (let [l-1-prior (repeatedly n #(sample* (:prior-1 args)))  
74       l-2-prior (repeatedly n #(sample* (:prior-2 args)))  
75       dat-out (->> (interleave l-1-prior l-2-prior)
```

```
76         (partition 2))]
77     (with-open [writer (io/writer outfile)]
78       (csv/write-csv writer [["l_1_prior" "l_2_prior"]])
79       (csv/write-csv writer dat-out))))
80
81 ;(util/write-lambdas! samples "lambdas.csv")
82 ;(write-lambda-priors! arg-map n-samples "priors.csv")
83
84 (defn plot-results! [samples]
85   (oz/start-plot-server!)
86   (oz/v! (util/dashboard-spec samples)))
87
88 (def cli-opts
89   [["-n" "--n-samples n-samples" "Number of program runs for influenza season"
90     :default 1000
91     :parse-fn #(Integer/parseInt %)]
92    ["-S" "--n-susceptible n-susceptible" "Population size"
93     :default 100000
94     :parse-fn #(Integer/parseInt %)]
95    ["-I" "--n-infected n-infected" "Number of initially infected individuals"
96     :default 2
97     :parse-fn #(Integer/parseInt %)]
98    ["-t" "--t-max t-max" "Number of weeks for season"
99     :default 40
100    :parse-fn #(Integer/parseInt %)]
101    ["-p" "--n-particles n-particles" "Number of particles"
102     :default 100
103     :parse-fn #(Integer/parseInt %)]
104    ["-o" "--outfile"
105     :id :outfile
106     :required "Season samples are written to this file. File is automatically
107     stored in data/."
108     ]
```

```
109 ["-l" "--lambda-outfile"  
110      :id :lambda-outfile  
111      :required "Estimated parameters are written to this file"]  
112 ["-d" "--data" "Empirical data to condition on"]  
113 ["-g" "--plot-results"]  
114 ["-h" "--help"]])  
115  
116 (defn -main [& args]  
117   (let [  
118     ;; get cli arguments  
119     parsed-opts (parse-opts args cli-opts)  
120     n-samples (get-in parsed-opts [:options :n-samples])  
121     n-susceptible (get-in parsed-opts [:options :n-susceptible])  
122     n-infected (get-in parsed-opts [:options :n-infected])  
123     t-max (get-in parsed-opts [:options :t-max])  
124     n-particles (get-in parsed-opts [:options :n-particles])  
125  
126     ;; update arguments  
127     arguments (-> arg-map  
128                (assoc-in [:n-samples] n-samples)  
129                (assoc-in [:inits :S] n-susceptible)  
130                (assoc-in [:inits :I] n-infected)  
131                (assoc-in [:t-max] t-max))  
132  
133     ;; run model with or without data  
134     samples (if (get-in parsed-opts [:options :data])  
135                (sampler  
136                  model/two-stage-poisson-query  
137                  (assoc-in arguments [:data] data)  
138                  n-samples  
139                  n-particles)  
140                (sampler  
141                  model/two-stage-poisson-query
```

```
142         arguments
143         n-samples
144         n-particles))
145     ]
146
147     (if-let [outfile (get-in parsed-opts [:options :outfile])]
148       (util/write-seasons-as-df! samples outfile))
149
150     (if-let [lambda-outfile (get-in parsed-opts [:options :lambda-outfile])]
151       (util/write-lambdas! samples lambda-outfile))))
```

# Anhang B

## Quellcode zu Kapitel 7

Die Simulation aus Kapitel 7 ist auf folgende namespaces aufgeteilt: Der namespace `model` enthält die Implementierungen der Funktionen des eigentlichen Modells. Hier ist z.B. die probabilistische Funktion `look-around` definiert, die für jeden Teilnehmer des Graphen aufgerufen wird. Dabei schaut sich jeder Teilnehmer seine Nachbarn an, um festzustellen, wieviele von Ihnen die Falschmeldung übernommen haben. Im namespace `gen` werden Funktionen bereit gestellt, mit denen die Graphen erzeugt werden. Funktionen, um die Ergebnisdaten in eine Datei zu schreiben, werden in der Datei `io.clj` implementiert, Funktionen zur Erzeugung von Graphiken in der Datei `view.clj`. Im namespace `core` werden die Funktionen zum Erzeugen der Graphen, zum Starten der Simulation und zum Schreiben der Ergebnisse aufgerufen.

### Der Namespace `model`

```
1 (ns diffusion.model
2   (:require [loom.graph :as g]
3             [loom.attr :as a])
4   (:use [anglican runtime emit]))
5
6
7 (defn initialize-graph
8   "Add attributes :green? and :time-without to nodes of g. Initialize all
```

```
9   with zero and set :green? of specified proportion of nodes to 1."
10  [g proportion]
11  (let
12    [n-nodes (count (g/nodes g))
13     n-adopters (int (* proportion n-nodes))
14     adopters (repeatedly n-adopters #(rand-nth (range n-nodes)))]
15    (-> g
16      (a/add-attr-to-all :fake? 0)
17      (a/add-attr-to-all :time-without 0)
18      (a/add-attr-to-nodes :fake? 1 adopters))))
19
20
21  (defn count-attr-for-neighbours
22    "Sum up values for attribute attr for all neighbours of node in g."
23    [g node attr]
24    (reduce #(+ %1 (a/attr g %2 attr)) 0 (g/successors g node)))
25
26
27  (defn collect-attr-in-graph
28    [g attr]
29    (reduce #(conj %1 (a/attr g %2 attr)) [] (g/nodes g)))
30
31
32  (defn count-attr-in-graph
33    [g attr]
34    (reduce #(+ %1 (a/attr g %2 attr)) 0 (g/nodes g)))
35
36
37  (defn count-fake-news
38    [g node]
39    (count-attr-for-neighbours g node :fake?))
40
41
```

```

42 (defdist decision-dist
43   [g node]
44   [n-neighbours (count (g/successors g node))
45    n-green (count-fake-news g node)
46    ratio (/ n-green n-neighbours)]
47   (sample* [this]
48    (if (zero? ratio)
49        (let [param (sample* (uniform-continuous 0.01 0.1))]
50            (sample* (bernoulli param)))
51        (sample* (bernoulli (* 0.1 ratio))))))
52
53
54 (with-primitive-procedures [decision-dist]
55   (defquery decision-dist-query [max-neighbours]
56     (let [ns (range (inc max-neighbours))
57           *sample (fn [green-neighbours]
58                     (let [ratio (/ green-neighbours max-neighbours)]
59                         (if (zero? ratio)
60                             (let [param (sample (uniform-continuous 0.01 0.1))]
61                                 (sample* (bernoulli param)))
62                             (sample (bernoulli (* 0.1 ratio))))))]
63       {:decisions (reduce (fn [coll val] (conj coll (*sample val))) [] ns))}))
64
65
66 (with-primitive-procedures
67   [count-fake-news g/successors a/attr a/add-attr decision-dist]
68   (defm look-around
69     "Decision function for the diffusion model. Every node in the graph looks
70     at his neighbours and counts how many of them are already green. This number
71     is used as a parameter for a probabilistic decision to buy a green car."
72     [g node]
73     (if (pos? (attr g node :fake?))
74         g

```

```
75     (let
76       [time (attr g node :time-without)
77         ;n-neighbours (count (successors g node))
78         ;green-neighbours (count-green-neighbours g node)
79         ;; new param to make decision harder
80         ;param (* 0.001 (/ green-neighbours n-neighbours))
81
82         ;; use custom dist
83         green? (sample (decision-dist g node))
84       ]
85
86       (if (pos? green?)
87         (add-attr g node :fake? 1)
88         (add-attr g node :time-without (inc time))))))
89
90
91 (with-primitive-procedures
92   [g/nodes]
93   (defm diffusion-step
94     [g]
95     (reduce #(look-around %1 %2) g (nodes g))))
96
97
98 (with-primitive-procedures
99   [count-attr-in-graph]
100   (defm run-sim
101     [t-max g]
102     (loop [t 0
103            graph g
104            n-green []]
105       (if (= t t-max)
106         {:n-green n-green :graph graph}
107         (recur (inc t) (count-attr-in-graph g) (count-attr-in-graph g)))))
```

```

108         (recur (inc t)
109                (diffusion-step graph)
110                (conj n-green (count-attr-in-graph graph :fake?))))))
111
112
113 (with-primitive-procedures
114   [count-fake-news count-attr-for-neighbours count-attr-in-graph
115    collect-attr-in-graph]
116   (defquery
117     diffusion-query [graph]
118     (let
119       [t 50
120        history (run-sim t graph)
121        ]
122       {:history history})))
123
124
125 (defn list-degrees [g]
126   "Returns a vector where each element corresponds to the degree of a node in g."
127   (reduce #(conj %1 (count (g/successors g %2))) [] (g/nodes g)))

```

## Der Namespace gen

```

1 (ns diffusion.gen
2   (:require [loom.graph :as g]
3             [loom.attr :as a]))
4
5
6 (defn gen-circle
7   "Adds num-nodes nodes to graph g and connects each one to out-degree
8   other nodes."
9   [g num-nodes out-degree]
10  {:pre [(> num-nodes (* out-degree 2))]}

```

```
11 (let [nodes (range num-nodes)
12       edges (for [n nodes
13                 d (range 1 (inc out-degree))]
14                 [n (mod (+ n d) (count nodes))])]
15     (-> g
16         (g/add-nodes* nodes)
17         (g/add-edges* edges))))
18
19
20 (defn ^:private add-shortcuts
21     "Computes additional edges for graph g as described in Newman and Watts (1999).\"
22     ([g phi seed]
23      (let [rnd (java.util.Random. seed)
24            nodes (g/nodes g)
25            shortcuts (for [n nodes
26                          :when (> phi (.nextDouble rnd))]
27                          [n (.nextInt rnd (count nodes))])]
28          (-> g
29              (g/add-edges* shortcuts))))))
30
31
32 (defn gen-newman-watts
33     "Generate a graph with small-world properties as described in Newman and Watts
34     (1999).\"
35     ([g num-nodes out-degree phi seed]
36      (-> g
37          (gen-circle num-nodes out-degree)
38          (add-shortcuts phi seed)))
39     ([g num-nodes out-degree phi]
40      (gen-newman-watts g num-nodes out-degree phi (System/nanoTime))))
41
42
43 (defn create-graph-from-adj-list
```

---

```

44 [adj-lst]
45 (-> adj-lst
46   (g/graph)
47   (a/add-attr-to-all :fake? 0)
48   (a/add-attr-to-all :time-without 0)))
49
50
51 ;; "deprecated",
52 (defn create-graph-from-snap-edge-list
53   "Creates a graph from an edge list as created by one of the functions from the
54   Stanford Network Analysis Project (SNAP),"
55   [path]
56   (as-> path v
57     (slurp v)
58     (re-seq #"\d+" v)
59     (map #(Integer/parseInt %) v)
60     (partition 2 v)
61     (apply g/graph v)))
62
63
64 ;(def path "datasets/data")
65 (def path "/home/dusty/dev/python/snappy/edgeList.txt")
66
67
68 (defn gen-barabasi-albert
69   "Generate a preferential attachment graph as described in Barabasi
70   and Albert (1999)."

```

```
77     ;; predicate for deciding whether a node
78     ;; should be connected to a new node
79     connect? (fn [g node]
80               (let [degree-node (count (g/successors g node))
81                     degree-sum (reduce #(+ %1 (count (g/successors g %2)))
82                                         0 (g/nodes g))]
83                 (<= (/ degree-node degree-sum) (.nextDouble rnd))))
84     ;; go through all nodes in g and decide whether
85     ;; they connect to new
86     new-edges (fn [g new]
87                (for [n (g/nodes g)
88                      :when (connect? g n)]
89                  [new n]))
90     ;; compute num-edges edges for new in graph g
91     get-new-edges-and-connect (fn [g new num-edges]
92                                (as-> g v
93                                      (new-edges v new)
94                                      (take num-edges v)
95                                      (filter #(= 2 (count %)) v)
96                                      (apply g/add-edges g v)))
97     ;; two nodes are already in the initialized graph
98     ;; the remaining nodes will be added
99     remaining-nodes (range 2 num-nodes)
100   ]
101
102   (reduce #(get-new-edges-and-connect %1 %2 num-edges) g-0 remaining-nodes))
103 ([g num-nodes num-edges]
104  (gen-barabasi-albert g num-nodes num-edges (System/nanoTime))))
```

## Der Namespace view

```
1 (ns diffusion.view
2   (:require [loom.attr :as a]
```

```
3         [loom.graph :as g]
4         [ministrants.core :as m]))
5
6
7 (defn from-maps
8   "Extracts all vals for key(s) ks from a coll of maps. ks must be given as
9   a vector. Returns a vector."
10  [coll ks]
11  (reduce #(conj %1 (get-in %2 ks)) [] coll))
12
13
14 (defn extract-values
15   [raw-results kw]
16   (reduce #(conj %1 (get-in %2 [:result kw])) [] raw-results))
17
18
19 (defn times-in-g
20   [g]
21   (reduce #(conj %1 (a/attr g %2 :time-without)) [] (g/nodes g)))
22
23
24 (defn times-in-gs
25   "Compute vega-lite-spec for histogram of time until green."
26   [gs]
27   (reduce #(conj %1 (times-in-g %2)) [] gs))
28
29 (defn histo-spec
30   [data]
31   {:data {:values data}
32    :mark "bar"
33    :encoding {:x {:field :data :type "quantitative" }
34              :y {:aggregate "count" :type "quantitative"}}})
35
```

```
36
37 (defn vec->vega-time-series
38   "Converts a vector of values. Returns a vector of maps, which have two
39   key-value pairs, one for :week and one for :data (cases). This format can be
40   supplied as a values vector for vega-lite."
41   [vec]
42   (letfn [(vec->series
43           [v]
44           (let [steps (range (count v))
45                 steps-and-vals (zipmap steps v)]
46             (for [[t v] steps-and-vals]
47               {:week t :data v}))
48           )
49   ]
50   (flatten
51     (map
52       #(vec->series %)
53       vec))))
54
55
56 (defn filter-by-step
57   "Takes anglican samples and returns only those for specified timestep."
58   [samples step]
59   (let
60     [values-by-step (vec->vega-time-series (extract-values samples :history))
61       week-only (filter #(= step (get % :week)) values-by-step)]
62     week-only))
63
64
65 (defn week-histo-spec
66   "Returns a spec for vega-lite to plot a histogram of new infections for a
67   specified week."
68   [samples week]
```

```

69   {:data      {:values (from-maps (filter-by-step samples week) [:data])}
70    :mark      "bar"
71    :encoding  {:x {:field "data" :type "quantitative"}
72               :y {:aggregate "count" :type "quantitative"}}})
73
74
75 (defn season-plot-spec
76   [samples]
77   (let [seasons (m/from-results samples [:history :n-green])
78         formatted (vec->vega-time-series seasons)]
79     {:data      {:values formatted}
80      :mark      "tick"
81      :encoding  {:x {:field :week :type "ordinal"}
82                 :y {:field :data :type "quantitative"}}
83     })

```

## Der Namespace io

```

1 (ns diffusion.io
2   (:require [ministrants.core :as m]
3             [clojure.java.io :as jio]
4             [clojure.data.csv :as csv]))
5
6 (defmulti #^{:private true} data-for-single-season
7   (fn [query-result f sim-id] (sequential? f)))
8
9 (defmethod data-for-single-season false [query-result f sim-id]
10  (let
11    [weeks (range (count (m/from-result query-result [:history :n-green])))
12     cases (f query-result)
13     sim-ids (repeat (count weeks) sim-id)]
14
15    (partition 3

```

```
16         (interleave weeks cases sim-ids))))
17
18
19 (defmethod data-for-single-season true [query-result f sim-id]
20   (let
21     [weeks (range (count (m/from-result query-result [:history :n-green]))))
22     sim-ids (repeat (count weeks) sim-id)
23     cases (map #(%1 query-result) f)]
24
25     (as-> cases v
26       (apply interleave v)
27       (partition (count cases) v)
28       (interleave weeks v sim-ids)
29       (flatten v)
30       (partition (+ (count cases) 2) v))))
31
32 (defn write-seasons!
33   "Takes output generated from anglican and writes data for seasons generated
34   by getter-fn to outfile. An additional header can be specified. getter-fn can
35   be a single function or a vector of functions, which can be used to simply
36   collect data from the output or to compute values based on them. Note that
37   these functions have to operate on a single sample, since they are called
38   recursively inside a loop."
39   [samples getter-fn outfile & header]
40   (letfn
41     [(csv-data [samples]
42       (loop [coll []
43             from-query samples
44             n 0]
45
46         (if (not (seq from-query))
47             coll
```

```

49         (let
50             [single-sample (first from-query)
51              csv-dat (data-for-single-season single-sample getter-fn n)]
52
53             (recur (apply conj coll csv-dat)
54                    (rest from-query)
55                    (inc n))))))]]
56
57 (with-open [writer (jio/writer outfile)]
58   (when header
59     (csv/write-csv writer header))
60   (csv/write-csv writer (csv-data samples))))))
61
62
63 (defn write-degrees-csv!
64   [datavec outfile]
65   (let [out (map vector datavec)]
66     (with-open [writer (clojure.java.io/writer outfile)]
67       (csv/write-csv writer out))))

```

## Der Namespace core

```

1 (ns diffusion.core
2   (:require [loom.graph :as g]
3             [loom.io :as io]
4             [oz.core :as oz]
5             [ministrants.core :as m]
6             [anglican.core :refer [doquery]]
7             [diffusion.model :refer [diffusion-query initialize-graph list-degrees]
8             [diffusion.io :as dio]
9             [diffusion.gen :refer :all]
10            [clojure.tools.cli :refer [parse-opts]]
11            [clojure.data.csv :as csv])

```

```
12  (:gen-class))
13
14
15  (def cli-opts
16    [ ["-g" "--graph-type graph-type" "Type of graph used in simulation"
17      :default "newman-watts"]
18      ["-d" "--degree degree" "(Out-)Degree for nodes in graph"
19      :default 5]
20      ["-h" "--phi phi" "Parameter for adding edges to Newman-Watts graph"
21      :parse-fn #(Float/parseFloat %)
22      :default 0.3
23      :validate [#(<= 0.0 % 1.0) "Parameter for adding edges to Newman-Watts
24                  graph has to be between 0 and 1"]]
25      ["-n" "--n-samples n-samples" "Number of simulations run"
26      :default 100
27      :parse-fn #(Integer/parseInt %)]
28      ["-N" "--n-nodes n-nodes" "Number of nodes in graph"
29      :default 50
30      :parse-fn #(Integer/parseInt %)]
31      ["-a" "--algorithm algorithm" "Algorithm used for sampling from model"
32      :default :smc
33      :parse-fn #(keyword %)]
34      ["-o" "--outfile"
35      :required "Path to write results to"
36      :id :outfile]
37      ["-v" "--dotfile"
38      :required "Path to write dotfile to"
39      :id :dotfile]
40      ["-e" "--degreefile"
41      :required "Path to write degrees to"
42      :id :degreefile]
43    ])
44
```

```

45
46 (defn -main
47   "Runs model and writes results according to arguments passed via command line."
48   [& args]
49   (let [parsed-args (parse-opts args cli-opts)
50         graph-type (get-in parsed-args [:options :graph-type])
51         n-nodes (get-in parsed-args [:options :n-nodes])
52         degree (get-in parsed-args [:options :degree])
53
54         ;; construct graph
55         in-graph (-> (if (= graph-type "newman-watts")
56                        (gen-newman-watts (g/graph) n-nodes degree
57                                          (get-in parsed-args [:options :phi]))
58                        (gen-barabasi-albert (g/graph) n-nodes degree))
59                (initialize-graph 0.1))
60
61         ;; run model
62         n-samples (get-in parsed-args [:options :n-samples])
63         samples (doall (take n-samples
64                             (doquery
65                               :smc
66                               diffusion-query
67                               [in-graph]
68                               :number-of-particles 100)))
69         ]
70     ;; write results
71     (if-let [dotfile (get-in parsed-args [:options :dotfile])]
72       (io/dot in-graph dotfile))
73
74     (if-let [outfile (get-in parsed-args [:options :outfile])]
75       (dio/write-seasons! samples #(m/from-result % [:history :n-green]) outfile))
76
77     (if-let [degreefile (get-in parsed-args [:options :degreefile])]

```

```
78     (dio/write-degrees-csv!  
79       (-> (first samples)  
80           (m/from-result [:history :graph])  
81             (list-degrees))  
82       degreefile))))
```

# Anhang C

## Quellcode zu Kapitel 8

### Der Namespace model

```
1 (ns social-unequality.model
2   (:require
3     [anglican.runtime :refer [uniform-continuous flip normal sample*]]
4     [anglican.emit :refer [defquery query defm fm with-primitive-procedures]]
5     [loom.gen :refer [gen-newman-watts gen-barabasi-albert]]
6     [loom.graph :refer [graph nodes successors add-edges remove-edges]]
7     [loom.attr :refer [add-attr attr]]
8     [clojure.set :as set]))
9
10 ;; initializing
11
12 (defn commodity
13   "Get commodity value for node in g."
14   [g node]
15   (attr g node :commodity))
16
17 (defn init-commodity
18   "Distributing a commodity randomly over the nodes of graph g."
19   [g low high]
```

```
20 (reduce #(add-attr %1 %2 :commodity
21           (sample* (uniform-continuous low high))) g (nodes g)))
22
23 (defn init-graph
24   "Generates either Newman-Watts or Barabasi-Albert graph and
25   initializes commodity values to random numbers between 0 and 1."
26   ([n-nodes out-degree]
27     (->
28       (gen-barabasi-albert (graph) n-nodes out-degree)
29       (init-commodity 0 1)))
30   ([n-nodes out-degree param]
31     (->
32       (gen-newman-watts (graph) n-nodes out-degree param)
33       (init-commodity 0 1))))
34
35 ;; model functions
36
37 (with-primitive-procedures [commodity add-attr]
38   (defm unicast-commodity
39     "Adds commodity to node in graph g. Amount is samples from a normal
40     distribution centered around the old value for node."
41     [g node]
42     (let [bonus (sample (normal (commodity g node) 0.01))]
43       (if (pos? bonus)
44         (let [old-value (commodity g node)
45               new-value (+ old-value bonus)]
46           (add-attr g node :commodity new-value))
47         g))))
48
49 (with-primitive-procedures [nodes]
50   (defm broadcast-commodity [g]
51     (reduce #(unicast-commodity %1 %2) g (nodes g))))
52
```

---

```

53 ;; should this be an anglican function?
54 (defn difference-too-big?
55   [g n1 n2]
56   (let [diff (- (commodity g n1) (commodity g n2))]
57     ;; Is this the right test? Somewhat arbitrarily
58     ;; so far.
59     (> diff (/ (commodity g n1) 4))))
60
61 (with-primitive-procedures [difference-too-big? commodity remove-edges]
62   (defm break-single-tie
63     "Decides whether node breaks tie with neighbour."
64     [g node neighbour]
65     (if (difference-too-big? g node neighbour)
66         (let
67           [diff (- (commodity g node) (commodity g neighbour))]
68             (if (sample (flip diff))
69                 (remove-edges g [node neighbour])
70                 g))
71         g)))
72
73 (with-primitive-procedures [successors]
74   (defm break-all-ties
75     "For all neighbours of node, a decision is made wether to break a tie or not."
76     [g node]
77     (let [neighbours (successors g node)]
78       (if (not-empty neighbours)
79           (reduce #(break-single-tie %1 node %2) g neighbours)
80           g))))
81
82 (with-primitive-procedures [successors difference-too-big? commodity add-edges]
83   (defm new-tie
84     "Proposes a new connection from node to candidate. Candidate decides, if
85     she wants to accept the proposal."

```

```
86     [g node candidate]
87     (if (or
88         (contains? (successors g node) candidate)
89         (difference-too-big? g candidate node))
90         g
91         (let
92             [diff (- (commodity g candidate) (commodity g node))]
93             (if (sample (flip (- 1 diff)))
94                 (add-edges g [node candidate])
95                 g))))))
96
97 (with-primitive-procedures [set/difference successors]
98 (defm new-ties
99     "Goes through the coll of candidates and proposes a new tie to each one."
100    [g node candidates]
101    (let [true-candidates (difference (set candidates) (successors g node))]
102        (if (seq true-candidates)
103            (reduce #(new-tie %1 node %2) g true-candidates)
104            g))))
105
106 (defn get-candidates
107     "Returns the top n nodes with respect to commodity from g. If percent is
108     given, nodes are chosen at random from the given top percent of nodes."
109    ([g n]
110     (take n (sort-by #(commodity g %) > (nodes g))))
111    ([g n percent]
112     (->> (nodes g)
113           (sort-by #(commodity g %) >)
114           (take (int (* (count (nodes g)) percent)),,,)
115           (shuffle,,,)
116           (take n,,))))))
117
118 (defn dissimilar-neighbours
```

```
119 [g node]
120 (for [neighbour (successors g node)
121       :when (difference-too-big? g node neighbour)]
122   neighbour))
123
124 (defn pick-neighbour
125   "Pick a neighbour at random."
126   [neighbours]
127   (-> neighbours
128     (shuffle)
129     (first)))
130
131 (with-primitive-procedures [dissimilar-neighbours pick-neighbour remove-edges]
132   (defm remove-tie
133     "Decides wether to remove a tie to a dissimilar neighbour.
134     dissimilar-neighbours returns a seq of dissimilar neighbours.
135     pick-neighbour picks on of them. Choosing a neighbour is only
136     done after flipping a coin."
137     [g node]
138     (let [dissimilar (dissimilar-neighbours g node)]
139       (if (and
140           (sample* (flip 0.5))
141           (seq dissimilar))
142         (remove-edges g [node (pick-neighbour dissimilar)])
143         g))))
144
145 (with-primitive-procedures [get-candidates nodes successors]
146   (defm single-run
147     "Runs the simulation for a single timestep."
148     [g]
149     (let [
150           ;; number of proposals and top percentage of
151           ;; considered nodes are hard coded as follows
```

```
152     candidates (get-candidates g 2 0.05)
153
154     ;; bind functions for proposal and removal
155     ;; of nodes to the following vars to allow
156     ;; for easier modification of the model
157     ;; propose-fn must take a graph, a node and
158     ;; a coll with candidates
159     ;; remove-fn must take a graph and a node
160     propose-fn new-ties
161     remove-fn remove-tie
162
163     ;; perform all actions necessary for a node that
164     ;; are necessary for the simulation
165     cast-propose-remove (fn [g node]
166                          (let [g-update
167                                (propose-fn
168                                 (unicast-commodity g node)
169                                 node
170                                 candidates)]
171                                ;; only call remove-fn if new edges
172                                ;; have been added
173                                (if (>
174                                    (count (successors g-update node))
175                                    (count (successors g node)))
176                                    (remove-fn g-update node)
177                                    g-update)))
178     ]
179     ;; reduce the above function over all nodes
180     (reduce #(cast-propose-remove %1 %2) g (nodes g))))
181
182 (defm sim
183   "Takes a graph g and a number n-timesteps and runs the simulation for
184   n-timesteps. Returns a new graph."
```

```

185 [g n-timesteps]
186 (loop [gg g
187       timer (range n-timesteps)]
188   (if (empty? timer)
189       ;; if time's up return the graph
190       gg
191       ;; else do another round
192       (recur (single-run gg)
193              (rest timer))))))
194
195 (defquery social-inequality [g n-timesteps]
196   (let [
197         g-out (sim g n-timesteps)
198         ;g-out (single-run g)
199         ]
200     {:g-out g-out}))

```

## Der Namespace analysis

```

1 (ns social-unequality.analysis
2   (:require
3     [loom.graph :refer [nodes successors]]
4     [loom.attr :refer [add-attr attr]]
5     [ministrants.core :refer [from-results histo-spec]]
6   ))
7
8
9 (defn get-attr-values
10  "Helper function to collect values for attr-id."
11  ([g node attr-id]
12   (attr g node attr-id))
13  ([g attr-id]
14   (reduce #(conj %1 (attr g %2 attr-id)) [] (nodes g))))

```

```
15
16 (defn commodities
17   "Collect values for commodity."
18   ([g node]
19    (get-attr-values g node :commodity))
20   ([g]
21    (get-attr-values g :commodity)))
22
23 (defn max-commodity
24   [g]
25   (let [commodities (get-attr-values g :commodity)]
26     (apply max commodities)))
27
28 (defn sum-commodities
29   [g]
30   (let [commodities (get-attr-values g :commodity)]
31     (reduce + 0 commodities)))
32
33 ;; this works in principle, but due to rounding errors
34 ;; it does not sum to 1.0
35 ;; what can I do?
36 (defn normalize-commodity
37   [g]
38   (let [max-com (max-commodity g)
39         sum-com (sum-commodities g)
40         commodities (get-attr-values g :commodity)]
41     (-> commodities
42          (map #(/ % max-com))
43          (map #(/ % sum-com)))))
44
45
46 (defprotocol Degrees
47   (list-degrees [g-or-gs] "Collects all the degrees found in a coll."))
```

```
48
49 (extend-protocol Degrees
50   loom.graph.BasicEditableGraph
51   (list-degrees [g]
52     "Returns a vector where each element corresponds to the degree of a node
53     in g."
54     (reduce #(conj %1 (count (successors g %2))) [] (nodes g)))
55
56   clojure.lang.Sequential
57   (list-degrees [gs]
58     "Returns a coll holding all the degrees found in the graphs gs."
59     (reduce #(into %1 (list-degrees %2)) [] gs)))
60
61 (defn degree-frequencies
62   "Returns the frequencies of degrees found in graph g as a map with keys
63   representing the degree and vals representing the number of occurrences
64   of this degree in g."
65   [g]
66   (->
67     (reduce #(conj %1 (count (successors g %2))) [] (nodes g))
68     (frequencies)))
69
70 (defn degree-frequency-distribution
71   "Returns the distribution over degree distributions in a number of samples
72   from an anglican run. The resulting distribution shows how often each degree
73   was found over the course of the different runs of the program."
74   [samples]
75   (let [gs (from-results samples [:final])]
76     (reduce
77       #(merge-with + %1 (degree-frequencies %2))
78       (degree-frequencies (first gs))
79       (rest gs))))
```

## Der Namespace core

```
1 (ns social-inequality.core
2   (:require [loom.gen :refer [gen-newman-watts gen-barabasi-albert]]
3             [loom.graph :refer [graph nodes successors add-edges remove-edges]]
4             [loom.io :as io]
5             [anglican.core :refer [doquery]]
6             [ministrants.core :refer [from-results histo-spec]]
7             [oz.core :as oz]
8             [clojure.data.csv :as csv]
9             [social-inequality.model :refer [social-inequality init-graph commodity
10            social-inequality.analysis :refer :all])
11   (:gen-class))
12
13 ;; running model and analyzing results
14
15 (defn get-samples [query g n-timesteps n-samples]
16   (take n-samples (doquery :smc query [g n-timesteps] :number-of-particles 100)))
17
18 (defn write-degrees-csv!
19   ([before after outfile]
20    (let [out (map vector before after)]
21      (with-open [writer (clojure.java.io/writer outfile)]
22        (csv/write-csv writer out))))
23   ([datavec outfile]
24    (let [out (map vector datavec)]
25      (with-open [writer (clojure.java.io/writer outfile)]
26        (csv/write-csv writer out))))))
27
28 (defn write-out
29   "Calls f to write data to outfile."
30   [f outfile data]
31   (do
```

```

32     (println (str "writing " outfile "..."))
33     (f data outfile)))
34
35 (defn write-results!
36   [graph-type g-init samples & args]
37   (let [gs (from-results samples [:g-out])
38         g-after (first gs)
39         degrees-before (list-degrees g-init)
40         degrees-after (list-degrees g)
41         commodity-before (get-attr-values g-init :commodity)
42         commodity-after (commodities g-after)]
43     (dorun
44       (map write-out
45         [io/dot io/dot write-degrees-csv! write-degrees-csv!
46          write-degrees-csv! write-degrees-csv!]
47         [(str graph-type "_g-init.dot") (str graph-type "_g-after.dot")
48          (str graph-type "_degrees-before.csv") (str graph-type
49                                                    "_degrees-after.csv")
50          (str graph-type "_commodity-before.csv")
51          (str graph-type "_commodity-after.csv")]
52         [g-init g-after degrees-before degrees-after
53          commodity-before commodity-after])))
54
55 (defmulti -main "Run num-samples simulations and write results to disk."
56   (fn [graph-type n-samples n-timesteps n-nodes out-degree & phi]
57     (class n-samples)) )
58
59 (defmethod -main java.lang.Number
60   [graph-type n-samples n-timesteps n-nodes out-degree & phi]
61   (let [g-init (if (seq phi)
62                  (init-graph n-nodes out-degree (first phi))
63                  (init-graph n-nodes out-degree))
64         samples (get-samples social-inequality g-init n-timesteps n-samples)

```

```
65     ]
66     (write-results! graph-type g-init samples)))
67
68 (defmethod -main java.lang.String
69 [graph-type n-samples n-timesteps n-nodes out-degree & phi]
70 (let [n-samples-int (Integer/parseInt n-samples)
71       n-timesteps-int (Integer/parseInt n-timesteps)
72       n-nodes-int (Integer/parseInt n-nodes)
73       out-degree-int (Integer/parseInt out-degree)]
74 (if (seq phi)
75     (-main graph-type
76           n-samples-int
77           n-timesteps-int
78           n-nodes-int
79           out-degree-int
80           (Double/parseDouble (first phi))))
81 (-main graph-type
82       n-samples-int
83       n-timesteps-int
84       n-nodes-int
85       out-degree-int))))
```

# Eigenständigkeitserklärung

Hiermit erkläre und versichere ich, dass ich die vorliegende Arbeit mit dem Titel

**Modellierung und Simulation Systemischer Risiken mittels Probabilistischer Programmierung auf Basis eines integrativen Rahmenmodells**

selbständig verfasst und—einschließlich beigefügter Abbildungen—keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Dies gilt in gleicher Weise für gedruckte Quellen wie für Quellen aus dem Internet. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, sind unter Angabe der Quelle als Zitat kenntlich gemacht worden. Graphische Darstellungen habe ich, wo keine Quelle explizit angegeben ist, selbständig angefertigt.

Stuttgart, den 30. November 2019